STUDY PLAN

Informatics Master Science YEAR 1 (SEMESTER 1)

	Area of Specialization	Unit Code	Unit	Courses Code	Courses	Semester Credit Hours					ECTS Credits	
						Lecture (Courses /TD)	Workshop	Project	Self Study	Total	ECTS-C	ECTS-U
First Semester	common	U-1.1	Mathematical Foundations 1	C-1.1.1	Mathematics for engineers	31,5	0	0	25	56	2	4
				C-1.1.2	Probability and Statistics	21	0	0	35	56	2	
		U-1.2	Fundamental Computing 1	C-1.2.1	Algorithms and Data Structures 1	21	21	0	42	84	3	5
				C-1.2.2	Automata Theory	21	0	0	34	56	2	
		U-1.3	Architecture and Operating System	C-1.3.1	Computer Architecture	21	21	0	42	84	3	6
				C-1.3.2	Operating Systems & Workshop	21	42	0	21	84	3	
		U-1.4	Web and Programming	C-1.4.1	Python Programming Workshop	0	21	0	35	56	2	5
				C-1.4.2	Web Development 1	21	21	10,5	32	84	3	
		U-1.5	Network and Data Transmission	C-1.5.1	Data Transmission	21	21	0	42	84	3	6
				C-1.5.2	Computing Networks	21	21	0	42	84	3	
		U-1.6	Transversal Unit 1	C-1.6.1	Communication Skills	21	0	0	35	56	2	4
				C-1.6.2	Technical English 1 (Scientific English)	21	0	0	35	56	2	
		Sub Total				**241,5**	**168**	**10,5**	**420**	**840**	**30**	
		Total Semester Workload				**420**						

STUDY PLAN

Informatics Master Science YEAR 1 (SEMESTER 2)

	Area of Specialization	Unit Coded	Unit	Courses Code	Courses	Semester Credit Hours					ECTS Credits	
						Lecture (Courses /TD)	Workshop	Project	Self Study	Total	ECTS-C	ECTS-U
Second Semester	common	U-2.1	Mathematical Foundations 2	C-2.1.1	Discret mathematics (Graph Theory and Operational Research)	31,5	0	0	25	56	2	4
				C-2.1.2	Numerical Analysis	31,5	0	0	25	56	2	
		U-2.2	Fundamental Computing 2	C-2.2.1	Algorithms and data Structures 2	21	21	0	42	84	3	5
				C-2.2.2	Computational Complexity	21	0	0	35	56	2	
		U-2.3	Design and Theory	C-2.3.1	Language Theory and Compilation	10,5	21	10,5	14	56	2	5
				C-2.3.2	Analysis and Design Methodology	21	21	0	42	84	3	
		U-2.4	Web and Programming 2	C-2.4.1	Web Development 2	21	10,5	0	25	56	2	5
				C-2.4.2	Object Oriented Programming (OOP)	21	21	0	42	84	3	
		U-2.5	Network and Databases	C-2.5.1	Internet and Protocols	21	10,5	0	23	56	2	5
				C-2.5.2	Database Fundamentals	21	10,5	10,5	42	84	3	
		U-2.6	Transversal Unit 2	C-2.6.1	Engineer's Ethics	21	0	0	35	56	2	4
				C-2.6.2	English TOEIC 1	21	0	0	35	56	2	
		U-2.7	Project Unit 1	C-2.7.1	Personal Project	0	21	0	35	56	2	2
		Sub Total				**262,5**	**136,5**	**21**	**420**	**840**	**30**	
		Total Semester Workload				**420**						

STUDY PLAN

Informatics Master Science YEAR 2 (SEMESTER 3)

	Area of Specialization	Unit Coded	Unit	Courses Code	Courses	Semester Credit Hours					ECTS Credits	
						Lecture (Courses /TD)	Worksh op	Project	Self Study	Total	ECTS-C	ECTS-U
Third Semester	common	U-3.1	Advanced Topic 1	C-3.1.1	Artificial Intelligence	21	21	0	42	84	3	5
				C-3.1.2	IoT Foundations (and Embedded Systems)	21	10,5	0	25	56	2	
		U-3.2	Advanced Computing 1	C-3.2.1	Advanced Object-Oriented Programming	21	21	0	42	84	3	5
				C-3.2.2	Advanced Database and DBMS	10,5	21	10,5	14	56	2	
		U-3.3	Web and Mobile 1	C-3.3.1	Web Development 3 Symfony/Laravel	10,5	21	10,5	14	56	2	4
				C-3.3.2	Natif Mobile Development (Android/IOS)	10,5	21	10,5	14	56	2	
		U-3.4	Network, System and Information Security	C-3.4.1	System and Network Administration	21	10,5	0	25	56	2	4
				C-3.4.2	Information Systems Security	21	10,5	0	25	56	2	
		U-3.5	Transversal Unit 3	C-3.5.1	Project Management	21	0	0	35	56	2	6
				C-3.5.2	English TOEIC 2	21	0	0	35	56	2	
				C-3.5.3	Personnal Developement and Soft Skills	31,5	0	0	25	56	2	
	Software Engineering	SE-U-3.6	Software Engineering 1	SE-C-3.6.1	Advanced Software Engineering	21	21	0	42	84	3	6
				SE-C-3.6.2	Development Platform (.Net)	21	21	0	42	84	3	
	Big Data and Business Intelligence	BDBI-U-3.6	Big Data and Business Intelligence 1	BDBI-C-3.6.1	Analyse et Reporting	21	21	0	42	84	3	6
				BDBI-C-3.6.2	Business Intelligence	21	21	0	42	84	3	
	Networks and Embedded System	NES-U-3.6	Networks and Embedded System 1	NES-C-3.6.1	Local Area Networks	21	21	0	42	84	3	6
				NES-C-3.6.2	Service Administration	21	21	0	42	84	3	
		Sub Total				**252**	**178,5**	**31,5**	**380**	**840**	**30**	
		Total Semester Workload				**462**						

STUDY PLAN

Informatics Master Science YEAR 2 (SEMESTER 4)

S4

Area of Specialization	Unit Coded	Unit	Courses Code	Courses	Semester Credit Hours					ECTS Credits	
					Lecture (Cr /TD)	Work shop	Project	Self Study	Total	ECTS-C	ECTS-U
common	U-4.1	Advanced Topic 2	C-4.1.1	Virtualization and Cloud Computing	21	10,5	0	24	56	2	4
			C-4.1.2	Advanced IoT Systems & Integration	21	10,5	0	24	56	2	
	U-4.2	Advanced Computing 2	C-4.2.1	Software Architecture and Design Patterns	21	10,5	0	24	56	2	5
			C-4.2.2	Machine Learning and Deep learning	21	21	0	42	84	3	
	U-4.3	Web and Mobile 2	C-4.3.1	Hybride Mobile Development	21	10,5	0	25	56	2	4
			C-4.3.2	Web Developement 4 (ClientSide-ReactJS/Angular)	10,5	21	0	25	56	2	
	U-4.4	Transversal Unit 4	C-4.4.1	Scientific Research	21	0	0	35	56	2	4
			C-4.4.2	Quality Management Systems (QMS)	21	0	0	35	56	2	
	U-4.5	Project Unit 2	C-4.5.1	End of Year Project	0	42	0	42	84	3	3
	U-4.6	Elective Unit 1	C-4.6.1	Elective Course 1	21	10,5	0	25	56	2	4
			C-4.6.2	Elective Course 2	0	21	0	35	56	2	
Software Engineering	SE-U-4.7	Software Engineering 2	SE-C-4.7.1	Collaborative development and E-Commerce	21	21	0	42	84	3	6
			SE-C-4.7.2	Agile and hybrid methods	21	21	0	42	84	3	
Big Data and Business Intelligence	BDBI-U-4.7	Big Data and Business Intelligence 2	BDBI-C-4.7.1	Data Mining 1	21	21	0	42	84	3	6
			BDBI-C-4.7.2	BI Tools	21	21	0	42	84	3	
Networks and Embedded System	NES-U-4.7	Networks and Embedded System 2	NES-C-4.7.1	Routing and Switching	21	21	0	42	84	3	6
			NES-C-4.7.2	Antenna and Propagation	21	21	0	42	84	3	
	Sub Total				220,5	199,5	0	420	840	30	
	Total Semester Workload				420						

Elective Unit 1	Elective Courses
Elective Course 1: Development plateforms	Advanced JAVA Edition (EE / ME)
	ASP CORE
Elective Course 2: Multimedia	UI/UX Design
	Images processing

STUDY PLAN

Informatics Master Science YEAR 3 (SEMESTER 5)

S5

Area of Specialization	Unit Coded	Unit	Courses Code	Courses	Semester Credit Hours					ECTS Credits	
					Lecture (Courses /TD)	Work shop	Project	Self Study	Total	ECTS-C	ECTS-U
common	U-5.1	Advanced Topic 3	C-5.1.1	Cloud-Native Development	0	21	21	14	56	2	5
			C-5.1.2	Big Data	21	21	0	42	84	3	
	U-5.2	Advanced Computing 3	C-5.2.1	Programing paradigms	21	21	0	42	84	3	5
			C-5.2.2	Data warehouse	21	10,5	0	25	56	2	
	U-5.3	Advanced Web	C-5.3.1	Web Developement 5	10,5	21	0	25	56	2	6
			C-5.3.2	Microservices Architecture	21	0	0	35	56	2	
			C-5.3.3	Tests and Quality	10,5	21	10,5	14	56	2	
	U-5.4	Transversal Unit 5	C-5.4.1	Business creation and innovation	21	0	0	34	56	2	4
			C-5.4.2	Technical English 2 (Scientific Writing)	21	0	0	35	56	2	
	U-5.5	Elective Unit 2	C-5.5.1	Elective Course 1	21	0	0	35	56	2	4
			C-5.5.2	Elective Course 2	21	0	0	35	56	2	
Software Engineering	SE-U-5.6	Software Engineering 3	SE-C-5.6.1	ERP / CRM	21	21	0	42	84	3	6
			SE-C-5.6.2	DevOps	21	21	0	42	84	3	
Big Data and Business Intelligence	BDBI-U-5.6	Big Data and Business Intelligence 3	BDBI-C-5.6.1	Data Mining 2	21	21	0	42	84	3	6
			BDBI-C-5.6.2	NoSQL Databases	21	21	0	42	84	3	
Networks and Embedded System	NES-U-5.6	Networks and Embedded System 3	NES-C-5.6.1	Next Generation Network (NGN)	21	21	0	42	84	3	6
			NES-C-5.6.2	Embedded Systems	21	21	0	42	84	3	
	Sub Total				**231**	**157,5**	**31,5**	**420**	**840**	**30**	
	Total Semester Workload				**420**						

Elective Unit 2	Elective Course
Elective Course 1 : Certification	**Preparation for Cloud Certification (AZ-900)**
	Preparation for AI Certification (HCIA-AI)
Elective Course 2 : Advanced security	**Cybersecurity**
	Blockchain

S6	STUDY PLAN				
	Informatics Master Science YEAR 3 (SEMESTER 6)				
	Area of Specialization	**Unit Code**	**Unit**	**Credits Hours**	**ECTS**
	Common	UE-6.1	Graduation Research Project (4-6 months)	600	22
		UE-6.2	Operational Internship 1 (1-2 months)	120	4
		UE-6.3	Technical Internship 2 (1-2 months)	120	4
	TOTAL			**840**	**30**

Notes :

The Average Workload during Internship 30H/Semaine

1 Credit ECTS =28 hours

The Total Semester ECTS Credits = 30

Each elective unit provides two course options from which students can choose. For instance, Elective Unit 1, titled "Development Platforms," offers two elective courses: "JAVA EE" and "ASP CORE".

The validation of the teaching unit U-6.1 takes place in semester 6 and his the corresponding credits (ECTS) are accounted for in semester 6

The validation of the teaching unit U-6.2 takes place in the end of 1st year (semester 2), but the corresponding credits (ECTS) are accounted for in semester 6

The validation of the teaching unit U-6.3 takes place in the end of 2nd year (semester 4), but the corresponding credits (ECTS) are accounted for in semester 6

Department of Computer Science
C-1.1.1-Mathematics for engineers

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Mathematical Foundations 1**		**Course**	**Mathematics for engineers**		**Code**	**C-1.1.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**31h30**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**1**
	Workshop	**--**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**25h00**					
Total workload	**56h00 / Semester**			**Coefficient**		**2**	
				ECTS credit		**2**	
Responsible Teacher	**Mr Imed Ben Salha**		**Involved Teacher(s)**	**Mr Imed Ben Salha**			
Review committee	**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1 & Obj3**			
			PLO	**P7: Analytical and Critical Thinking**			

Prerequisites:	• Basic concept in the study of functions (Parity of functions, periodic functions.) • Concepts of integrals • Calculation of derivatives • Polynomials and rational fraction and decomposition into simple elements
Corequisites:	No corequisites
Learning outcomes	At the end of this subject, students will be able to: 1- Master the techniques of integral calculus. 2- Solve ordinary differential equations. 3- Understand the basic concepts of linear algebra. 4- Apply mathematical skills to problems in electrical engineering.
Content:	**Part 1: Course** **I. Numerical Series and Fourier Series** I.1. Convergence of numerical series I.1.1. Convergence criteria I.2. Fourier series I.2.1. Decomposition into Fourier series I.2.2. Applications to periodic signal analysis **II. Integral Calculation**

II.1. Integration Basics
II.1.1. Definite and indefinite integrals
II.1.2. Convergence and upper and lower limit
I I.2. Integration techniques
II.2.1. Integration by parts
I.2.2. Integration by substitution
II.3. Applications of integrals
II.3.1. Calculation of areas of plane regions
II.3.2. Calculation of volumes of revolution
II.3.3. Calculation of centers of mass

III. Ordinary Differential Equations

III.1. Basic Concepts of Differential Equations
III.1.1. Definitions and terminology
III.1.2. Solutions and solution family
III.2. Solving first order differential equations
III.2.1. Separable differential equations
III.2.2. First order linear differential equations
III.3. Solving second order differential equations
III.3.1. Second-order linear differential equations with constant coefficients
III.3.2. Method of variation of constants

IV. Linear algebra

IV.1. Vectors and vector spaces
IV.1.1. Basic definitions of vectors
IV.1.2. Vector operations
IV.2. Basic Matrix Operations
IV.2.1. Addition and multiplication of matrices
IV.2.2. Matrix inversion
IV.3. Systems of linear equations
IV.3.1. Solving systems of linear equations
IV.3.2. Augmented matrices and Gauss-Jordan methods

Part 2: Directed Work (TD)

I. Numerical Series and Fourier Series - TD

I.1. Exercises on the convergence of numerical series
I.2. Calculation of Fourier series coefficients
I.3. Applications to periodic signal analysis

II. Integral Calculation - TD

II.1. Solving area and volume problems
II.1.1. Calculation of areas of geometric shapes
II.1.2. Calculation of solid volumes
II.2. Calculations of centers of mass for simple objects
II.2.1. Calculating the center of mass of a flat plate
II.2.2. Calculation of the center of mass of a simple solid

	III. Ordinary Differential Equations - TD III.1. Solving Simple Differential Equations Problems III.1.1. Solving first order equations III.1.2. Solving second-order equations **IV. Linear Algebra - TD** IV.1. Manipulation of vectors and matrices IV.1.1. Addition and multiplication of vectors IV.1.2. Advanced matrix operations IV.2. Solving systems of linear equations IV.2.1. Using the Gauss-Jordan method to solve complex systems IV.2.2 . Applications to linear modeling
Teaching and learning method(s):	**Active Learning:** Students are encouraged to actively participate by asking questions, solving problems in class, and collaborating with their peers. **Self-Study:** Students are encouraged to study independently using resources such as textbooks, online tutorials, and supplemental exercises. **Assessments:** Regular assessments, such as exams and assignments, allow students to check their understanding and strengthen their math skills. **Online Support: Online** resources, including explanatory videos and discussion forums, can be made available to help students with their learning.
Teaching technique(s):	Lectures, tutorials, practical exercises, case studies.
Assessment method(s):	A midterm written test (DS) and a final written test (Exam) are planned. ● **Written Exams:** Written exams generally include theoretical questions and practical exercises. Students are assessed on their understanding of mathematical concepts and their ability to solve mathematical problems. ● **Homework:** Homework is assigned regularly to assess students' ability to apply mathematical skills to real-world problems. These assignments can be submitted individually or in groups. ● **Continuous Assessments:** Continuous assessment may include quizzes, online assessments, and periodic tests to check understanding throughout the semester.
Assessment criteria(s) :	1-Understanding of Mathematical Concepts 2-Problem Solving 3-Quality of Work 4-Class Participation 5-Continuous Evaluations 6-Progress Over Time
Assessment measure(s):	Assigning a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated according to the following formula: ● Mid-term written test: 20% ● Final written exam: 80%.
Target skill(s):	1. Skill 1: Mastery of integral calculus. 2. Skill 2: Solving differential equations. 3. Skill 3: Using Linear Algebra

Teaching Manual:	• Course materials • Guided work support
References:	[1] Erwin Kreyszig, “Advanced Engineering Mathematics .” (10th ed., Wiley, 2011). [2] Mary L. Boas, “ Mathematical Methods in the Physical Sciences.” (3rd ed., Wiley, 2010). [3] Dennis G. Zill, “Advanced Engineering Mathematics .” (10th ed., Pearson Education, 2012). [4] Paul Blanchard, Robert L. Devaney, and Glen R. Hall, “Differential Equations.” (6th ed., Brooks/Cole, 2012). [5] William E. Boyce and Richard C. DiPrima, “Elementary Differential Equations and Boundary Value Problems.” (11th ed., Wiley, 2013). [6] Stephen Wolfram, “Mathematica: A System for Doing Mathematics by Computer.” (12th ed., Wolfram Research, 2022).

Department of Computer Science
C-1.1.1-Probability and Statistics

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Mathematical Foundations 1**		**Course**	**Probability and Statistics**		**Code**	**C-1.1.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**1**
	Workshop	**--**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**35h00**					
Total workload		**56h00 / Semester**			**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher		**Mr Imed ben salha**	**Involved Teacher(s)**		**Mr Imed ben salha**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI** **-M.A. PHD. M.A. OMRANI** **-M.A. PHD. WAEL SELLAMI** **- Director of study A. SELMI**	**Objective**		**Obj 1 & Obj 3**		
			PLO		**P7: Analytical and Critical Thinking**		

Prerequisistes:	Calculation of series, calculation of integrals
Corequisites:	Mathematics For the engineer
Learning outcomes	Upon completion of this course, students will be able to: 1. Understand fundamental concepts in probability and statistics. 2. Apply statistical methods to analyze electrical engineering data. 3. Utilize statistical software tools for data analysis. 4. Interpret and communicate results of statistical analyses.
Content:	**Part 1 Courses:** **I. Introduction to Probability and Set Theory** I.1. Probability Fundamentals • Basic Definitions • Probability Axioms I.2. Set Theory and Probability • Sample Spaces and Events • Conditional Probability Independence **II. Random Variables and Probability Distributions** II.1. Random Variables • Discrete and Continuous • Probability Mass Functions (PMFs) and Probability Density Functions (PDFs) II.2. Expected Values and Moments

- Mean and Variance
- Higher Moments

II.3. Transformations of Random Variables

- Functions of Random Variables
- Sum of Random Variables

III. Descriptive Statistics and Data Presentation

III.1. Data Types and Sources

III.2. Data Visualization

III.3. Measures of Central Tendency

- Mean, Median, Mode

III.4. Measures of Dispersion

- Range, Variance, Standard Deviation

IV. Joint Distributions and Independence

IV.1. Joint Probability Distributions

- Joint PMFs and PDFs

IV.2. Independence of Random Variables

- Conditional Independence

V. Discrete Probability Distributions

V.1. Bernoulli and Binomial Distributions

- Probability Functions
- Mean and Variance

V.2. Poisson Distribution

- Poisson PMF
- Applications in Electrical Engineering

V.3. Hypergeometric Distribution

- Probability Function
- Sampling without Replacement

VI. Continuous Probability Distributions

VI.1. Continuous Probability Distributions

- Normal Distribution
- Exponential Distribution

VI.2. Applications of Continuous Distributions

- Modeling and Analysis

VII. Estimation and Hypothesis Testing

VII.1. Point Estimation

- Maximum Likelihood Estimation
- Method of Moments

VII.2. Confidence Intervals

- Confidence Level
- Margin of Error

VII.3. Hypothesis Testing

- Null and Alternative Hypotheses
- Significance Testing

VIII. Linear Regression

VIII.1. Simple Linear Regression

- Linear Regression Model

	● Least Squares Estimation VIII.2. Inference in Regression ● Hypothesis Testing in Regression VIII.3. Model Diagnostics ● Residual Analysis **IX. Statistical Software and Case Studies** IX.1. Statistical Software Tools ● R, SPSS, Excel IX.2. Real-World Case Studies ● Application of Statistical Concepts IX.3. Data Analysis Projects ● Hands-On Experience **Part 2: Tutorials (TD):** **TD1. Problem-Solving Sessions** ● Probability Problems **TD2. Group Discussions** ● Data Analysis Exercises **TD3. Hands-On Data Analysis** ● Real-World Case Studies
Teaching and learning methods:	**Active Learning:** Students are encouraged to actively participate by asking questions, solving problems in class, and collaborating with their peers. **Self-Study:** Students are encouraged to study independently using resources such as textbooks, online tutorials, and supplemental exercises. **Assessments:** Regular assessments, such as exams and assignments, allow students to check their understanding and strengthen their math skills. **Online Support: Online** resources, including explanatory videos and discussion forums, can be made available to help students with their learning.
Teaching techniques:	Lectures, tutorials, practical exercises, case studies.
Assessment methods:	A mid-term written test (DS) and a final written test (Exam) are planned. ● **Written Exams:** Written exams generally include theoretical questions and practical exercises. Students are assessed on their understanding of mathematical concepts and their ability to solve mathematical problems. ● **Homework:** Homework is assigned regularly to assess students' ability to apply mathematical skills to real-world problems. These assignments can be submitted individually or in groups. ● **Continuous Assessments:** Continuous assessment may include quizzes, online assessments, and periodic tests to check understanding throughout the semester.
Assessment criteria:	1-Understanding of Mathematical Concepts 2-Problem Solving 3-Quality of Work 5-Continuous Evaluations
Assessment measure:	Assigning a final average
Criteria for awarding the final grade:	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated according to the following formula: ● Mid-term written test: 20% ● Final written exam: 80%.

Targeted skills:	1. Skill 1: Probability and statistical analysis. 2. Skill 2: Data interpretation and communication. 3. Skill 3: Software proficiency in statistical tools. 4. Skill 4: Problem-solving in electrical engineering
Teaching Manuals:	• Course materials • Guided work support
Bibliographic references:	• Walpole, R.E., Myers, R.H., Myers, S.L., & Ye, K. (2018). "Probability and Statistics for Engineers and Scientists." Pearson. • Devore, J.L. (2015). "Probability and Statistics for Engineering and the Sciences." Cengage Learning. • Montgomery, D.C., & Runger, G.C. (2016). "Applied Statistics and Probability for Engineers." Wiley. • This syllabus provides a comprehensive outline of the course's objectives, content, teaching methods, assessment criteria, and references for "Probability and Statistics for Electrical Engineering" in Semester 1.

Department of Computer Science

C-1.2.1-Algorithms and Data Structures 1

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Fundamental Computing 1**		**Course**	**Algorithms and Data Structures 1**		**Code**	**C-1.2.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Software Engineering**
Type of education	**Lecture**	**21h00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**1**
	Workshop	**21h00**					
	Project	**-**		**Mixed**	**XX**		
	Self Study	**42h00**					
Total workload	**84h00 / Semester**				**Coefficient**		**3**
					ECTS credit		**3**
Responsible Teacher	**Mr Wael Sallami**		**Involved Teacher(s)**	**Mr Wael Sallami**			
Review committee	**-Prof. PHD. MOUNIR ZRIGUI** **-M.A. PHD. M.A. OMRANI** **-M.A. PHD. WAEL SELLAMI** **- Director of study A. SELMI**		**Objective**	**Obj 1 & Obj 3**			
			PLO	**P2 : Implementation Techniques** **P7: Analytical and Critical Thinking**			

Prerequisites:	A strong foundation in mathematics, particularly discrete mathematics and calculus, is beneficial. Many algorithmic concepts are based on mathematical principles. Understanding of fundamental computer science concepts, including computer architecture, operating systems, and software engineering, can be helpful.
Corequisites:	A concurrent course in advanced programming, emphasizing software development principles, design patterns, and best practices, to enhance students' practical programming skills for implementing and optimizing algorithms and data structures.
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Analyze a given problem and define the algorithm translating the solution of the problem in a rigorous and optimized way. 2. Become familiar with recursion. 3. Understand the usefulness of data structures in the development of algorithms. 4. Master pointers and the principle of dynamic allocation. 5. Understand lists (simple, circular, doubly linked)
Content:	**Chapter 1: Recursion** 1. Recurrence 2. Definition of recursion

3. Mechanism of operation of recursion

Tutorial/ Practical Work:

TD1: Compare iterative and recursive programming approaches, understand the saving of calls in the execution memory of a program, apply recursion to manipulate character strings, master sorting algorithms.

TP1: Practice recursive programming to solve complex problems.

Chapter 2: Tables

1. Usefulness of the tables
2. Notation and use of tables
3. Two-dimensional arrays

Tutorial/ Practical Work:

TD2: Understand the importance of arrays in the structuring of programs, know the different processes that can be performed on one- or two-dimensional arrays.

TP2: Mastering the manipulation of tables and matrices, implementing sorting algorithms (rapid, merging, etc.).

Chapter 3: Sorting

1. What is sorting, and why do we do it?
2. Properties of a sorting algorithm
3. Insertion sort
4. Sort by selection
5. Bubble Sort
6. Merge Sort
7. Quick Sort
8. Comparison between sorting algorithms

Chapter 4: Data Structures

1. Introduction
2. Notion of data structure
3. Manipulation of record/structure type variables
4. Nesting structures
5. Exercise

Tutorial/ Practical Work:

TD3: Compare structured types and elementary types with real examples, define the constituent elements of a data structure, access the various fields of a data structure.

TP3: Define the data structures appropriate to the proposed problems, manipulate access to the fields of the structures created.

Chapter 5: Pointers

1. Introduction
2. Concept of the pointer
3. Declaration of a pointer
4. Dynamic Memory Allocation
5. Dynamic Memory Release

Tutorial/ Practical Work:

TD4: Define the difference between a value and a memory address, access data in computer memory using pointers, manipulate arrays using pointers.

TP4: Implement pointers to write more compact programs for a problem.

Chapter 6: Lists

1. Introduction
2. Linked lists
3. Circular Linked Lists
4. Doubly linked lists

Tutorial/ Practical Work:

TD5: Develop functions to insert or delete an element in a list, study the differences between the types of lists (simple, doubly circular and chained), know how to use the recursive structure of lists to design an algorithm.

TP5: Implement the different types of linked lists (simply linked, circular and doubly linked), exploit recursive structures with lists to design advanced algorithms.

Project :

Project Title: Dynamic Contact Management System

Project Description:

Create a dynamic contact management system that uses a doubly linked list to store and manage a list of contacts. This project will not only help you practice working with linked lists but also provide a practical application for data structure usage

Self-study:

1. Research
   - Study programming fundamentals and its relevance in engineering.
   - Collect relevant resources and references.
2. Proposal
   - Choose an engineering problem.
   - Explain how to resolve complex algorithms.
4. Project Report
   - Create a report summarizing the project.
   - Reflect on the experience and insights gained.
5. Presentation (Optional)
   - Share project findings with the class if desired.
6. Doing Exercise and QCM published in easylearn.ovh platform.

Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2 and 3** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes, 4 and 5**
Assessment criteria(s) :	During the midterm exam, students should be able to manipulate arrays and matrices. Thus, they should master the operating mechanism of recursion as well as the constituent elements of a data structure. By the end of the final exam, students should be able to develop algorithms based on singly and doubly linked lists.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	● Master programming in C language ● Have notions of recursion and computational complexity ● Know and correctly use basic data structures (lists, stacks, queues, etc.) ● Understand basic data structures (trees, binary search trees): study of their abstract properties, their concrete representations, their application and the main algorithms that manipulate them.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Michael Griffiths, Algorithmique et programmation, Hernes, 1992, ISBN: 2-86601-323-9. [2] Sébastien Rohaut, « Algorithmique et Techniques fondamentales de programmation », Edition Eni, 2007. [3] Patrice Lignelet et Jean Girerd, « Algorithmique. Méthodes et modèles », Paris : Masson, 1985. Zitouni Baghdadi, « Algorithmique et structures de données », Centre de Publication Universitaire, 2003

Department of Computer Science
C-1.2.2-Automata Theory

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Fundamental Computing 1**		**Course**	**Automata Theory**		**Code**	**C-1.2.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**1**
	Workshop	**--**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**34h00**					
Total workload		**56h00 / Semester**			**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher		**Mr Mohsen Maraoui**	**Involved Teacher(s)**		**Mr Mohsen Maraoui**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**	**Objective**		**Obj 1 & Obj3**		
			PLO		**P2 : Implementation Techniques P7: Analytical and Critical Thinking**		

Prerequisites:	Discrete Mathematics Formal Logic Basic Programming Knowledge Data Structures Algorithms Mathematical Proof Techniques
Corequisites:	None
Learning outcomes	Upon completion of this teaching module, the student will be able to: **1.** Understand the fundamental concepts of automata theory. **2.** Analyze and design automata. **3.** Identify regular, context-free, and context-sensitive languages. **4.** Apply automata in practical applications. **5.** Demonstrate the relationship between automata and other areas of theoretical computer science. **6.** Analyze the complexity of automata. **7.** Solve advanced problems using automata theory. **8.** Communicate effectively about complex concepts.
Content:	**Chapter I: Fundamentals of Words and Languages** 1. Alphabetic Systems

2. Lexical Structures
3. Language Frameworks
4. Language Operations

Chapter II: Exploration of Regular Languages, Finite Automata, and Grammatical Structures

1. Characteristics of Regular Languages
2. An Introduction to Regular Expressions
3. Techniques for Simplifying Regular Expressions
4. Finite-State Machines and Automata
   a. Understanding Automata: A Definition
   b. Transition Tables in Automata
   c. Graphical Representation: State Diagrams
   d. Key Principles: Configuration and Derivation
   e. Recognizing Languages via Automata
   f. The Realm of Non-Deterministic Finite Automata
   g. Equivalence in Automaton Systems
   h. Operational Dynamics in Automata
5. Grammatical Systems and Structures
   a. The Basis of Grammar: A Definition
   b. Generating Languages Through Grammatical Structures
   c. The Construct of Derivation Trees
   d. Diving into Regular Grammars
   e. The Intersection of Regular Grammars and Finite Automata
   f. Correlations: Regular Grammars and Regular Expressions
   g. The Pumping Lemma Explained

Chapter III: The World of Pushdown Automata and Algebraic Grammar Systems

1. Pushdown Machines: An Overview
   a. Delving into Pushdown Machines: A Definition
   b. Theoretical Foundations: Configuration and Derivation
   c. Navigating Non-Deterministic Pushdown Automata
2. Context-Free Grammatical Systems
   a. Understanding Context-Free Grammars
   b. The Ambiguities in Grammatical Structures
   c. The Implications of the Pumping Lemma
   d. The Inherent Properties of Context-Free Grammars
   e. Drawing Parallels: Context-Free Grammar vs. Pushdown Automata

Chapter IV: Turing Machines and Their Relationship with Recursive Grammars

1. Turing Machines: Unveiling the Definition
2. The Principle of Determinism in Computation

Project: Designing and Analyzing Automata Systems
Objective:
To deepen understanding of automata concepts by designing automata systems and analyzing their capabilities, limitations, and potential applications.

Tasks:
Regular Language Recognizer:

Design a deterministic finite automaton (DFA) that recognizes strings over the alphabet {a, b} which have an even number of a's.
Extend the DFA to recognize strings that have an even number of a's and odd number of b's.
Convert the DFA to its equivalent regular expression.

Simple Language Parser:

Design a pushdown automaton (PDA) that recognizes the balanced parentheses language (e.g., “()”, “(())”, “()()”).
Extend the PDA to recognize a language where every opening bracket ‘(‘ is followed by a closing bracket ‘)’ and then an 'a' (e.g., “(a)”, “((a)a)a”).

Automata Application:
Imagine an application or a system where automata play a crucial role (e.g., text parsing, lexical analysis, pattern matching). Briefly describe the application and how automata would be applied.
Comparative Analysis:

Compare the expressive power of DFAs, PDAs, and Turing machines. In which scenarios would one be preferred over the other? Provide examples.

Contextual Analysis:
Pick a real-world application (e.g., a simple text editor, search engine query processing). Describe how context-free or context-sensitive languages play a role in that application. Design a basic grammar for a segment of the application.

Reflection:
Write a brief report reflecting on the challenges faced during the project, insights gained about the complexity and versatility of automata, and potential real-world applications of the studied concepts.

Deliverables:
DFA diagrams and corresponding regular expressions.
PDA diagrams.
Description and analysis of the imagined application.
Comparative analysis report.
Contextual analysis with designed grammar.
Reflection report.

Evaluation Criteria:
Correctness and efficiency of the designed automata.
Depth of understanding demonstrated in the application, comparative, and contextual analyses.
Clarity and depth of the reflection report.
Ability to effectively communicate complex concepts.

Self-study and Research Task: Exploration of Modern Applications of Automata Theory in Computer Science

Objective:
To understand the contemporary significance of automata theory by exploring its applications in modern computing domains and analyzing its impact on problem-solving within those domains.
References:

	[1] Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2022). Introduction to Automata Theory, Languages, and Computation (3rd ed.). Pearson. Natural Language Processing: [2] Shallit, J. (2008). A second course in formal languages and automata theory. Cambridge University Press. Sakarovitch, J. (2009). Elements of automata theory. Cambridge University Press. [3] Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2016). Compilers: Principles, techniques, & tools (2nd ed.). Pearson. [4] Navarro, G., & Raffinot, M. (2002). Flexible pattern matching in strings: practical on-line search algorithms for texts and biological sequences. Cambridge University Press.
Teaching and learning method(s):	Face to face 34 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2, 3 and 4.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 5, 6, 7 and 8.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain foundational concepts, demonstrate a solid grasp on the analysis of automata and linguistic structures, and illustrate the basic principles behind the design and development of automata systems. By the end of the final exam, students should be able to evaluate computational complexities, exhibit proficiency in problem-solving, and communicate their thoughts effectively, showcasing a clear articulation of ideas and concepts in automata theory.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Mastery in understanding and applying foundational automata concepts. 2. Expertise in analyzing and designing specific automata structures. 3. Ability to evaluate and address issues of computational complexity. 4. Skillset for effectively solving complex problems using automata principles. 5. Proficient communication skills, particularly in articulating intricate automata concepts. 6. Aptitude for integrating automata theory with other theoretical computer science domains.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks

References:	[1] Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2006). Introduction to Automata Theory, Languages, and Computation (3rd ed.). Pearson. [2] Sipser, M. (2012). Introduction to the Theory of Computation (3rd ed.). Cengage Learning. [2] Lin, Z., & Radoi, C. (2016). Formal Languages and Automata Theory. [3] Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006). Compilers: Principles, techniques, & tools (2nd ed.). Pearson. [4] Roche, E., & Schabes, Y. (Eds.). (1997). Finite-state language processing. MIT press. [5] Shallit, J. (2008). A second course in formal languages and automata theory. Cambridge University Press.

Department of Computer Science
C-1.3.1-Computer Architecture

Last Update Date: May 10, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Architecture and Operating System</td><td>Course</td><td colspan="2">Computer Architecture</td><td>Code</td><td>C-1.3.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">1</td></tr>
<tr><td>Workshop</td><td>21h00</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>42h00</td><td colspan="2"></td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Yahia Salah</td><td>Involved Teacher(s)</td><td colspan="2">Mr Yahia Salah</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj1</td></tr>
<tr><td>PLO</td><td colspan="2">P1 : Systems Specification and Design Methodologies
P5: Network and Information Systems Management</td></tr>
<tr><td colspan="2">Prerequisites:</td><td colspan="6">Logic Systems</td></tr>
<tr><td colspan="2">Corequisites:</td><td colspan="6">Operating Systems</td></tr>
<tr><td colspan="2">Learning outcomes</td><td colspan="6">Upon completion of this teaching module, the student will be able to:

1. Understand and recount the historical evolution of computer systems and architectures.
2. Analyze and describe the foundational principles of processor design, including ALU, control units, and instruction sets.
3. Understand the basics of pipelining, including stages, hazards, and the concepts of superscalar and VLIW architectures.
4. Appraise the principles of memory hierarchy, cache design, and cache coherence for optimal system performance.
5. Understand the opportunities and challenges of parallelism, including the design and implementation of multicore systems and hardware threads.
6. Model, simulate, and benchmark the performance of diverse computer architectures using appropriate tools.</td></tr>
</table>

	7. Understand the role and basic applications of hardware description languages, such as VHDL and Verilog, in architectural prototyping.
Content:	**Chapter 1: Introduction to Computer Architecture** 1.1 Historical Evolution of Computer Systems 1.2 Basic Architectural Components and Their Interactions 1.3 Von Neumann Architecture vs. Harvard Architecture **Chapter 2: Fundamentals of Processor Design** 2.1 Basic CPU Operations 2.2 ALU and Control Unit Design 2.3 Instruction Sets and Assembly Language **Chapter 3: Pipeline and Advanced Architectures** 3.1 Basics of Pipelining: Stages and Hazards 3.2 Superscalar and VLIW Architectures 3.3 Dynamic Scheduling and Out-of-Order Execution **Chapter 4: Memory Hierarchy and Cache Optimization** 4.1 Principles of Memory Hierarchy 4.2 Cache Design and Mapping Techniques 4.3 Cache Coherence and Memory Consistency **Chapter 5: Parallel Architectures and Multicore Systems** 5.1 Parallelism: Opportunities and Challenges 5.2 SIMD and MIMD Architectures 5.3 Multicore Processors and Hardware Threads **Chapter 6: Emerging Technologies and Future Architectures** 6.1 Role of Non-Volatile Memories 6.2 Quantum Computing and Optical Computing 6.3 Reconfigurable Architectures **Chapter 7: Performance Modeling and Simulation** 7.1 Metrics for Performance Evaluation 7.2 Benchmarking and Analytical Modeling 7.3 Computer Architecture Simulation Tools **Chapter 9: Hardware/Software Interaction** 9.1 Firmware, BIOS, and Microcode 9.2 Compilers and Architecture Dependency 9.3 OS Interaction: System Calls and Virtual Memory **Chapter 10: Advanced Topics and Case Studies** 10.1 Power and Energy-Efficient Architectures 10.2 Case Study: ARM vs. x86 Architectures 10.3 Current Trends and Future Predictions in Computer Architecture

	Self-study and Research Topic: "Processing Algorithms based on Emerging Non-Volatile Memory (NVM) Technologies" **Description**: Emerging non-volatile memory (NVM) technologies, such as Phase-Change Memory (PCM), Spin-Transfer Torque RAM (STT-RAM), and Resistive RAM (ReRAM), present unique characteristics distinct from traditional memory. This research aims to investigate how processing algorithms can be optimized to exploit the advantages and mitigate the challenges of these new memory technologies, with a particular emphasis on improving processing speed, energy efficiency, and memory resilience. **References**: [1] Qureshi, M. K., & Patt, Y. N. (2016). Symbiotic relationships between DRAM memory systems and emerging non-volatile memories. Proceedings of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS XIII). [2] Zhou, P., Zhao, B., & Yang, J. (2009). Energy reduction for STT-RAM using early write termination. Proceedings of the International Conference on Computer-Aided Design (ICCAD '09). [3] Meza, J., Luo, Y., Khan, S., Zhao, J., Mutlu, O., & Xie, Y. (2015). Efficiently managing hybrid main memories with a page-utility driven approach. Proceedings of the 31st Symposium on Mass Storage Systems and Technologies (MSST '15).
Teaching and learning method(s):	Face to face 42 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,2 and 3.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 4, 5, 6, and 7.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain the foundational concepts of computer architecture, basic processor design, and the principles of memory hierarchy and pipelining. By the end of the final exam, students should be able to analyze complex hardware/software interactions, evaluate emerging technologies in architecture, and model the performance of diverse computer architectures.
Assessment measure(s):	Assigning of a final average

Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Mastery in analyzing and designing computer systems at the architectural level. 2. Proficiency in optimizing performance across various computer architectures. 3. Understanding the nuances of memory hierarchy and processus.. 4. Capability in emerging hardware technologies. 5. A strong understanding of the hardware and software layers and their impacts on system performance.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] : Stallings, W. (2015). Computer Organization and Architecture: Designing for Performance. 10th Edition. Pearson. [2]Patterson, D. A., & Hennessy, J. L. (2013). Computer Organization and Design: The Hardware/Software Interface. 5th Edition. Morgan Kaufmann. [3] Hennessy, J. L., & Patterson, D. A. (2019). Computer Architecture: A Quantitative Approach (6th ed.). Elsevier. [4] Tannenbaum, A. S. (2013). Structured Computer Organization (6th ed.). Prentice Hall. [5] Stallings, W. (2016). Computer Organization and Architecture (10th ed.). Pearson.

Department of Computer Science

C-1.3.2-Operating Systems & Workshop

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Architecture and Operating System**		**Course**	**Operating Systems & Workshop**		**Code**	**C-1.3.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**1**
	Workshop	**42h00**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**21h00**					
Total workload	**84h00 / Semester**				**Coefficient**		**3**
					ECTS credit		**3**
Responsible Teacher	**Mr Yahia Salah**		**Involved Teacher(s)**	**Mr Yahia Salah**			
Review committee	**-Prof. PHD. MOUNIR ZRIGUI** **-M.A. PHD. M.A. OMRANI** **-M.A. PHD. WAEL SELLAMI** **- Director of study A. SELMI**		**Objective**	**Obj 1**			
			PLO	**P5: Network and Information Systems Management**			

Prerequisites:	Fundamentals of Computer Systems Basics of Computer Architecture Programming Fundamentals
Corequisites:	None
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Understand the core concepts and principles of operating systems. 2. Design and implement basic OS components and algorithms. 3. Analyze and manage system processes, memory, and storage. 4. Configure and administer Unix-based systems. 5. Develop proficiency in shell scripting and Unix system programming. 6. Understand the deadlock phenomena in concurrent computing, including its causes, effects, and various forms of detection and resolution strategies.
Content:	**Chapter 1:** Basic Notions 1. Introduction to Operating Systems 2. Roles and functions of an OS 3. Types of Operating Systems **Chapter 2: Processes and Threads** 1. Process states and life cycle 2. Context switches

3. Threads and multi-threading

Chapter 3: Memory Management

1. Physical and Virtual Memory
2. Paging and Segmentation
3. Memory Allocation Strategies

Chapter 4: Storage Management

1. File systems
2. Disk Scheduling
3. I/O Management

Chapter 5: Unix Fundamentals

1. Introduction to Unix
2. Basic Unix commands
3. Unix File System

Chapter 6: Unix Workshop

1. Shell scripting basics
2. System programming in Unix
3. Advanced Unix utilities

Chapter 7: Concurrency and Synchronization

1. Critical sections and mutual exclusion
2. Deadlocks
3. Semaphores and Monitors

Workshops:

1. Setting up a Virtual Machine Environment
2. Process Creation and Monitoring on Unix
3. Memory Allocation Simulations
4. Getting Started with Unix Commands
5. Shell Scripting Challenges
6. Deadlock Detection and Resolution on Unix

Project & Research:

Research Task: Exploration of Modern Operating System Vulnerabilities and Unix-based Mitigation Techniques

Objective:
To understand and evaluate current vulnerabilities in modern operating systems and explore Unix-based techniques and tools to mitigate such vulnerabilities.

Description:
With the evolution of operating systems, new vulnerabilities often emerge, challenging the security and robustness of both personal and enterprise systems. This research will delve deep into understanding such vulnerabilities, their root causes, potential impacts, and how Unix-based systems or tools can help in countering them.

	Tasks: Identify and Document Vulnerabilities: Unix-based Mitigation Techniques: Hands-on Analysis: Future-Proofing: Present Your Findings:
Teaching and learning method(s):	Face to face 42 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2 and 3** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 4 5 and 6.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain the foundational concepts of operating systems, detail the workings of processes, thread and memory management. By the end of the final exam, students should be able to demonstrate proficiency with basic Unix functionalities, design and implement concurrency and synchronization mechanisms, ensure security and protection measures within OS environments, delve into advanced OS concepts, and proficiently use OS diagnostic tools for performance monitoring.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. OS Design Expertise: Ability to understand and design operating system components. 2. Unix Mastery: Proficiency in Unix administration, shell scripting, and system programming. 3. Concurrency Control: Skilled in handling concurrent processes and ensuring synchronization. 4. Security Acumen: Awareness of OS-level security and protection mechanisms. 5. Diagnostic Skills: Capability to utilize system tools for monitoring and troubleshooting. 6. Adaptive Learning: Ability to stay updated with emerging trends in OS and platform-specific nuances.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks

References:	[1] Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts. 10th Edition. Wiley. [2] Stevens, W. R., & Rago, S. A. (2013). Advanced Programming in the UNIX Environment. 3rd Edition. Addison-Wesley. [3] Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems. 4th Edition. Pearson. [4] UNIX and Linux System Administration Handbook by Evi Nemeth, Garth Snyder, Trent R. Hein, and Ben Whaley. [5] Operating System Concepts by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne. [6] The Linux Programming Interface: A Linux and UNIX System Programming Handbook by Michael Kerrisk.

Department of Computer Science

C-1.4.1-Python Programming Workshop

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Web and Programming</td><td>Course</td><td colspan="2">Python Programming Workshop</td><td>Code</td><td>C-1.4.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>-</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td rowspan="2">--</td><td rowspan="4">Semester</td><td rowspan="4">1</td></tr>
<tr><td>Workshop</td><td>21h 00</td></tr>
<tr><td>Project</td><td>-</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Self Study</td><td>35h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Wael Sellami</td><td>Involved Teacher(s)</td><td colspan="3">Mr Wael Sellami</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="3">P2: Implementation Techniques</td></tr>
<tr><td colspan="2"><u>Prerequisites:</u></td><td colspan="6">A strong foundation in mathematics, particularly discrete mathematics and calculus, is beneficial. Many algorithmic concepts are based on mathematical principles.
Understanding of fundamental computer science concepts, including computer architecture, operating systems, and software engineering, can be helpful.</td></tr>
<tr><td colspan="2"><u>Corequisites:</u></td><td colspan="6">Software development principles, design patterns, and best practices, to enhance students' practical programming skills for implementing python programs</td></tr>
<tr><td colspan="2"><u>Learning outcomes</u></td><td colspan="6">Upon completion of this teaching module, the student will be able to:
− Write Python programs to solve complex problems.
− Utilize Python libraries for data analysis, scientific computing, and visualization.
− Develop object-oriented Python applications.
− Handle file I/O and perform advanced operations using Python.
− Apply Python for real-world applications and demonstrate their skills through a final project.</td></tr>
<tr><td colspan="2"><u>Content:</u></td><td colspan="6">Chapter 1: Introduction to Python
1. Basics of Python
2. Data types and variables
3. Control flow (if-else, loops)
4. Functions and modules
5. Input/output
Practical Work:
TP1: Write a Python program that calculates and displays basic mathematical operations.</td></tr>
</table>

Chapter 2: Data Structures and Libraries

1. Lists, tuples, and dictionaries
2. NumPy for numerical computing
3. Pandas for data manipulation
4. Matplotlib for data visualization

Practical Work:

TP2: Implement a program that utilizes lists, tuples, and dictionaries for data storage and manipulation.

Chapter 3: Object-Oriented Programming

1. Classes and objects
2. Inheritance and polymorphism
3. Encapsulation and abstraction
4. Exception handling

Practical Work:

TP3: Design and implement a simple class with methods and attributes.

Chapter 4: Advanced Python Programming

1. File handling and I/O operations
2. Regular expressions
3. Functional programming in Python
4. Concurrency with threads and multiprocessing

Practical Work:

TP4: Read and write data to files, demonstrating various file handling operations.

Chapter 5: Applications and Project Work

1. Scientific computing with SciPy
2. Data analysis and visualization projects
3. GUI development with Tkinter or PyQt
4. Web development with Flask or Django

Practical Work:

TP5: Solve scientific computing problems using NumPy and SciPy.

Project :

Project Title: Data Analysis and Visualization Tool

Project Description:

The project aims to develop a Python-based data analysis and visualization tool that allows users to load, analyze, and visualize datasets efficiently. The tool will be designed to cater to various industries and domains, enabling users to gain insights from data in a user-friendly and interactive manner.

Self-study:

1. Research
   - Study programming fundamentals and its relevance in engineering.
   - Collect relevant resources and references.
2. Proposal
   - Choose an engineering problem.
   - Explain how to implement python programs.
4. Project Report
   - Create a report summarizing the project.
   - Reflect on the experience and insights gained.
5. Presentation (Optional)

	Share project findings with the class if desired. 6. Doing Exercise and QCM published in easylearn.ovh platform.
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1 and 2** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes ., 3, 4 and 5**
Assessment criteria(s) :	During the midterm exam, students should be able to succinctly define and explain fundamental Python concepts, including variables, data structures, control flow, functions, modules, object-oriented programming, file handling, regular expressions, advanced Python features, and their applications in data analysis and scientific computing. By the end of the final exam, students should be able to demonstrate a comprehensive understanding of Python, showcasing proficiency in programming, data analysis, and software development, and effectively applying Python to solve complex real-world problems.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	**1.** Demonstrate proficiency in Python programming, encompassing a deep understanding of the language's core syntax, data structures, and control flow. **2.** Showcase competence in data analysis and manipulation, effectively using libraries such as Pandas and NumPy for data handling and statistical operations. **3.** Exhibit expertise in object-oriented programming, with the ability to design and implement classes, leverage inheritance and polymorphism, and encapsulate data for real-world modeling. **4.** Display skills in software development, including the creation of graphical user interfaces (GUI) and web applications using Python frameworks such as Tkinter, Flask, or Django, reflecting their comprehension of the software development lifecycle.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks

References:	[1] Python Official Documentation: Python Software Foundation, Python Documentation. [2] NumPy Official Documentation: NumPy Developers, NumPy Documentation. [3] Pandas Official Documentation: Pandas Development Team, Pandas Documentation. [4] Mark Lutz, Learning Python, 2013, O'Reilly Media.

Department of Computer Science
C-1.4.2-Web Development 1

Last Update Date: May 17, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Web and Programming</td><td>Course</td><td colspan="2">Web Development 1</td><td>Code</td><td>C-1.4.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">1</td></tr>
<tr><td>Workshop</td><td>21h 00</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>10,5h</td></tr>
<tr><td>Self Study</td><td>32h00</td><td colspan="2"></td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Mohamed Amine Omrani</td><td>Involved Teacher(s)</td><td colspan="3">Mr Allani Abderrahim</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="3">P1 : Systems Specification and Design Methodologies
P2: Implementation Techniques
P4: Web and Software Development</td></tr>
<tr><td>Prerequisites:</td><td colspan="7">General knowledge of basic concepts related to Internet</td></tr>
<tr><td>Corequisites:</td><td colspan="7">Algorithms and Data Structures</td></tr>
<tr><td>Learning outcomes</td><td colspan="7">Upon completion of this teaching module, the student will be able to:

1. Plan and design a website using appropriate languages.
2. Master client-side web programming languages: HTML5, CSS3.
3. Utilize the Bootstrap Framework for Developing Responsive Websites.
4. Integrate JavaScript features and advantages.
5. Properly Utilize the jQuery Library.
6. Natively create dynamic, interactive and responsive web pages from A to Z.</td></tr>
<tr><td>Content:</td><td colspan="7">Chapter 1: Basic concepts
1. Internet services
2. Protocols
3. Web working principles
4. Client-side languages vs server-side languages
5. Concepts of CMS and framework

Chapter 2: HTML 5</td></tr>
</table>

1. HTML5 Document Structure
2. Basic tags
3. Lists
4. Hypertext links, images and relative paths
5. Tables
6. Forms
7. New HTML5 features
8. HTML5 and multimedia
9. HTML5 Advanced Topics

Practical Work:

TP1: HTML: Basics, images, links and relative paths.
TP2: HTML: Tables, forms, new HTML5 features

Chapter 3: CSS 3

1. The CSS rule structure
2. Declaration
3. Media Queries
4. Selectors
5. CSS values and units
6. Colors, texts, sizes, lists, tables
7. Blocks
8. CSS Box Model
9. Margins
10. Borders
11. Outer and Inner Spacing
12. Background
13. Sizing items in CSS
14. CSS Layouts: Grid vs Flexbox vs Float vs Position
15. Animations
16. Responsive web design

Practical Work:

TP3: CSS Integration: colors, characters, text, background, etc.
TP4: CSS Layouts
TP5: Responsive web design

Chapter 4: Bootstrap framework

1. How to Use The Bootstrap Framework
2. Overview of Bootstrap CSS
3. Bootstrap Layout Components
4. Overview of Bootstrap Plugins
5. Other Bootstrap visual components
6. Responsive web design

Practical Work:

TP6: Bootstrap Practice Project.

Chapter 5: Javascript Fundamentals

1. Introduction
2. Javascript syntax
3. Operators
4. Loops and iteration in JavaScript
5. JavaScript Tables & Data Grids
6. Functions
7. Predefined Core Objects and Functions

	8. Events 9. DOM 10. JavaScript Arrow Function **Practical Work:** **TP7:** JavaScript basics **Chapter 6: Advanced Javascript** 11. Object-Oriented Programming in JavaScript 12. Javascript Object Notation (JSON) 13. Regular expressions 14. File management 15. Fetch API - How to Make GET and POST Requests 16. Error Handling in JavaScript 17. JavaScript Geolocation API **Practical Work:** **TP8:** javascript - advanced concepts **Chapter 7: jQuery** 1. Difference Between JavaScript and jQuery 2. jQuery Methods 3. jQuery DOM Manipulation 4. jQuery Event Methods 5. jQuery and Ajax **Practical Work:** **TP9:** Getting Started with jQuery —---------- **Project 1 :** - - *"Country Game" (HTML + CSS +* Javascript + jQuery) *We intend to guide students in the development of a little game called "Country Game".* *A map is displayed, it represents several countries; a number is associated with each country. On the left side of the page, we find these numbers with a scrolling list of country names next to them. It is therefore a question of associating each number with the name of the country which corresponds to it.* **Project 2 :** - - *"snake game" (Canvas + CSS +* Javascript + jQuery) In this game, the player controls a long, thin creature, resembling a snake, which roams around on a bordered plane, picking up some item, trying to avoid hitting the edges of the playing area or its own tail. —---------- **Self-study:** - - Doing Exercise and QCM published in easylearn.ovh platform.
Teaching and learning method(s):	Face to face (face-to-face) 53 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning and self study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work (Integrated course, Practical work, Debate , Conduct project).

Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2. and 3.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes ., 4, 5, 6 and 7.**
Assessment criteria(s) :	During the midterm exam, students should be able to create attractive web pages. By the end of the final exam, students should be able create responsive interactive Web pages.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Ability to develop attractive and responsive web page using CSS and Bootstrap 2. Ability to develop interactive web page using JavaScript and jQuery
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks ● Polytech EasyLearn Plateform
References:	[1] HTML, CSS & JavaScript for Complete Beginners: A Step by Step Guide to Learning HTML5, CSS3 and the JavaScript Programming Language Paperback – by Ikram Hawramani, December 1, 2018 [2] Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming, Marijn Haverbeke – 4 décembre 2018. [3] HTML5 and CSS3 All-in-One For Dummies (English Edition), Andy Harris.

Department of Computer Science
C-1.5.1-Data Transmission

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Network and Data Transmission</td><td>Course</td><td colspan="2">Data Transmission</td><td>Code</td><td>C-1.5.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">1</td></tr>
<tr><td>Workshop</td><td>21h00</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>-</td><td rowspan="2">Coefficient</td><td rowspan="2">3</td></tr>
<tr><td>Self Study</td><td>42h00</td><td colspan="2">Total workload</td><td colspan="2">84h00 / Semester</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mme ElHEM Blel</td><td>Involved Teacher(s)</td><td colspan="2">Mme ElHEM Blel</td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="3">P5: Network and Information Systems Management</td></tr>
<tr><td colspan="2">Prerequisites:</td><td colspan="6">• Foundational Networking Knowledge
• Operating Systems Knowledge
• Computer Architecture</td></tr>
<tr><td colspan="2">Corequisites:</td><td colspan="6">• Advanced Data Analysis and Statistics:
• Network Simulation and Modeling</td></tr>
<tr><td colspan="2">Learning outcomes</td><td colspan="6">Upon completion of this teaching module, the student will be able to:

1. Analyze and understand the principles of data transmission.
2. Evaluate and select appropriate data transmission technologies.
3. Design and implement effective data transmission solutions.
4. Troubleshoot data transmission issues and optimize network performance
5. Adapt to emerging trends and technologies in the field.
6. Integrate Wireless and Mobile Data Transmission:
7. Implement Data Transmission Security Measures:</td></tr>
<tr><td colspan="2">Content:</td><td colspan="6">Chapter 1: Fundamentals of Data Transmission
1. Data communication models and concepts
2. Signals and modulation techniques
3. Data encoding and transmission media
4. Signal quality, noise, and error detection/correction
Chapter 2: Data Link Layer Protocols
1. Ethernet and IEEE 802.3 standards</td></tr>
</table>

	2. MAC addressing and frame formats 3. LAN switching and bridging 4. VLANs and advanced switching technologies **Chapter 3: WAN technologies and topologies** 1. Point-to-Point Protocol (PPP) 2. Frame Relay and ATM 3. MPLS and SD-WAN technologies **Chapter 4: WAN technologies** 1. IP addressing and subnetting 2. Routing algorithms and protocols (e.g., OSPF, BGP) 3. IPv6 and its implementation 4. Network virtualization and overlay networks **Chapter 5: WAN technologies** 1. Transport layer protocols (TCP, UDP) 2. Flow control and congestion management 3. Advanced transport layer features 4. Quality of Service (QoS) in transport **Chapter 6: Wireless and Mobile Data Transmission** 1. Wireless communication fundamentals 2. Cellular networks (2G, 3G, 4G, 5G) 3. Mobile IP and mobile network optimization 4. Emerging trends in wireless data transmission **chapter 7: Data Transmission Security** 1. Security threats in data transmission 2. Cryptographic protocols and techniques 3. Virtual Private Networks (VPNs) and secure data transmission 4. Intrusion detection and prevention in data transmission —---------- **Project :** Students undertake a substantial project that involves the design, implementation, and evaluation of a data transmission solution or addresses a real-world data transmission challenge. —---------- **Self-study: Objective:** Deepen your understanding of data link layer: ❑ Study data link layer protocols such as Ethernet and IEEE 802.3 standards. ❑ Experiment with MAC addressing, frame formats, and LAN switching using practical exercises.
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.

Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,2. and 3** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes , 4,5,6 and 7**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain provides an in-depth exploration of data transmission technologies, protocols, and practices. By the end of the final exam, students should be able covers both theoretical foundations and practical aspects of data communication and transmission, with a focus on advanced concepts and emerging trends..
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Advanced Data Transmission Protocol Analysis 2. Data Transmission System Design:
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] EndNote, Zotero, or Mendeley to organize and manage references for research projects. [2] Skill in using these tools to generate citations and bibliographies automatically.

Department of Computer Science
C-1.5.2-Computing Networks

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Network and Data Transmission</td><td>Course</td><td colspan="2">Computing Networks</td><td>Code</td><td>C-1.5.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="4">Semester</td><td rowspan="4">1</td></tr>
<tr><td>Workshop</td><td>21h00</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>-</td></tr>
<tr><td>Self Study</td><td>42h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mme Elhem Blel</td><td>Involved Teacher(s)</td><td colspan="3">Mme Elhem Blel</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="3">P5: Network and Information Systems Management</td></tr>
<tr><td colspan="2">Prerequisites:</td><td colspan="6">● fundamental networking and computer science concepts
● data structures, and computer architecture</td></tr>
<tr><td colspan="2">Corequisites:</td><td colspan="6">● Network Simulation and Modeling
● Advanced Security and Privacy in Computing:</td></tr>
<tr><td colspan="2">Learning outcomes</td><td colspan="6">Upon completion of this teaching module, the student will be able to:

1. Analyze and design complex network architectures.
2. Implement and troubleshoot network protocols and technologies.
3. Apply network security principles and practices.
4. Utilize Virtualization and Software-Defined Networking (SDN)
5. Evaluate and adapt to emerging trends in computing networks
6. Design, Implement, and Secure Complex Networks</td></tr>
<tr><td colspan="2">Content:</td><td colspan="6">Chapter 1: Introduction to Computer Networks
1. Networking fundamentals
2. Network topologies and architecture
3. ISO/OSI model
4. Network components and devices
Chapter 2: Network Protocols and Communication
1. TCP/IP protocol suite</td></tr>
</table>

	2. Routing and switching 3. Network addressing and subnetting 4. Transport and application layer protocols **Chapter 3: Network Design and Management** 1. LAN and WAN design 2. Network performance optimization 3. Network management tools and techniques 4. Network troubleshooting **Chapter 4: Network Security** 1. Threats, vulnerabilities, and risk assessment 2. Firewall and intrusion detection systems 3. VPNs and secure communication 4. Security best practices **Chapter 5: Advanced Network Technologies** 1. Virtualization and cloud computing 2. Software-Defined Networking (SDN) 3. Internet of Things (IoT) 4. 5G and future network technologies —---------- **Project** : Students will undertake a comprehensive project, applying their knowledge to design, implement, and secure a computer network solution. This project will include documentation and presentations. —---------- **Self-study:** **Objective:** Deepen your knowledge of network security. ❑ Read books and research papers on network security, focusing on advanced topics. ❑ Set up a home lab to practice firewall configurations and intrusion detection systems. ❑ Explore penetration testing and ethical hacking techniques to understand vulnerabilities.
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work

Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,2. and 3** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes , 4,5 and 6**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain the fundamental principles, technologies, and advanced concepts related to computer networks. By the end of the final exam, students should be able provides a comprehensive understanding of network design, protocols and security.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Advanced Networking Design Skills. 2. Advanced Networking Protocol Implementation 3. Network Security Expertise 4. Virtualization and Software-Defined Networking (SDN) 5. Network Performance Optimization
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] "Computer Networking: Principles, Protocols, and Practice" by Olivier Bonaventure [2] "Computer Networking: Principles and Protocols" by Dr. Anish Arkatkar [3] "Routing and Switching Essentials Companion Guide" by Cisco Networking Academy [4] "Computer Networks" by Andrew S. Tanenbaum and David J. Wetherall

Department of Computer Science
C-1.6.1 Communication Skills

Last Update Date: May 10, 2024

Syllabus Sheet

Unit	**Transversal Unit 1**		**Course**	**Communication Skills**		**Code**	**C-1.6.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**1**
	Workshop	**-**					
	Project	**-**		**Mixed**	**XX**		
	Self Study	**35h00**					
Total workload		**56h00 / Semester**			**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher		**Mme Sonia Agili**		**Involved Teacher(s)**	**Mme Sonia Agili**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI** **-M.A. PHD. M.A. OMRANI** **-M.A. PHD. WAEL SELLAMI** **- Director of study A. SELMI**		**Objective**	**Obj 2 & Obj 3**		
				PLO	**P8 : Communication and Collaboration** **P9: Lifelong Learning and Adaptability**		

Prerequisites:	**Language Proficiency:** Engineering students at EQF level 7 are expected to have a good command of the French language. Depending on the university and program, this may require proof of a minimum level of French proficiency, such as a DELF (Diploma in French Studies) certificate or an equivalent. **Basic Engineering Knowledge:** A basic understanding of engineering principles is often assumed as a prerequisite for effective communication within the context of engineering. Students should have completed courses in subjects like mathematics, physics, and basic engineering principles. **Computer Literacy:** Proficiency in using computer applications and software relevant to engineering tasks and communication tools is crucial. Familiarity with word processing, presentation software, and email communication is often expected.
Corequisites:	**Technical Writing and Documentation:** Learning how to write technical reports, proposals, and documentation is crucial for engineering students. This course would teach students how to effectively communicate their ideas in writing. **Oral Communication Skills:** A course focusing on oral communication, including giving presentations, participating in meetings, and conveying ideas clearly and persuasively.

	Intercultural Communication: In an international engineering program, understanding the cultural nuances of communication is important. This course can help students navigate communication with people from diverse backgrounds. **Project-Based Learning:** Often, engineering programs include project-based courses where students work in teams. Effective communication within a team and the ability to present project findings are essential.
Learning outcomes	Upon completion of this teaching course, the student will be able to acquire: 1. **Fundamental concepts of communication**: awareness of the complex process of communication, identification of the means of communication, the functions of speech, the obstacles to communication enhancing interpersonal and intercultural communication skills. 2. **Expression techniques:** Expressing oneself correctly in terms of written language and in the context of professional documents (letters, minutes of meetings, notes, files, press reviews, etc.), by adapting communication strategies to different audiences and contexts and applying critical thinking and problem-solving techniques to communication challenges. 3. **Presentation techniques:** Utilize multimedia and digital tools for effective communication.
Content:	**Chapter 1: Introduction to communication skills** 1. Understanding the importance of communication in computer science engineering. 2. Key communication principles. 3. Barriers to effective communication and how to overcome them. 4. Developing self-awareness in communication. 5. Verbal communication techniques. **Chapter 2: Written Communication Skills** 1. Professional writing in computer science engineering. 2. Technical report and documentation. 3. Email etiquette and effective writing. 4. Grammar and style in technical writing. 5. Effective presentation skills. 6. Public speaking and audience engagement. 7. Handling Q&A sessions. **Chapter 3: Interpersonal Communication and Teamwork** 1. Building effective interpersonal relationships. 2. Conflict resolution and negotiation skills. 3. Effective teamwork and collaboration. 4. Leadership and group dynamics. **Chapter 4: Cross-Cultural Communication** 1. Communicating in a diverse global environment. 2. Cultural sensitivity and global awareness. 3. Ethical considerations in computer science and engineering communication. 4. Intellectual property and plagiarism. 5. Privacy and security in communication. 6. Responsible use of technology in communication. **Chapter 5: Special Topics in Communication** 1. Effective job interviewing and resume writing. 2. Communication in project management.

	3. Social media and online communication. 4. Industry-specific communication skills. **Practical Work:** - Students will apply their communication skills in a real-world scenario, such as a group presentation or a technical report. - Peer and instructor evaluations. - Course reflection and feedback. **Self-study:** Students will: - Practice grammar rules and expand their vocabulary. - Read books, articles, and engineering-related materials in French. - Keep a journal to practice writing in the language. - Create presentations on engineering topics in French then will record and review them for improvement. - Collaborate with fellow students or professionals on engineering projects in French.
Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work.
Assessment method(s):	A midterm written test (D.S) is planned to ensure the evaluation of the **learning outcomes** achievement level. A practical project / portfolio presentation to evaluate: - Class Participation and Engagement - Individual and Group Presentations - Written Assignments and Reports
Assessment criteria(s) :	**Knowledge and understanding:** Fundamental concepts of communication: awareness of the complex process of communication, identification of the means of communication, the functions of speech, the obstacles to communication. **Skills:** Expressing oneself correctly in terms of written language and in the context of professional documents (letters, minutes of meetings, notes, files, press reviews, etc.), which presupposes an adequate command of the French language itself and of certain computer tools. **Professional writing:** writing an agenda, a memo, a summons, letters, reports, minutes, etc.
Assessment measure(s):	Assigning of a final average: Assignments, Quizzes, Exams, Assignments, Quizzes, Exams,

Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	**1.** Acquire knowledge of organizational communication. **2.** Express themselves orally in front of an audience or in a small group, in terms of expression as such, gestures, attitudes and mastery of the material assigned to them. **3.** Expressing oneself correctly in terms of written language and in the context of professional documents (letters, minutes of meetings, notes, files, press reviews, etc.), which presupposes an adequate command of the French language itself and of certain computer tools. **4.** Acquire the necessary skills to enter professional life: Digital and Multimedia Communication & Cross-Cultural Communication and Ethical Practices.
Teaching Manual:	● PowerPoint presentation ● Lecturing ● Tutorials ● Courses project Lecturing
References:	[1] Richard Jones : "Communication in the Real World: An Introduction to Communication Studies". [2] Jérôme Koechlin : “Professional communication: the keys to success”. Paris : Edition PPUR 2015(1st edition). [3] Stephen Lucas : "The Art of Public Speaking" .Academic journals and articles relevant to advanced communication studies.

Department of Computer Science

C-1.6.2 Technical English 1 (Scientific English)

Last Update Date: February 17, 2024

Syllabus Sheet

Unit	**Transversal Unit 1**		**Course**	**Technical English 1 (Scientific English)**		**Code**	**C-1.6.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**1**
	Workshop	**-**					
	Project	**-**		**Mixed**	**XX**		
	Self Study	**35h00**					
Total workload	**56h00 / Semester**				**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher	**Mme Besma Ben Ismail**		**Involved Teacher(s)**	**Mme Besma Ben Ismail**			
Review committee	**-Prof. PHD. MOUNIR ZRIGUI** **-M.A. PHD. M.A. OMRANI** **-M.A. PHD. WAEL SELLAMI** **- Director of study A. SELMI**		**Objective**	**Obj 2 & Obj 3**			
			PLO	**P8 : Communication and Collaboration** **P9: Lifelong Learning and Adaptability**			

Prerequisites:	**English Proficiency:** Students should have a solid foundation in English language skills, at least at the B2 level or equivalent on the Common European Framework of Reference for Languages (CEFR).
Corequisites:	**Introduction to Computer Science / Electrical Engineering studies**: A basic understanding of the field. **Academic Writing Skills:** Focus on academic writing skills, including grammar, style, and citation. This is essential for students to succeed in writing research papers. **Technical Communication:** Covering general technical and scientific communication skills, including presentation techniques and professional communication. **Research Methods:** Introducing students to research methodologies and practices, as this knowledge is essential for conducting research and writing research papers in English.
Learning outcomes	Upon completion of this teaching course, the student will be able to: **1.** Introduce themselves both formally and informally to others. **2.** Understand different types of written texts and summarize them. **3.** Understand graphics, visuals, tables, and diagrams and analyze them orally or transfer them into a written text. **4.** Speak fluently and confidently.
Content:	**Chapter 1: Introduction to technical English** 1. Course overview, expectations, and assessment criteria 2. Academic writing fundamentals

	3. Listening and speaking exercises 4. Computer science and engineering glossary building **Chapter 2: Reading and understanding technical texts** 1. Reading strategies for research papers and technical documents 2. Summarizing technical texts 3. Discussing and analyzing scientific articles **Chapter 3: Writing techniques for computer science** 1. Structuring academic essays and reports 2. Citing and referencing sources in APA or IEEE style 3. Peer review and feedback on writing assignments **Chapter 4: Developing technical presentation skills** 1. Preparing and delivering technical presentations 2. Effective use of visuals (charts, graphs, etc.) 3. Peer presentations and feedback **Chapter 5: Communication engineering and IT** 1. Business English and professional communication 2. Emails, reports, and documentation for engineering projects 3. Collaborative group projects **Practical Work:** 1. Student presentations on research papers 2. Course reflection and feedback 3. Final assessments and grading **Self-study:** Students will: - Choose a textbook or course material specifically designed for engineering students. Some good options include "English for Engineering" or "Technical English" textbooks. - Explore engineering-related websites, blogs, and forums to access technical articles and resources. Websites like IEEE, ACM, and engineering journals can be valuable. - Invest in a technical English dictionary or use online resources like Cambridge or Oxford English Dictionaries for engineering terminology. - Start with basic writing exercises and gradually move to more complex technical writing.
Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work , activities , T. B. L And P.B.L.
Assessment method(s):	A midterm written test (D.S) is planned to ensure the evaluation of the **learning outcomes** achievement level. A practical project / portfolio presentation to evaluate: - Class Participation and Engagement - Individual and Group Presentations

	- Written Assignments and Reports - Final Project.
Assessment criteria(s) :	**Knowledge and understanding:** Students should be able to introduce themselves both formally and informally to others, understand different types of written texts and summarize them and understand graphics, visuals, tables, and diagrams and analyze them orally or transfer them into a written text. **Skills :** - Speaking fluently and confidently. - Acquiring new vocabularies related to the main topics. - Writing accurately and coherently.
Assessment measure(s):	Assigning of a final average: Placement test, indirect assessment, exercises, peer to peer assessment, Recording a conversation, weekly assignments, and Quizzes.
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Develop advanced listening skills in technical and scientific contexts. 2. Enhance reading comprehension of complex technical and research texts. 3. Improve written communication skills for academic and professional purposes. 4. Expand computer science and engineering vocabulary and terminology. 2. Develop effective presentation and discussion skills in English. 3. Apply language skills to create reports, research papers, and presentations in the field of computer science and engineering.
Teaching Manual:	● PowerPoint presentation ● Lecturing ● Tutorials ● Courses project Lecturing.
References:	[1] Murphy, Raymond. English Grammar in Use. Fifth Edition. Cambridge University Press. 2019. [2] Eastwood, John. Oxford guide to English grammar. Oxford University Press. 1997. [3] "Technical English: Writing, Reading, and Speaking" by Nell Ann Pickett and Ann Laster Online academic databases and research journals. [4] Language learning software (e.g., Duolingo, Rosetta Stone). [5] TED Talks and technical lectures.

Department of Computer Science

C-2.1.1-Discrete mathematics (Graph Theory and Operational Research)

Last Update Date: February 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Mathematical Foundations 2</td><td>Course</td><td colspan="2">Discrete mathematics (Graph Theory and Operational Research)</td><td>Code</td><td>C-2.1.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>31,5h</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">2</td></tr>
<tr><td>Workshop</td><td>--</td></tr>
<tr><td>Project</td><td>--</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Self Study</td><td>25h00</td></tr>
<tr><td colspan="2" rowspan="2">Total workload</td><td colspan="3" rowspan="2">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Ali Ben Mrad</td><td>Involved Teacher(s)</td><td colspan="2">Mr Ali Ben Mrad</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1 & Obj 3</td></tr>
<tr><td>PLO</td><td colspan="2">P2: Implementation Techniques
P7: Analytical and Critical Thinking
P3: Innovation, Research and Emerging Technologies</td></tr>
<tr><td colspan="2">Prerequisites:</td><td colspan="6">Proficiency in elementary mathematical concepts, including algebra, calculus, and logic.
No specific prior coursework in discrete mathematics required.</td></tr>
<tr><td colspan="2">Corequisites:</td><td colspan="6">Concurrent enrollment in courses related to algorithms and data structures is recommended.</td></tr>
<tr><td colspan="2">Learning outcomes</td><td colspan="6">Upon completion of this teaching module, the student will be able to:

1. Grasp the essential concepts of discrete mathematics, including sets, relations, and combinatorics, providing a solid mathematical foundation.
2. Understand core principles of graph theory, including graph representations, connectivity, traversal algorithms, and applications in various domains.
3. Apply linear and integer programming techniques to formulate and solve optimization problems, with an emphasis on real-world applications.
4. Analyze and solve combinatorial problems, such as network flows, transportation assignments, and scheduling, using algorithmic approaches.
5. Develop mathematical models for practical scenarios, apply critical thinking to make informed decisions, and communicate mathematical ideas effectively.</td></tr>
<tr><td colspan="2">Content:</td><td colspan="6">Chapter 1: Introduction to Discrete Mathematics:
1. Overview of discrete mathematics and its significance.</td></tr>
</table>

2. Sets, relations, and functions.
3. Fundamental principles of combinatorics and counting.

Chapter 1: Graph Theory Foundations:
1. Introduction to graphs, vertices, edges, and graph types.
2. Basic graph operations, properties, and applications.
3. Graph connectivity, trees, and their importance.

Chapter 2: Linear Programming Basics:
1. Fundamentals of linear programming and its components.
2. Formulating optimization problems as linear programs.
3. Graphical solutions and optimization principles.

Chapter 3: Combinatorial Problem Solving:
1. Exploring combinatorial optimization problems and their real-world applications.
2. Greedy algorithms, dynamic programming, and algorithmic strategies.
3. Practical examples in scheduling, routing, and assignment.

Chapter 4: Operational Research Essentials:
1. Introduction to operational research, its objectives, and methodologies.
2. Application of linear programming in operational research models.
3. Network optimization, decision analysis, and case studies.

Chapter 5: Mathematical Modeling and Applications:
1. Techniques for developing mathematical models for real-world scenarios.
2. Application of mathematical models to practical decision-making processes.
3. Case studies and projects involving mathematical modeling

Self Study & Project Proposal: Optimizing a Transportation Network

Project Overview:
The aim of this project is to apply the fundamental principles learned in the course to optimize a transportation network. Students will work in teams to tackle real-world logistics and transportation challenges by employing concepts from discrete mathematics, graph theory, and operational research.

Project Objectives:
- Apply Graph Theory: Utilize graph theory concepts to model the transportation network, considering nodes as locations and edges as routes.
- Linear Programming Formulation: Formulate a linear programming model to optimize the transportation process, including considerations for cost minimization, resource allocation, and time constraints.
- Algorithmic Solutions: Implement optimization algorithms, such as the simplex method and network flow algorithms, to solve the linear programming model efficiently.
- Practical Decision-Making: Analyze the results and make informed decisions regarding route selection, resource allocation, and cost optimization.
- Communication Skills: Present the project findings and optimization strategies in a clear and concise manner through written reports and presentations.

Project Stages:
1. Stage 1: Problem Definition

	Teams select a transportation problem, defining objectives, constraints, and variables. Begin researching relevant literature and data sources. 2. Stage 2: Mathematical Modeling Apply discrete mathematics concepts to create a mathematical model. Formulate the optimization problem as a linear programming model. 3. Stage 3: Algorithm Implementation Implement linear programming algorithms to solve the optimization model. Analyze and refine the algorithms for efficiency. 4. Stage 4: Data Collection and Testing Collect real-world data related to the transportation problem. Test the algorithms with real data and fine-tune parameters. 5. Stage 5: Results and Recommendations Analyze the results, considering the practical implications. Develop recommendations for optimizing the transportation network. 6. Stage 6: Documentation and Presentation Prepare a comprehensive project report detailing the problem, mathematical model, algorithmic solutions, data analysis, and recommendations. Deliver a clear and engaging presentation to the class. **Target Skills Developed:** 1. Mathematical Modeling: Apply discrete mathematics concepts to real-world problems. 2. Linear Programming Proficiency: Utilize linear programming techniques for optimization. 3. Algorithmic Problem-Solving: Implement and analyze algorithms to solve complex problems. 4. Data Analysis: Gather, analyze, and interpret real-world data for decision-making. 5. Communication Skills: Effectively present findings and recommendations to an audience. **Self-Study:** Throughout the project, students are encouraged to engage in self-study and research to deepen their understanding of discrete mathematics, graph theory, and operational research concepts. Assign self-study tasks related to the specific project challenges, such as advanced linear programming techniques, network flow algorithms, and case studies of transportation optimization in various industries. By completing the self-study assignments, students will enhance their knowledge and skills in preparation for project work and the final exams, ensuring a well-rounded understanding of the course material. This project integrates course content, develops target skills, and encourages self-study, providing students with a practical and engaging learning experience in discrete mathematics, graph theory, and operational research.
Teaching and learning method(s):	Face to face (face-to-face) around 32 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1 and 2.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 3, 4 and 5.**

Assessment criteria(s) :	During the midterm exam, students will define and explain fundamental discrete mathematics concepts) and demonstrate comprehensive understanding of core principles of graph theory. In the final exams, students will apply linear and integer programming to real-world optimization problems and showcase proficiency in solving complex combinatorial challenges. Additionally, they will demonstrate the ability to develop and communicate mathematical models for practical decision-making scenarios through project work.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Fundamental Mathematical Proficiency: Mastery of essential mathematical concepts, including sets, relations, and combinatorics. 2. Core Graph Theory Knowledge: Comprehensive understanding of core principles in graph theory and its diverse applications. 3. Linear Programming Skills: Proficiency in linear programming formulation, graphical solutions, and optimization techniques. 4. Combinatorial Problem-Solving Abilities: Skill in solving combinatorial optimization problems using algorithmic strategies. 5. Mathematical Modeling Expertise: Ability to develop and apply mathematical models for practical decision-making.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Epp, Susanna S. "Discrete Mathematics with Applications." Cengage Learning, 2019. [2] Bondy, John Adrian, and U. S. R. Murty. "Graph Theory." Springer, 2008. [3] Winston, Wayne L. "Operations Research: Applications and Algorithms." Cengage Learning, 2018. [4] Papadimitriou, Christos H., and Kenneth Steiglitz. "Combinatorial Optimization: Algorithms and Complexity." Courier Dover Publications, 1998. [5] Hillier, Frederick S., and Gerald J. Lieberman. "Introduction to Operations Research." McGraw-Hill Education, 2018.

Department of Computer Science
C-2.1.2-Numerical Analysis

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Mathematical Foundations 2**		**Course**	**Numerical Analysis**		**Code**	**C-2.1.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**31h0**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**2**
	Workshop	**--**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**25h00**					
Total workload		**56h00 / Semester**			**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher		**Mr Imed Ben Salha**		**Involved Teacher(s)**	**Mr Imed Ben Salha**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI** **-M.A. PHD. M.A. OMRANI** **-M.A. PHD. WAEL SELLAMI** **- Director of study A. SELMI**		**Objective**	**Obj 1 & Obj3**		
				PLO	**P7: Analytical and Critical Thinking**		

Prerequisites:	• Basic knowledge of calculus and linear algebra. • Familiarity with computer programming (recommended).
Corequisites:	• Mathematics for engineers
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Know the methods of Gauss, factorization and Chalersky 2. Define iterative methods 3. Know the methods of Jacobi, Gauss, relaxation and gradient. 4. Define and Classify Approximations 5. Master the fixed point methods, Newton's 6. Acquire the algorithms 7. Compare algorithms 8. Solve differential equations
Content:	**A- Course content** **Chapter 1:** Introduction to Numerical Analysis 1.1. Overview of Numerical Methods • Introduction to numerical methods and their importance. • Classification of numerical methods. 1.2. Sources of Errors in Numerical Computations • Round-off errors. • Truncation errors.

	● Error analysis in numerical calculations. **Chapter 2:** Root Finding and Optimization 2.1. Bisection Method ● Principles of the bisection method. ● Applications and limitations. 2.2. Newton-Raphson Method ● Iterative techniques for root finding. ● Convergence and practical considerations. 2.3. Optimization Techniques ● Introduction to optimization problems. ● Gradient-based methods and algorithms. **Chapter 3:** Linear Algebraic Equations 3.1. Gaussian Elimination ● Solving linear systems using Gaussian elimination. ● LU decomposition and its significance. 3.2. LU Decomposition ● Detailed exploration of LU decomposition. ● Numerical stability considerations. 3.3. Iterative Methods ● Iterative approaches to solving linear systems. ● Conjugate gradient and Jacobi methods. **Chapter 4:** Interpolation and Approximation 4.1. Polynomial Interpolation ● Interpolation principles and polynomial representation. ● Lagrange and Newton interpolation methods. 4.2. Least Squares Approximation ● Least squares fitting for data approximation. ● Application in curve fitting. 4.3. Spline Interpolation ● Introduction to spline interpolation. ● Cubic splines and their use in data smoothing. **Chapter 5:** Numerical Integration 5.1. Trapezoidal Rule ● Approximating definite integrals using trapezoidal rule. ● Error estimation and improvement. 5.2. Simpson's Rule ● Numerical integration using Simpson's rule. ● Comparison with other methods. 5.3. Adaptive Quadrature ● Adaptive techniques for accurate integration. ● Principles of adaptive quadrature. **Chapter 6:** Ordinary Differential Equations 6.1. Initial Value Problems ● Introduction to initial value problems in ODEs. ● Euler's method and other basic approaches. 6.2. Numerical Stability and Stiffness ● Discussion on stability of numerical methods for ODEs. ● Handling stiff ODEs and implicit methods.

	Chapter 7: Applications of Numerical Analysis in Scientific Research 7.1. Overview of Applications ● Role of numerical analysis in scientific research. ● Examples of scientific domains utilizing numerical methods. 7.2. Case Studies and Examples ● Detailed case studies from physics, biology, engineering, and other fields. ● How numerical analysis aids in solving real scientific problems. 7.3. Importance of Numerical Simulations ● Understanding complex scientific phenomena through simulations. ● Benefits and challenges of numerical modeling in research. 7.4. High-Performance Computing ● The role of supercomputing and high-performance clusters in scientific research. ● Parallel computing and its impact on scientific simulations. **B- Self-study:** ● Additional reading assignments and exercises for in-depth understanding. ● This detailed syllabus provides a comprehensive overview of the course content with specific topics, subtopics, and sub-subtopics.
Teaching and learning method(s):	● Lectures ● Interactive discussions ● Laboratory sessions ● Group projects ● Self-paced learning
Teaching technique(s):	1. Demonstrations 2. Problem-solving exercises 3. Case studies 4. Guest lectures from industry experts
Assessment method(s):	A mid-term written test and a final written test (Exam) are provided. • The mid-term written test will be devoted to the evaluation of the level of achievement of learning outcomes 1, 2, 3 and 4. • The final written test is devoted to the evaluation of the level of achievement of learning outcomes 5, 6, 7, and 8.
Assessment criteria(s) :	During the mid-term test, students should be able to: ● Define direct methods for solving linear systems During the final exam, students should be able to: ● Solve linear and nonlinear systems, ● Master iterative methods, ● Know approximations, ● Compare algorithms, etc.
Assessment measure(s):	Award of the final academic grade

Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated according to the following formula: ● Mid-term written test: 20% ● TP exam: 20% ● Final written exam: 60%.
Target skill(s):	● Problem-solving. ● Critical thinking. ● Numerical algorithm implementation. ● Effective communication. ● Programming skills.
Teaching Manual:	● Course material+tutorial ● Practice session support
References:	● Burden, R. L., & Faires, J. D. (2010). Numerical Analysis. Cengage Learning. ● Kincaid, D., & Cheney, W. (2013). Numerical Analysis: Mathematics of Scientific Computing. American Mathematical Society. ● Quarteroni, A., Sacco, R., & Saleri, F. (2006). Numerical Mathematics. Springer. ● Gautschi, W. (2012). Numerical Analysis: An Introduction. Birkhäuser.

Department of Computer Science

C-2.2.1-Algorithms and data Structures 2

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Fundamental Computing 2**		**Course**	**Algorithms and data Structures 2**		**Code**	**C-2.2.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Software Engineering**
Type of education	**Lecture**	**21h 00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**2**
	Workshop	**21h 00**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**42h00**					
Total workload	**84h00 / Semester**				**Coefficient**		**3**
					ECTS credit		**3**
Responsible Teacher	**Mr Wael Sallami**		**Involved Teacher(s)**	**Mr Wael Sallami**			
Review committee	**-Prof. PHD. MOUNIR ZRIGUI** **-M.A. PHD. M.A. OMRANI** **-M.A. PHD. WAEL SELLAMI** **- Director of study A. SELMI**		**Objective**	**Obj 1 & Obj 3**			
			PLO	**P2 : Implementation Techniques** **P7: Analytical and Critical Thinking**			

Prerequisites:	Procedural Programming: Mastery of fundamental programming concepts, such as loops, conditional statements, functions, etc. Basic Algorithmic Concepts: Understanding of simple algorithms, the structure of a program, and programming logic.
Corequisites:	An understanding of discrete mathematics, including topics such as set theory, logic, and combinatorics, is often essential for the analysis of algorithms and data structures.
Learning outcomes	Upon completion of this teaching module, the student will be able to: **1.** Demonstrate an understanding of advanced data structures, including trees, graphs, heaps, and hash tables, and their applications in solving complex problems. **2.** Implement and analyze advanced sorting and searching algorithms, such as QuickSort, MergeSort, HeapSort, and binary search, and evaluate their performance. **3.** Apply graph theory concepts to solve real-world problems, including graph traversal, shortest path algorithms, and minimum spanning tree algorithms. **4.** Employ dynamic programming techniques to solve problems efficiently and understand the principles of overlapping subproblems and memoization.
Content:	**Chapter 1:** Stacks and Queues 1. Introduction to stacks and queues. 2. Implementation of stacks and queues using arrays and linked lists. 3. Applications of stacks and queues in algorithm design. 4. Comparison and contrast of stacks and queues.

Tutorial:
- TD1: Solve a set of exercises related to stack and queue operations to reinforce your understanding.

Chapter 2: Trees
1. Binary search trees (BSTs): operations, balancing.
2. AVL trees and red-black trees.
3. B-trees and B+ trees.

Tutorial:
- TD2: implement operations for binary trees, such as finding the size, height, and lowest common ancestor of two nodes.

Chapter 3: Graphs
1. Representation of graphs (adjacency matrices, adjacency lists).
2. Graph traversal (DFS, BFS).
3. Shortest paths (Dijkstra, Bellman-Ford).
4. Minimum spanning trees (Prim, Kruskal).

Tutorial:
- TD3: Solve practical applications of the Depth-First Search algorithm for graph traversal, providing step-by-step guidance on how to explore and analyze graphs.

Chapter 4: Advanced Data Structures
1. Binary heaps and their applications.
2. Priority queues.
3. Advanced hash tables.
4. Trie data structures.

Tutorial:
- TD4: implementing priority queues using binary heaps, demonstrating how to efficiently manage and manipulate data with this advanced data structure.

Project :
Project Title: Advanced Algorithm Implementation and Analysis of a graph problem
Project Description:
In this project, students will select a real-world problem or scenario that requires the application of advanced data structures and algorithms. They will be tasked with designing, implementing, and analyzing an algorithmic solution to address the problem. The project can be done individually or in small teams.

Self-study:
1. Research
   Study programming fundamentals and its relevance in engineering.
   Collect relevant resources and references.
2. Proposal
   Choose an engineering problem.
   Explain how to implement complex algorithms.
4. Project Report
   Create a report summarizing the project.
   Reflect on the experience and insights gained.
5. Presentation (Optional)
   Share project findings with the class if desired.
6. Doing Exercise and QCM published in easylearn.ovh platform.

Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1 and 2.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes ., 3 and 4.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain the core concepts of stacks and queues and tree including their characteristics, operations, and use cases. By the end of the final exam, students should be able to represent various types of graphs using different data structures, such as adjacency matrices and adjacency lists, and explain the implications of these representations on computational efficiency,
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Design algorithms for various data structures, analyze their efficiency, and make informed choices regarding algorithm selection. 2. Implement the data structures using various methods, such as arrays, linked lists, and trees, and apply them to solve complex problems effectively. 3. Develop problem-solving skills, particularly in graph-related scenarios, by applying algorithms like Depth-First Search, Breadth-First Search, and Dijkstra's algorithm to find solutions and optimize intricate problems. 4. Analyze and compare the computational complexity of algorithms and data structures, understand time and space complexity, and evaluate the efficiency of different approaches in problem-solving.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks

References:	[1] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms. The MIT Press. [2] Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2014). Data Structures and Algorithms in Java. John Wiley & Sons. [3] Bellman, R. (1958). On a routing problem. Quarterly of Applied Mathematics, 16(1), 87-90. [4] Dijkstra, E. W. (1959). A note on two problems in connection with graphs. Numerische Mathematik, 1(1), 269-271.

Department of Computer Science
C-2.2.2-Computational Complexity

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Fundamental Computing 2**		**Course**	**Computational Complexity**		**Code**	**C-2.2.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Software Engineering**
Type of education	**Lecture**	**21h 00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**2**
	Workshop	**--**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**35h00**					
Total workload		**56h00 / Semester**			**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher		**Mr Wael Sallami**	**Involved Teacher(s)**		**Mr Wael Sallami**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI** **-M.A. PHD. M.A. OMRANI** **-M.A. PHD. WAEL SELLAMI** **- Director of study A. SELMI**	**Objective**		**Obj 1 & Obj 3**		
			PLO		**P2 : Implementation Techniques** **P7: Analytical and Critical Thinking**		

Prerequisites:	• Mastery of fundamental programming concepts, such as loops, conditional statements, functions, etc • Basic Algorithmic Concepts: Understanding of simple algorithms, the structure of a program, and programming logic.
Corequisites:	Familiarity with the analysis of algorithms, which focuses on time and space complexity analysis, can provide a practical foundation for understanding computational complexity.
Learning outcomes	Upon completion of this teaching module, the student will be able to: **1.** Develop efficient and effective algorithms **2.** Understand the notion of complexity of an algorithm **3.** Know how to estimate the complexity of an iterative or recursive algorithm **4.** Mastering the “divide and conquer” approach **5.** Understand graph and tree traversal algorithms **6.** Understand the complexity classes: P, NP, NP hard
Content:	**Chapter 1: Complexity of iterative algorithms** 1. Reminders and General Information on Algorithms 2. General information on complexity 3. First complexity calculations 4. Asymptotic behavior of functions

Tutorial:

- TD1: determine iterative algorithms solving the same problem, know how to calculate the time complexity, compare the efficiency of these algorithms.

Chapter 2: Complexity of recursive algorithms

1. Recursion
2. Types of recursivity.
3. Linear recurrent sequences
4. Divide and conquer algorithms.

Tutorial:

- TD2: Study the case of the complexity of recursive algorithms using recurrent sequences.
- TD3: Apply the “Master Theorem” theorem to solve certain recurrence equations.

Chapter 3: NP-Completeness

1. Introduction
2. Classes of problems
3. Class P (Polynomial time)
4. Class NP (Non-deterministic polynomial time)
5. Class NP-Complete
6. Class NP-Hard

Tutorial:

- TD4: Classify P and NP-Complete problems, solve NP-hard class problems.

Chapter 4: Dynamic Programming

1. Introduction to Dynamic Programming
2. Knapsack Problem and Dynamic Programming
3. Dynamic Programming for TSP
4. Dynamic Programming for Subset Sum

Tutorial:

- TD5: practice designing and analyzing approximation algorithms, and understand the trade-offs between optimality and efficiency.

Project :

This project is to improve students' basic knowledge of advanced programming. It was like being a complement for practical work. Each student is required to carefully read the case study, solve the problem raised and seek the complexity of the sub-programs implemented using the concepts studied such as graphs, dynamic tables, etc.

Self-study:

1. Research
   Study programming fundamentals and its relevance in engineering.
   Collect relevant resources and references.
2. Proposal
   Choose an engineering problem.
   Explain how to calculate algorithm complexity.
4. Project Report

	Create a report summarizing the project. Reflect on the experience and insights gained. 5. Presentation (Optional) Share project findings with the class if desired. 6. Doing Exercise and QCM published in easylearn.ovh platform.
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,..2.. and 3.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes ., …4,5 and 6.**
Assessment criteria(s) :	During the midterm exam, students should be able to to calculate time-based algorithmic complexity by comparing the efficiency of algorithms solving the same problem. Thus, they should determine which of the proposed algorithms is the most optimal. By the end of the final exam, students should be able to develop algorithms based on trees and graphs. In addition, they should master the principle of NP Completeness and the resolution of hard problems.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	**1.** Master programming in C language **2.** Have notions of recursion and computational complexity **3.** Know and correctly use basic data structures (lists, stacks, queues, etc.) **4.** Understand basic data structures (trees, binary search trees): study of their abstract properties, their concrete representations, their application and the main algorithms that manipulate them.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks

References:	[1] Aldon, G., Germoni, J., Mény, J. M.:Cours Complexité en algorithmique.IREM de Lyon, (2012). [2] Malka, D.: Cours Complexité Algorithmique. Creative Commons, (2014) [3] Cormen, L & Rivest, Introduction to algorithms, mcgraw hill, 1028 pages. [4] Labelle, théorie des graphes, modulo, 183 pages [5] Olivier B, fondements de l'informatique logique, modèles, et calculs, cours inf423 de l'ecole polytechnique, 2013, pp. 234. [6] Jean F et Alexandre S. optimisation combinatoire – théorie et algorithmes, chapitre 15: np-complétude, 2010, 664p

Department of Computer Science

C-2.3.1-Language Theory and Compilation

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Design and Theory</td><td>Course</td><td colspan="2">Language Theory and Compilation</td><td>Code</td><td>C-2.3.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>10,5h</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">2</td></tr>
<tr><td>Workshop</td><td>21h00</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>10,5h</td></tr>
<tr><td>Self Study</td><td>14h00</td><td colspan="2"></td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Mohsen Maraoui</td><td>Involved Teacher(s)</td><td colspan="3">Mme Safa Ben Salem</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj1 & Obj3</td></tr>
<tr><td>PLO</td><td colspan="3">P1 : Systems Specification and Design Methodologies
P2 : Implementation Techniques
P7: Analytical and Critical Thinking</td></tr>
<tr><td colspan="2"><u>Prerequisites:</u></td><td colspan="6">Advanced coursework in programming languages and algorithms.
Thorough understanding of formal language theory.
Proficiency in a high-level programming language (e.g., Java, C++).</td></tr>
<tr><td colspan="2"><u>Corequisites:</u></td><td colspan="6">None</td></tr>
<tr><td colspan="2"><u>Learning outcomes</u></td><td colspan="6">Upon completion of this teaching module, the student will be able to:

1. Language Understanding: Develop a deep understanding of formal languages, grammars, and language hierarchies, including context-sensitive languages.
2. Compiler Design: Design and implement compilers for complex programming languages, encompassing lexical analysis, parsing, semantic analysis, optimization, and code generation.
3. Optimization Techniques: Apply advanced optimization techniques to enhance the efficiency and performance of generated code.
4. Error Handling: Implement advanced error detection, recovery, and correction mechanisms, ensuring robustness in the face of errors.
5. Language Formalization: Formally define and analyze the syntax and semantics of programming languages using cutting-edge formal methods and semantic frameworks.</td></tr>
</table>

Content:

Chapter 1: Introduction to Language Theory and Compilation

1. Overview of the compilation process.
2. Formal language theory fundamentals.
3. Historical context of compiler development.
4. Emerging trends in language design.

Chapter 2: Lexical Analysis

1. Regular expressions and finite automata.
2. Lexical analyzers and tokenization.
3. Challenges in handling complex lexemes.
4. Practical examples and case studies.

Chapter 3 ISyntax Analysis

1. Context-free grammars and parsing techniques.
2. Syntax-directed translation and abstract syntax trees.
3. Parsing table generation and optimization.
4. Advanced parsing algorithms.

Chapter 4: ISemantic Analysis

1. Semantic analysis and type checking.
2. Symbol tables, scope analysis, and resolution.
3. Handling polymorphism and type inference.
4. Advanced techniques in semantic analysis.

Chapter 5: I Intermediate Code Generation

1. Three-address code generation.
2. Intermediate code optimization techniques.
3. Representing control flow in intermediate code.
4. Case studies of real-world intermediate code.

Chapter 6: ICode Generation and Optimization

1. Code generation for target machines.
2. Code optimization strategies for maximizing code efficiency.
3. Register allocation and addressing modes.
4. Profile-guided optimization and dynamic compilation.

Chapter 7: I Error Handling and Recovery

1. Error detection, reporting, and recovery strategies.
2. Advanced techniques for error correction and resilience.
3. Design principles for error-resistant compilers.
4. Case studies in error handling.

Chapter 8: I Programming Language Formalization

1. Formal specification of programming languages.
2. Denotational and operational semantics.
3. Advanced semantic frameworks and theorem proving.
4. Practical applications of language formalization.

Project and Self-study :

Project Proposal: Compiler Development and Optimization

	Objective: Design and implement a compiler for a subset of a high-level programming language, focusing on advanced compiler development, code optimization, and error handling. **Project Description:** Language Subset: Select a subset of a programming language with essential features. Compiler Development: Create a complete compiler, including lexical, syntax, semantic analysis, intermediate code generation, and code optimization. Error Handling: Implement advanced error detection, reporting, and recovery mechanisms for graceful error handling. **Self-Study Tasks:** Advanced Lexical Analysis: Explore complex lexeme handling and tokenization optimization. Advanced Code Optimization: Study advanced code optimization techniques, including data flow analysis. Formal Language Formalization: Investigate formal language frameworks like operational semantics. Error Handling Best Practices: Research best practices in error handling and recovery. **Project Milestones:** Milestone 1: Submit project proposal with language subset and initial design. Milestone 2: Implement lexical and syntax analysis. Milestone 3: Add semantic analysis and intermediate code generation. Milestone 4: Integrate error handling and initial code optimization. Milestone 5: Submit the complete compiler with a detailed report. Assessment: Evaluate based on compiler completeness, correctness, error handling effectiveness, and code optimization. Documenting all tasks is also required.
Teaching and learning method(s):	Face to face 42 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1 and 2.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 3, 4 and 5.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain fundamental concepts in formal language theory, lexical analysis and syntax analysis techniques. By the end of the final exam, students should be able to design and implement complete compilers for complex programming languages, optimize generated code effectively, develop advanced error handling mechanisms, formally encompassing advanced compiler design, optimization techniques, error handling, and formal language formalization.
Assessment measure(s):	Assigning of a final average

Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Advanced Language Analysis: Master complex language analysis. 2. Compiler Development: Excel in compiler design. 3. Optimization Expertise: Optimize code efficiently. 4. Error Handling Mastery: Perfect error management. 5. Formal Language Formalization: Formalize language syntax. 6. Advanced Problem Solving: Solve complex challenges adeptly.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Muchnick, S. S. (1997). "Advanced Compiler Design and Implementation." Morgan Kaufmann. [2] Appel, A. W. (2002). "Modern Compiler Implementation in C/Java/ML." Cambridge University Press. [3] Cooper, K. D., & Torczon, L. (2011). "Engineering a Compiler." Morgan Kaufmann. [4] Nielson, F., & Nielson, H. R. (2012). "Semantics with Applications: An Appetizer." Springer.

Department of Computer Science

C-2.3.2-Analysis and Design Methodology

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Design and Theory**		**Course**	**Analysis and Design Methodology**		**Code**	**C-2.3.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Software Engineering**
Type of education	**Lecture**	**21h 00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**2**
	Workshop	**21h 00**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**42h00**					
Total workload		**84h00 / Semester**			**Coefficient**		**3**
					ECTS credit		**3**
Responsible Teacher		**Mme Emna Hkiri**		**Involved Teacher(s)**	**Mme Emna Hkiri**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj1**		
				PLO	**P1 : Systems Specification and Design Methodologies**		

Prerequisites:	Fundamentals of Software Engineering, Introduction to Databases and Object-Oriented Programming (OOP).
Corequisites:	None
Learning outcomes	At the end of this module, the student will be able to: 1. Understand the fundamentals of Information System Modeling. 2. Differentiate between the analysis and design phases of system development. 3. Master the creation and interpretation of Use Case Diagrams. 4. Analyze system dynamics using System Sequence Diagrams. 5. Design and implement effective static system architectures. 6. Integrate and scale systems using advanced architectural methodologies. 7. Implement advanced design patterns in system development. 8. Prioritize and implement security considerations in system design.
Content:	**Chapter 1: Introduction to Information System Modeling Learning Sub-Achievements:** 1. Explain the importance of modeling an Information System 2. Differentiate between the analysis phase and the design phase 3. Categorize UML diagrams based on the following dimensions: functional, static, and dynamic. **Chapter 2: Functional Analysis Learning Sub-Achievements:** 1. Explain the significance of the Use Case Diagram (UCD)

2. Identify and utilize the fundamental concepts of a UCD
3. Develop a UCD related to a case study

Chapter 3: Dynamic Analysis Learning Sub-Achievements:

1. Explain the significance of dynamic analysis in an Information System using the System Sequence Diagram
2. Identify and apply the fundamental concepts of a System Sequence Diagram
3. Construct the diagram related to the dynamic analysis of a given case study.

Chapter 4: Static Analysis and Static Design Learning Sub-Achievements:

1. Explain the significance of static analysis in an information system through the Analysis Class Diagram
2. Identify and use the basic concepts of an Analysis Class Diagram
3. Develop the Analysis Class Diagram of a given system
4. Explain the significance of static design in an information system
5. Identify and apply the basic concepts of a Design Class Diagram while considering a given logical architecture
6. Differentiate between an Analysis Class Diagram and a Design Class Diagram
7. Create the Design Class Diagram of a given system.

Chapter 5: Dynamic Design Learning Sub-Achievements:

1. Explain the significance of dynamic design in an information system
2. Identify and use the basic concepts of an Object Sequence Diagram while considering a given logical architecture
3. Differentiate between an Object Sequence Diagram and a System Sequence Diagram
4. Identify and apply the basic concepts of a State-Transition Diagram
5. Choose and create the most appropriate dynamic design diagram for a given description.

Chapter 6: Advanced System Integration and Scalability Design

1. Integration Principles in Modern Systems
2. Microservices Architecture and Design
3. Designing for Scalability
4. Event-Driven Architectures

Chapter 7: Advanced Design Patterns and Optimization Strategies

1. Exploring Advanced Design Patterns
2. Database Design and Optimization at Scale
3. Performance Testing and Optimization
4. Security Considerations in Design

Workshops :

1. Workshop on Information System Modeling Basics:
2. Use Case Diagram (UCD) Workshop:
3. Dynamic Analysis Workshop:
4. Static Analysis and Design Workshop:

	5. Dynamic Design Workshop:
Teaching and learning method(s):	Face to face 42 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of learning outcomes **1, 2, and 3.** ● The final written test is devoted to the evaluation of the level of achievement of learning outcomes **4, 5, 6, 7, and 8**
Assessment criteria(s) :	"During the midterm exam, students should be able to define and explain the fundamentals of Information System Modeling, differentiate between the analysis and design phases of system development, and master the creation and interpretation of Use Case Diagrams. By the end of the final exam, students should be able to analyze system dynamics using System Sequence Diagrams, design and implement effective static system architectures, integrate and scale systems using advanced architectural methodologies, implement advanced design patterns in system development, and prioritize and implement security considerations in system design.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Analyze an information system from a functional perspective 2. Analyze an information system from a dynamic perspective 3. Analyze an information system from a static perspective 4. Design an information system from a dynamic perspective 5. Design an information system from a static perspective
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks

References:

[1] "UML Distilled: A Brief Guide to the Standard Object Modeling Language" by Martin Fowler.

[2] "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development" by Craig Larman.

[3] "The Unified Software Development Process" by Ivar Jacobson, Grady Booch, and James Rumbaugh.

[4] "UML for the IT Business Analyst" by Howard Podeswa.

[5] Spécification de l'Unified Modeling Languag (UML) par l'OMG: http://www.omg.org/spec/UML/

[6] UML 2 - de l'apprentissage à la pratique - Laurent Audibert

Department of Computer Science
C-2.4.1-Web Development 2

Last Update Date: February 17, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Web and Programming 2</td><td>Course</td><td colspan="2">Web Development 2</td><td>Code</td><td>C-2.4.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="4">Semester</td><td rowspan="4">2</td></tr>
<tr><td>Workshop</td><td>10,5h</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>25h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Mohamed Amine Omrani</td><td>Involved Teacher(s)</td><td colspan="3">Mr Abderrahim Allani</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="3">P1 : Systems Specification and Design Methodologies
P2: Implementation Techniques
P4: Web and Software Development</td></tr>
<tr><td>Prerequisites:</td><td colspan="7">Web Development 1 or equivalent experience with basic web notions.
Familiarity with basic database concepts and SQL.
Understanding of HTML, CSS, Bootstrap, JavaScript and basic web development principles.</td></tr>
<tr><td>Corequisites:</td><td colspan="7">Advanced Database Management</td></tr>
<tr><td>Learning outcomes</td><td colspan="7">Upon completion of this teaching module, the student will be able to:

1. Understand and implement advanced PHP concepts and features.
2. Design and develop web applications using object-oriented PHP.
3. Implement secure database interactions using PDO.
4. Analyze and optimize web applications for performance and scalability.
5. Incorporate advanced web development practices, including API integrations and modern web architectures.
6. Understand and prevent common web security threats in PHP applications.
7. Implement and manage user authentication and session management.
8. Integrate third-party libraries and frameworks in PHP.</td></tr>
<tr><td>Content:</td><td colspan="7">Chapter 1: Introduction to Basic PHP Concepts
1.1 Introduction to Dynamic Web Development</td></tr>
</table>

1.2 PHP Syntax and Structure
1.3 Variables, Arrays, and Data Types
1.4 Control Structures and Functions
1.5 Forms and PHP Data Handling
1.6 Basic File Operations

Chapter 2: Advanced PHP Concepts and Techniques
2.1 Error Handling and Exception Handling
2.2 Namespaces and Autoloading
2.3 Advanced Array Functions and Manipulations
2.4 PHP Streams and File I/O
2.5 Regular Expressions in PHP

Chapter 3: Object-Oriented PHP
3.1 Introduction to OOP Concepts
3.2 Classes, Objects, and Inheritance
3.3 Polymorphism and Interfaces
3.4 Traits and Namespaces in OOP
3.5 Design Patterns in PHP

Chapter 4: Secure Database Interactions using PDO
4.1 Introduction to PDO and Why Use It
4.2 Connecting to Databases with PDO
4.3 Prepared Statements and Binding Values
4.4 Handling Errors in PDO
4.5 Transactions and Multiple Database Interactions

Chapter 5: Web Application Optimization and Scalability
5.1 Performance Metrics and Monitoring
5.2 Database Optimization Techniques
5.3 Caching Mechanisms in PHP
5.4 Efficient Code and Algorithm Optimization
5.5 Scalable Web Architecture Basics

Chapter 6: Web Security in PHP
6.1 Common Security Threats and Vulnerabilities
6.2 Input Validation and Sanitization
6.3 Securing File Uploads and System Operations
6.4 Defense against SQL Injection and XSS
6.5 Secure Sessions and Cookies

Chapter 7: User Authentication and Session Management
7.1 User Registration and Password Security
7.2 Login Mechanisms and Session Handling
7.3 Role-based Access Control
7.4 Secure Token-Based Authentication
7.5 OAuth and Third-party Authentication

Chapter 8: API Integration and Web Services
8.1 Introduction to Web APIs
8.2 Consuming APIs using PHP
8.3 Building RESTful Web Services in PHP
8.4 JSON and XML Parsing and Manipulation

	8.5 API Authentication and Security **Chapter 9: Advanced Web Architectures** 9.1 MVC and MVVM Design Patterns 9.2 Microservices Architecture in PHP 9.3 Introduction to PHP Frameworks 9.4 Modular and Component-based Development 9.5 Continuous Integration and Deployment in PHP **Chapter 10: Third-Party Libraries and Frameworks** 10.1 Composer and PHP Package Management 10.2 Introduction to Popular PHP Frameworks 10.3 Third-party Security Libraries and Extensions 10.4 Utility Libraries for Development Efficiency 10.5 Extending PHP with PECL Extensions
	Workshops: 1. Workshop on Object-Oriented PHP Development. 2. PDO-based CRUD Application Development Workshop. 3. Web Security Vulnerabilities and Defense Workshop. 4. API Development and Integration Workshop.
	Project and Self-study: **Topic**: Development of a Secure and Scalable E-commerce Web Application using Object-Oriented PHP and PDO. This project will challenge students to incorporate the advanced principles taught in the course to build a functional and secure e-commerce platform.
Teaching and learning method(s):	Face to face 32 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work (Integrated course, Practical work, Debate , Conduct project).
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2 and 3.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 4, 5, 6, 7 and 8.**

Assessment criteria(s) :	During the midterm exam, students should be able to define and explain advanced PHP concepts, object-oriented PHP practices, and secure database operations using PDO. By the end of the final exam, students should be able to optimize web applications, integrate APIs, understand common web security threats, manage user sessions, and incorporate third-party libraries.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Advanced PHP programming. 2. Web application design using OOP. 3. Secure and efficient database operations with PDO. 4. Web application optimization techniques. 5. Recognizing and mitigating web security threats. 6. Integration of external APIs and libraries.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks ● Polytech EasyLearn Platform
References:	[1] "PHP Objects, Patterns, and Practice" by Matt Zandstra. [2] "Modern PHP: New Features and Good Practices" by Josh Lockhart. [3] "PHP and MySQL Web Development" by Luke Welling and Laura Thomson. [4] "Pro PHP Security: From Application Security Principles to the Implementation of XSS Defenses" by Chris Snyder and Michael Southwell.

Department of Computer Science

C-2.4.2 Object Oriented Programming (OOP)

Last Update Date: February 17, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Web and Programming 2</td><td>Course</td><td colspan="2">Object Oriented Programming (OOP)</td><td>Code</td><td>C-2.4.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td rowspan="2">--</td><td rowspan="4">Semester</td><td rowspan="4">3</td></tr>
<tr><td>Workshop</td><td>21h00</td></tr>
<tr><td>Project</td><td>-</td><td rowspan="2">Mixed</td></tr>
<tr><td>Self Study</td><td>42h00</td><td>XX</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Mohamed Amine Omrani</td><td>Involved Teacher(s)</td><td colspan="3">Mr Mohamed Amine Omrani</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="3">P1 : Systems Specification and Design Methodologies
P2: Implementation Techniques</td></tr>
<tr><td>Prerequisites:</td><td colspan="7">Procedural Programming: Mastery of fundamental programming concepts, such as loops, conditional statements, functions, etc.
Basic Algorithmic Concepts: Understanding of simple algorithms, the structure of a program, and programming logic.</td></tr>
<tr><td>Corequisites:</td><td colspan="7">Programming Languages: Familiarity with at least one programming language. Even though Java is the language used for the course, having a foundation in any language facilitates the learning of OOP (Object-Oriented Programming) concepts.
Knowledge in discrete mathematics and algorithmics</td></tr>
<tr><td>Learning outcomes</td><td colspan="7">Upon successful completion of this course, students will:
1. Understand and apply the syntax and fundamental concepts of Java.
2. Analyze and implement programs using object-oriented programming.
3. Master the fundamental principles of OOP: Instancing (Classes & objects), encapsulation, inheritance, polymorphism, abstraction, etc.
4. Know and apply the data structures in Java.
5. Effectively use Java IDEs for development.</td></tr>
<tr><td>Content:</td><td colspan="7">Chapter 1: Introduction to Java: Basic Syntax, Types, Operators
1. History and evolution of Java
2. Structure of a Java program
3. Basic syntax: variables, constants, comments</td></tr>
</table>

4. Data types: primitive and reference
5. Operators: arithmetic, relational, logical

Practical Work:

TP1: Installing a Java development environment (such as Netbeans, Eclipse, or IntelliJ IDEA) and writing the first program "Hello, World!".

TP2: Manipulating data types, creating variables, and performing simple operations (addition, multiplication, etc.).

Chapter 2: OOP: Concepts and Paradigms

1. Introduction to OOP
2. Programming paradigms: procedural vs. object
3. Advantages of OOP
4. Key concepts: classes, objects, attributes, methods
5. Object lifecycle.

Practical Work:

TP3: Designing a simple class representing an entity like a "Book" or a "Person", then creating instances and displaying their attributes.

Chapter 3: Classes and Objects in Java

1. Definition and structure of a class
2. Instantiating an object
3. Using methods and attributes
4. Constructors: definition and overloading
5. Keywords this and new

Practical Work:

TP4: Developing a simple management application (for example, a library with books and members) using multiple classes and objects.

Chapter 4: Encapsulation and Access Modifiers

1. Introduction to encapsulation
2. Access modifiers: private, public, protected, and default
3. Getters and setters
4. Benefits of encapsulation in terms of security and flexibility.
5. Practical Work:
6. TP5: Enhancing the previous application by applying encapsulation principles. Adding getters and setters, and manipulating access modifiers to protect the data.

Chapter 5: Inheritance and Interface: Superclasses, Subclasses, Re-use classes

1. Concept of inheritance
2. Creating subclasses
3. Using the super keyword
4. Overloading vs. overriding
5. Blocking inheritance with final
6. Concept of an interface.

Practical Work:

TP6: Extending the previous application by introducing inheritance and applying interfaces. For instance, creating specialized classes like "Novel" and "Magazine" inheriting from the "Book" class.

Chapter 6: Polymorphism and Bindings

1. Introduction to polymorphism
2. Polymorphism with overloading
3. Polymorphism with overriding
4. Late binding vs. early binding

5. Use of the abstract keyword and abstract classes

Practical Work:

TP7: Implementing polymorphism by adding overloaded and overridden methods in the application. Exploring late bindings using superclass type references to manipulate subclass objects.

Chapter 7: Collections in Java

1. Introduction to collections
2. The list: ArrayList, LinkedList
3. The set: HashSet, LinkedHashSet, TreeSet
4. The map: HashMap, LinkedHashMap, TreeMap
5. Traversing and manipulating collections.

Practical Work:

TP8: Integrating collections to manage lists of entities (e.g., a list of books in a library). Using different structures such as ArrayList and HashSet to store, retrieve, and manipulate objects.

Chapter 8: Introduction to Exceptions in Java

1. What is an exception?
2. Try, catch, and finally
3. Creating custom exceptions
4. Managing common exceptions in Java
5. Benefits of exception handling

Practical Work:

TP9: Adding exception handling to the application (e.g., for cases where a book is not found in the library). Creating and managing custom exceptions for application-specific situations.

Project :

Project Title: "Library Management System in Java"

Project Description:

In this project, students will design and implement a Library Management System using Java. The goal of this project is to apply the principles of Object-Oriented Programming (OOP) and gain practical experience with Java syntax, data structures, and development tools.

Self-study:

1. Research
   - Study OOP fundamentals and its relevance in engineering.
   - Review engineering projects that successfully use OOP.
   - Collect relevant resources and references.
2. Proposal
   - Choose an engineering problem.
   - Propose a project with clear objectives and OOP principles.
   - Explain how OOP can benefit your project.
3. Implementation
   - Develop the engineering solution using an OOP-focused language.
   - Document design, coding, and challenges.
   - Include code examples.
4. Project Report
   - Create a report summarizing the project.
   - Describe OOP application and its impact.

	Reflect on the experience and insights gained. 5. Presentation (Optional) Share project findings with the class if desired. 6. Doing Exercise and QCM published in easylearn.ovh platform.
Teaching and learning method(s):	Face to face 42 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lecture with workshops and projects
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, and 2.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 3, 4, and 5.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain the main concepts of object-oriented programming, such as classes, objects, encapsulation, inheritance, interfaces, and polymorphism. They should also be proficient in the basic syntax of Java and be able to design, instantiate, and manipulate objects using appropriate classes. By the end of the final exam, students should be able to design a complete object-oriented application in Java, implementing object-oriented techniques such as handling collections, exception management, and the effective use of the principles of encapsulation, inheritance, and polymorphism. They should also be able to demonstrate a deep understanding of the advantages and challenges of object-oriented programming compared to other programming paradigms.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Ability to develop, test, and deploy complex and scalable Java applications. 2. Mastery of advanced design patterns and their application in industrial contexts. 3. Skill in identifying and resolving performance, security, and design issues in Java applications. 4. Ability to evaluate, integrate, and work with major Java frameworks.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Course support

References:	[1] P. J. Deitel and H. Deitel, Java: How to Program, Early Objects. Prentice Hall, 2009. [2] B. Eckel, Thinking in Java. Prentice Hall, 2020. [3] H. Schildt, Java: The Complete Reference. McGraw-Hill, 2022. [4] K. Sierra and B. Bates, Head First Java. O'Reilly Media, 2022.

Department of Computer Science
C-2.5.1-Internet and Protocols

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Network and Databases</td><td>Course</td><td colspan="2">Internet and Protocols</td><td>Code</td><td>C-2.5.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="4">Semester</td><td rowspan="4">2</td></tr>
<tr><td>Workshop</td><td>10,5h</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>-</td></tr>
<tr><td>Self Study</td><td>23h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mme Elhem Blel</td><td>Involved Teacher(s)</td><td colspan="3">Mme Elhem Blel</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="3">P5: Network and Information Systems Management</td></tr>
<tr><td>Prerequisites:</td><td colspan="7">Foundation in Networking
Data Communication Basics</td></tr>
<tr><td>Corequisites:</td><td colspan="7">Advanced Network Protocols
Network Simulation and Modeling</td></tr>
<tr><td>Learning outcomes</td><td colspan="7">Upon completion of this teaching module, the student will be able to:
1. Analyze and understand the internet's architecture and its role in global communication.
2. Evaluate and explain the core protocols that drive internet communication.
3. Design and implement internet-based solutions with a focus on advanced protocols.
4. Analyze and troubleshoot complex internet-related issues.
5. Stay informed about emerging trends and technologies in internet and networking.
6. Explore Internet of Things (IoT) and Emerging Technologies</td></tr>
<tr><td>Content:</td><td colspan="7">Chapter 1: Introduction to the Internet and Its Architecture
1. Internet evolution and history
2. Internet architecture: backbone, edge, and access networks
3. Internet service providers (ISPs) and peering
4. Internet governance and organizations
Chapter 2: Internet Protocol Suite (TCP/IP)
1. Overview of the TCP/IP protocol suite
2. IPv4 and IPv6 addressing and routing</td></tr>
</table>

	3. Transport layer protocols: TCP and UDP 4. Network layer protocols: IP, ICMP, and IGMP **Chapter3: Internet Routing and BGP** 1. Routing algorithms and protocols 2. BGP (Border Gateway Protocol) and autonomous systems 3. Internet routing security and BGP hijacking 4. Internet exchange points (IXPs) and route servers **Chapter 4: DNS and Web Protocols** 1. Domain Name System (DNS) fundamentals 2. DNS hierarchy and resolution process 3. Web protocols: HTTP, HTTPS, and web servers 4. Content delivery networks (CDNs) **Chapter 5: Email and Messaging Protocols** 1. SMTP (Simple Mail Transfer Protocol) for email 2. POP and IMAP for email retrieval 3. Instant messaging and chat protocols 4. Security and encryption in email and messaging **Chapter 6: Internet Security and Privacy** 1. Internet security threats and vulnerabilities 2. Firewall and intrusion detection systems (IDS) 3. VPN (Virtual Private Network) technologies 4. Privacy concerns and measures **Chapter 7: Network Address Translation (NAT) and NAT64** 1. NAT fundamentals and applications 2. IPv6 transition mechanisms, including NAT64 3. NAT traversal for peer-to-peer communication 4. Practical implementation of NAT and NAT64 **Chapter 8: Internet of Things (IoT) and Emerging Technologies** 1. Internet of Things (IoT) architecture and protocols 2. 5G and its impact on internet connectivity 3. Edge computing and fog computing 4. Future trends and challenges in internet technology —---------- **Self-study:** -Network Design and Routing Research advanced routing concepts, including BGP and OSPF. Simulate network designs using software like GNS3 or Packet Tracer.
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned.

	● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,2 and 3.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 4,5 and 6.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain understanding of advanced topics related to internet technologies and protocols, as well as their ability to apply this knowledge. effectively. By the end of the final exam, students should be able to Applies critical thinking to analyze complex internet-related issues and devise effective solutions.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. provides an advanced understanding of the internet's architecture, core protocols, and the role they play in modern communication and information systems. 2. It covers both theoretical foundations and practical aspects of internet technologies and protocols. ……
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1]"Computer Networking: Principles, Protocols, and Practice" by Olivier Bonaventure: This comprehensive textbook covers the fundamentals of computer networking, including internet protocols, routing, and network security. [2]"TCP/IP Illustrated" by Richard Stevens: A classic resource for understanding the TCP/IP protocol suite with in-depth explanations and real-world examples. [3] "Computer Networks" by Andrew S. Tanenbaum and David J. Wetherall: This well-known textbook provides insights into computer network principles, including internet protocols

Department of Computer Science
C-2.5.2-Database Fundamentals

Last Update Date: may 7, 2024

Syllabus Sheet

Unit	**Network and Databases**		**Course**	**Database Fundamentals**		**Code**	**C-2.5.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Software Engineering**
Type of education	**Lecture**	**21h**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**2**
	Workshop	**10,5h**					
	Project	**10,5h**		**Mixed**	**XX**		
	Self Study	**42h**					
Total workload		**84h00 / Semester**			**Coefficient**		**3**
					ECTS credit		**3**
Responsible Teacher		**Mr Mohsen Maraoui**	**Involved Teacher(s)**	**Mr Mohsen Maraoui**			
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**	**Objective**	**Obj 1**			
			PLO	**P1 : Systems Specification and Design Methodologies P2: Implementation Techniques**			

Prerequisites:	Structured Programming
Corequisites:	programming and Dynamic Structures
Learning outcomes	At the end of this course, the student should master the concepts related to relational databases, enabling their creation and manipulation, including: − Understanding all the underlying concepts of databases. − Delving into the concepts of modeling, design, and database implementation. − Designing a database while adhering to data model rules and standards. − Converting a conceptual schema into a coherent logical (relational) schema. − Applying relational algebra operations to query a database.
Content:	Chapter 1 : Introduction to Databases 1. Definition of a database 2. Importance of databases in information management

3. Evolution of database systems

Chapter 2 : Data Models

1. Hierarchical model
2. Network model
3. Relational model
4. Object-oriented model
5. NoSQL models (document, key-value, columnar, graph, etc.)

Chapter 3 : Relational Database Management System (RDBMS)

1. Introduction to RDBMS
2. Key components: tables, rows, columns, keys
3. SQL (Structured Query Language)
4. Basic SQL commands: SELECT, INSERT, UPDATE, DELETE

Chapter 4 : Database Design

1. Entity-Relationship (ER) modeling
2. Relational schema
3. Normalization
4. Denormalization

Chapter 5 : Database Architecture

1. Client-server architecture
2. Three-tier architecture
3. Data storage and indexing
4. Transaction management
5. Concurrency control

Chapter 6 : Data Integrity and Constraints

1. Primary keys
2. Foreign keys
3. Unique constraints
4. Check constraints
5. Referential integrity

Chapter 7 : Query Optimization and Performance Tuning

1. Indexing strategies
2. Query execution plans
3. Performance monitoring and tuning techniques

Chapter 8 : Database Administration

1. Installation and configuration of DBMS
2. Backup and recovery procedures
3. Monitoring and optimization
4. Disaster recovery planning

Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1 and 2** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes3,4 and 5**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain databases fundamental and database design By the end of the final exam, students should be able create and inquire databases with sql language
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. modeling, design, and database implementation. 2. Converting a conceptual schema into a coherent logical (relational) schema. 3. Applying relational algebra operations to query a database.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	− Gardarin G., "Bases de Données - objet/relationnel," Eyrolles, 1999, ISBN: 2-212-09060-9 − Gardarin G., "Maîtriser les Bases de Données: modèles et langages," Eyrolles − Carrez C., "Des Structures aux Bases de Données," Masson − Marcenac, P., "SGBD relationnels, Optimisation des performances," Eyrolles − Date C.J., "An Introduction to Database Systems," Addison Wesley − Date C.J., "A Guide to SQL"

Department Computer science

C-2.6.1 Engineer's Ethics

Last Update Date: October 7, 2023

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Transversal Unit 2</td><td>Course</td><td colspan="2">Engineer's Ethics</td><td>Code</td><td>C-2.6.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer science</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">2</td></tr>
<tr><td>Workshop</td><td>-</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>-</td><td rowspan="2">Coefficient</td><td rowspan="2">2</td></tr>
<tr><td>Self Study</td><td>35h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mme Hjiri Mouna</td><td>Involved Teacher(s)</td><td colspan="3">Mme Hjiri Mouna</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 2</td></tr>
<tr><td>PLO</td><td colspan="3">P10: Ethical Responsibility and Integrity
P8: Communication & Collaboration</td></tr>
<tr><td>Prerequisites:</td><td colspan="7">Students should have a demonstrated proficiency in English, either through strong interest in computer science and technology and a willingness to engage in discussions on ethical topics are beneficial for success in the course.</td></tr>
<tr><td>Corequisites:</td><td colspan="7">Engineering Fundamentals: It's beneficial if students have some foundational knowledge of engineering principles and concepts. A corequisite in basic engineering courses can help students relate engineering ethics to practical engineering work.
Introduction to Philosophy or Ethics: While not mandatory, having a corequisite in an introductory philosophy or ethics course can help students grasp the fundamental concepts of ethics and morality, which are essential in understanding engineering ethics.
Communication Skills: Engineering ethics often involve discussions and presentations. Therefore, having a corequisite in a communication or public speaking course can help students express their ethical viewpoints effectively.</td></tr>
</table>

Learning outcomes	Upon completion of this teaching course, the student will be able to: **1.** Explain key ethical concepts relevant to computer science. **2.** Identify ethical issues and dilemmas in computing. **3.** Analyze and evaluate ethical scenarios, considering different perspectives. **4.** Apply ethical principles to make informed decisions in the field of computer science. **5.** Communicate and debate ethical issues effectively. **6.** Work effectively in teams and collaborate with colleagues in addressing ethical challenges within engineering projects and organizations. **7.** Comprehend and appreciate the societal and environmental impact of engineering projects and strive to promote sustainable and socially responsible engineering practices. **8.** Recognize their role in addressing broader societal issues through engineering solutions, such as equity, safety, and well-being.
Content:	**Chapter 1: Introduction to Engineering Ethics** 1. Definition of ethics and morality 2. Historical perspective of engineering ethics 3. Ethical codes and guidelines in engineering 4. Ethical Theories: Deontology, Consequentialism, Virtue ethics and Ethical relativism. **Chapter 2: Professional Responsibility** 1. The ethical decision-making process 2. Case studies and analysis 3. Ethical frameworks in engineering 4. The role of engineers in society 5. Codes of ethics for engineers 6. Whistleblowing and professional integrity **Chapter 3: Moral Dilemmas in Engineering and social responsibility** 1. Analysis of real-world ethical dilemmas 2. Group discussions and case studies 3. Ethical decision-making exercises 4. Sustainability in engineering 5. Environmental ethics 6. Social responsibility in engineering practice 7. Guest Speakers and Industry Perspective. **Practical Work:** 1. Final Project and Review 2. Students work on an ethical analysis project. 3. Final presentations and class discussion 4. Course review and reflections **Self-study:**

	Students will: - Study the codes of ethics relevant to your engineering field. Most professional engineering organizations, like the American Society of Civil Engineers (ASCE) or the IEEE, have their own codes of ethics. Familiarize yourself with these codes. - Learn about different ethical frameworks and theories. Utilitarianism, deontology, virtue ethics, and others can provide a structured approach to ethical decision-making. - If possible, seek out a mentor who is experienced in your field of engineering and has a strong ethical background. They can provide guidance and insights. - Put what you learn into practice. Apply ethical principles in your work and everyday life, and encourage others to do the same.
Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the **easylearn.ovh** platform to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners. Invitation of professionals and experts from the industry sharing ethical experiences (Q&A sessions).
Teaching technique(s):	Classroom lectures with tutorials and practical work and activities including: Inclusive Teaching, Active Learning , Flipped Classroom , Experiential Learning , Problem-Based Learning (PBL) , Role-Playing and Simulations , Technology Integration , Assessment and Feedback and Collaborative Learning.
Assessment method(s):	A midterm written test (D.S) is planned to ensure the evaluation of the **learning outcomes** achievement level. A practical project / portfolio presentation to evaluate: - Class Participation and Engagement - Individual and Group Presentations - Written Assignments and Reports - Final project.
Assessment criteria(s) :	**Knowledge and understanding:** Students Understanding of Ethical Concepts (Cognitive): Identification and in-depth analysis of ethical dilemmas in engineering scenarios. **Skills :** Ethical Decision-Making (Cognitive and Behavioral) , Effective application and interpretation of relevant ethical codes and guidelines within engineering context and Demonstrated critical thinking (cognitive) skills when evaluating ethical issues and proposing solutions.
Assessment measure(s):	Assigning a final average: Exercises, Peer to peer assessment, class participation and discussions assignments and Quizzes.
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 40% ● Final written exam: 60%.

Target skill(s):	1. Understand the fundamental principles of engineering ethics. 2. Develop the ability to identify ethical issues in engineering practice. 3. Enhance ethical decision-making skills in engineering contexts. 4. Appreciate the societal and environmental impact of engineering projects. 5. Cultivate a sense of professional responsibility and integrity. 6. Respect for diverse cultural, societal, and ethical perspectives when engaging with stakeholders and team members in an international engineering context.
Teaching Manual:	● PowerPoint presentation ● Lecturing ● Tutorials ● Courses project Lecturing
References:	[1] Fleddermann, Charles B., Engineering Ethics, 2012, 4th Edition. [2] Engineering Ethics by Fleddermann, 2012, 4th Edition. [3] Martin, M.W. and R. Schinzinger. Ethics in Engineering. 4 th Edition. (McGraw-Hill, Inc., 2005). [4] Harris Jr., C.E., Pritchard, M.S., Rabins, M.J., Engineering Ethics, Concepts, and Cases: 4 th edition (California: Wadsworth Learning, 2009).

Department of Computer Science
C-1.6.1 Communication Skills

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Transversal Unit 1</td><td>Course</td><td colspan="2">Communication Skills</td><td>Code</td><td>C-1.6.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">1</td></tr>
<tr><td>Workshop</td><td>-</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>-</td><td rowspan="2">Coefficient</td><td rowspan="2">2</td></tr>
<tr><td>Self Study</td><td>35h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mme Sonia Agili</td><td>Involved Teacher(s)</td><td colspan="2">Mme Sonia Agili</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 2</td></tr>
<tr><td>PLO</td><td colspan="2">P10: Ethical Responsibility and Integrity
P8: Communication & Collaboration</td></tr>
<tr><td><u>Prerequisites:</u></td><td colspan="7">Language Proficiency: Engineering students at EQF level 7 are expected to have a good command of the French language. Depending on the university and program, this may require proof of a minimum level of French proficiency, such as a DELF (Diploma in French Studies) certificate or an equivalent.

Basic Engineering Knowledge: A basic understanding of engineering principles is often assumed as a prerequisite for effective communication within the context of engineering. Students should have completed courses in subjects like mathematics, physics, and basic engineering principles.

Computer Literacy: Proficiency in using computer applications and software relevant to engineering tasks and communication tools is crucial. Familiarity with word processing, presentation software, and email communication is often expected.</td></tr>
<tr><td><u>Corequisites:</u></td><td colspan="7">Technical Writing and Documentation: Learning how to write technical reports, proposals, and documentation is crucial for engineering students. This course would teach students how to effectively communicate their ideas in writing.

Oral Communication Skills: A course focusing on oral communication, including giving presentations, participating in meetings, and conveying ideas clearly and persuasively.</td></tr>
</table>

	Intercultural Communication: In an international engineering program, understanding the cultural nuances of communication is important. This course can help students navigate communication with people from diverse backgrounds. **Project-Based Learning:** Often, engineering programs include project-based courses where students work in teams. Effective communication within a team and the ability to present project findings are essential.
Learning outcomes	Upon completion of this teaching course, the student will be able to acquire: 1. **Fundamental concepts of communication**: awareness of the complex process of communication, identification of the means of communication, the functions of speech, the obstacles to communication enhancing interpersonal and intercultural communication skills. 2. **Expression techniques:** Expressing oneself correctly in terms of written language and in the context of professional documents (letters, minutes of meetings, notes, files, press reviews, etc.), by adapting communication strategies to different audiences and contexts and applying critical thinking and problem-solving techniques to communication challenges. 3. **Presentation techniques:** Utilize multimedia and digital tools for effective communication.
Content:	**Chapter 1: Introduction to communication skills** 1. Understanding the importance of communication in computer science engineering. 2. Key communication principles. 3. Barriers to effective communication and how to overcome them. 4. Developing self-awareness in communication. 5. Verbal communication techniques. **Chapter 2: Written Communication Skills** 1. Professional writing in computer science engineering. 2. Technical report and documentation. 3. Email etiquette and effective writing. 4. Grammar and style in technical writing. 5. Effective presentation skills. 6. Public speaking and audience engagement. 7. Handling Q&A sessions. **Chapter 3: Interpersonal Communication and Teamwork** 1. Building effective interpersonal relationships. 2. Conflict resolution and negotiation skills. 3. Effective teamwork and collaboration. 4. Leadership and group dynamics. **Chapter 4: Cross-Cultural Communication** 1. Communicating in a diverse global environment. 2. Cultural sensitivity and global awareness. 3. Ethical considerations in computer science and engineering communication. 4. Intellectual property and plagiarism. 5. Privacy and security in communication. 6. Responsible use of technology in communication. **Chapter 5: Special Topics in Communication** 1. Effective job interviewing and resume writing. 2. Communication in project management.

	3. Social media and online communication. 4. Industry-specific communication skills. **Practical Work:** - Students will apply their communication skills in a real-world scenario, such as a group presentation or a technical report. - Peer and instructor evaluations. - Course reflection and feedback. **Self-study:** Students will: - Practice grammar rules and expand their vocabulary. - Read books, articles, and engineering-related materials in French. - Keep a journal to practice writing in the language. - Create presentations on engineering topics in French then will record and review them for improvement. - Collaborate with fellow students or professionals on engineering projects in French.
Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work.
Assessment method(s):	A midterm written test (D.S) is planned to ensure the evaluation of the **learning outcomes** achievement level. A practical project / portfolio presentation to evaluate: - Class Participation and Engagement - Individual and Group Presentations - Written Assignments and Reports
Assessment criteria(s) :	**Knowledge and understanding:** Fundamental concepts of communication: awareness of the complex process of communication, identification of the means of communication, the functions of speech, the obstacles to communication. **Skills:** Expressing oneself correctly in terms of written language and in the context of professional documents (letters, minutes of meetings, notes, files, press reviews, etc.), which presupposes an adequate command of the French language itself and of certain computer tools. **Professional writing:** writing an agenda, a memo, a summons, letters, reports, minutes, etc.
Assessment measure(s):	Assigning of a final average: Assignments, Quizzes, Exams, Assignments, Quizzes, Exams,

Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	**1.** Acquire knowledge of organizational communication. **2.** Express themselves orally in front of an audience or in a small group, in terms of expression as such, gestures, attitudes and mastery of the material assigned to them. **3.** Expressing oneself correctly in terms of written language and in the context of professional documents (letters, minutes of meetings, notes, files, press reviews, etc.), which presupposes an adequate command of the French language itself and of certain computer tools. **4.** Acquire the necessary skills to enter professional life: Digital and Multimedia Communication & Cross-Cultural Communication and Ethical Practices.
Teaching Manual:	● PowerPoint presentation ● Lecturing ● Tutorials ● Courses project Lecturing
References:	[1] Richard Jones : "Communication in the Real World: An Introduction to Communication Studies". [2] Jérôme Koechlin : “Professional communication: the keys to success”. Paris : Edition PPUR 2015(1st edition). [3] Stephen Lucas : "The Art of Public Speaking" .Academic journals and articles relevant to advanced communication studies.

Department of Computer Science
C-2.6.2-English TOEIC 1

Last Update Date: February 17, 2024

Syllabus Sheet

Unit	**Transversal Unit 2**		**Course**	**English TOEIC 1**		**Code**	**C-2.6.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**3**
	Workshop	**-**					
	Project	**-**		**Mixed**	**XX**		
	Self Study	**35h00**					
Total workload		**56h00 / Semester**			**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher		**Mme Besma Ben Ismail**		**Involved Teacher(s)**	**Mme Besma Ben Ismail**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 2 & Obj 3**		
				PLO	**P8 : Communication and Collaboration P9: Lifelong Learning and Adaptability**		

Prerequisites:	**English Proficiency:** Students should have a basic to intermediate level of English proficiency, as TOEIC is designed to assess skills ranging from beginner to advanced levels. **Vocabulary:** Engineering students should have a good grasp of general and technical English vocabulary, including terms related to their field of study. **Grammar:** Understanding English grammar rules and being able to construct clear and coherent sentences is essential. Listening and Reading: It's important to have developed listening and reading comprehension skills, as the TOEIC test includes sections for both listening and reading comprehension.
Corequisites:	None corequisites for TOEIC 1 as it is primarily designed to assess an individual's English
Learning outcomes	- Improved Listening Skills: Students will develop the ability to understand spoken English in various work-related contexts, such as meetings, conferences, and phone calls. - Enhanced Reading Comprehension: Students will learn to comprehend written English in different forms, including emails, reports, and articles. - Improve vocabulary - Better Time Management - Business Communication Proficiency - Readiness to function in English-speaking workplaces. - Ability to perform tasks that require English language.

Content:	**Section 1: Listening Comprehension** 1. Technical Presentations: You may encounter recordings of technical presentations on topics related to computer science and electrical engineering. 2. Conversations: Understanding conversations related to workplace situations, such as discussing project requirements or solving technical problems. **Section 2: Reading Comprehension** 1. Technical Documents: Be able to read and understand technical documents, manuals, and reports specific to computer science and electrical engineering. 2. Business Correspondence: Understand and respond to emails, memos, and other business-related communication. 3. Graphs and Charts: Interpret information presented in graphs, charts, and tables. **Section 3: Vocabulary and Grammar** 1. Technical Vocabulary: Build a strong technical vocabulary related to computer science and electrical engineering. 2. Professional Language: Understand and use business and professional English vocabulary. 3. Grammar: Demonstrate a good grasp of English grammar, including tenses and sentence structure. **Section 4 : Speaking and Writing** 1. Professional Communication: Practice speaking and writing in a professional, business-like manner. This may include writing reports, emails, or summaries of technical information. 2. Presentation Skills: Prepare for speaking tasks where you need to present your ideas or answer questions on technical topics. 3. Telephone Skills: Develop the ability to have effective phone conversations related to work. **Practical Work:** - Practicing TOIEC 2 tests - Running groups and peer to peer conversations - Course reflection and feedback **Self-study:** Students should: - Familiarize themselves with the TOEIC format, which typically consists of two sections: Listening Comprehension and Reading Comprehension. - Purchase or download official TOEIC preparation materials from the ETS (Educational Testing Service) website. - Improve Listening Skills - Regularly take full-length practice TOEIC tests to become familiar with the format and to gauge your progress. - Review your answers and understand the explanations for any questions you got wrong.

Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work , activities , T. B. L And P.B.L.
Assessment method(s):	A midterm written test (D.S) is planned to ensure the evaluation of the **learning outcomes** achievement level. A practical project / portfolio presentation to evaluate: - Class Participation and Engagement - Individual and Group Presentations - Written Assignments and Reports
Assessment criteria(s) :	Students' proficiency should be at an advanced level, demonstrating the ability to understand complex technical materials, express your ideas clearly and concisely, and engage in professional discussions in English.
Assessment measure(s):	Assigning a final average: Placement test, Indirect assessment, Exercises, Peer to peer assessment, Recording a conversation, Weekly assignments, and Quizzes.
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final examination assessing writing skills and comprehension : 60%
Target skill(s):	1. Develop language skills necessary for effective scientific communication. 2. Understand and apply scientific writing conventions, including structure, clarity, and conciseness. 3. Analyze and critically evaluate technical literature in the field of computer science and engineering. 4. Produce well-structured and coherent technical documents, including research reports, papers, and project proposals. 5. Use appropriate referencing and citation techniques in accordance with academic and professional standards.
Teaching Manual:	● PowerPoint presentation ● Lecturing ● Tutorials ● Courses project Lecturing
References:	[1] TOEIC Official Test Preparation Guide: ETS (Educational Testing Service), the organization that administers the TOEIC, often releases official test preparation materials. These materials are specifically designed to help familiarizing oneself with the test format and questions. [2] Online Resources: There are numerous online resources available to help preparation for TOEIC Level 1. ETS offers official sample questions and practice tests on their website. [3] TOEIC preparation materials, textbooks, and online resources specifically designed for the TOEIC exam.

Department of Computer Science
C-2.7.1-Personal Project

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Project Unit 1**		**Course**	**Personal Project**		**Code**	**C-2.7.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**-**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**1**
	Workshop	**21h00**					
	Project	**-**		**Mixed**	**XX**		
	Self Study	**35h00**					
Total workload	**56h00 / Semester**			**Coefficient**		**2**	
				ECTS credit		**2**	
Responsible Teacher	**Mr Mohamed Amine Omrani**		**Involved Teacher(s)**	**Mr Mohamed Amine Omrani**			
Review committee	**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1, Obj 2 and Obj3**			
			PLO	**P1, P2, P3, P4, P5, P6, P7, P8, P9 and P10**			

Prerequisites:	Enrollment in the first year of a computer science engineering program. **Fundamental Computer Science Knowledge:** Students should have a basic understanding of computer science concepts, including programming, data structures, and algorithms. This knowledge can be acquired through introductory computer science courses taken in the first year of the program. **Mathematics Proficiency:** An understanding of fundamental mathematics concepts, including discrete mathematics, is beneficial as it provides a foundation for problem-solving and algorithm design.
Corequisites:	**Introduction to Software Engineering:** Basic course in software engineering principles can be a valuable corequisite. This course can provide students with a framework for software development practices, project management, and collaboration, which will be directly applicable to their personal project. **Advanced Mathematics Courses:** Depending on the nature of the project, corequisites in advanced mathematics courses might be required. For example, if the project involves complex algorithms, students might need to have completed advanced mathematics courses. **Technical Writing or Communication Skills Course:** These skills are crucial for documenting and presenting their research effectively. **Ethics and Legal Issues in Research:** If the personal project involves data collection or experimentation, a corequisite course on research ethics and legal considerations might be necessary to ensure students conduct their projects ethically and within the legal boundaries.

Learning outcomes	Upon completion of this teaching course, the student will be able to: **1.** Demonstrate a deep understanding of a specific topic or problem in computer science engineering. **2.** Apply research methodologies to investigate and address complex issues in the field. **3.** Develop, implement, and evaluate a computer science project independently. **4.** Communicate their findings and outcomes effectively through written reports and presentations. **5.** Reflect on the project experience and identify areas for personal growth and improvement.
Content:	**Chapter 1: Introduction to Personal Projects** 1. Understanding the importance of personal projects 2. Identifying potential project ideas 3. Selecting and justifying a project idea **Chapter 2: Project Planning** 1. Setting project objectives and scope then developing a project proposal 2. Creating a project timeline and milestones 3. Budget estimation (if applicable) 4. Conducting user interviews and defining functional and non-functional requirements 5. Formulating a project problem statement **Chapter 3: Project Implementation** 1. Implementing the project according to the project plan 2. Version control and collaborative coding 3. Agile development practices (e.g., Scrum or Kanban) 4. Documentation of code and progress **Chapter 4: Project Evaluation and Improvement** 1. Testing and debugging the project 2. Gathering user feedback 3. Iterative development and continuous improvement 4. Monitoring and performance optimization **Practical Work: Project preparing, Showcase and Assessment** - Preparing a final project report - Creating a project presentation - Demonstrating the project to peers and instructors - Reflecting on lessons learned and future improvements - Students showcase their personal projects to the class - Peer evaluation and feedback - Assessment by the instructor. **Self-study:** Students will: - Conduct thorough research on the topic of their projects: Using reliable sources such as academic journals, textbooks, and reputable websites to gather information and deepen your understanding of the subject matter. - Consider the challenges you overcame and the skills you developed. - Ask for feedback from professors, mentors, or experienced engineers to provide insights and suggestions for improvement.

	- Use your newfound skills and knowledge to continue exploring more advanced projects.
Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work, activities.
Assessment method(s):	A midterm written test (D.S) is planned to ensure the evaluation of the **learning outcomes** achievement level. A practical project / portfolio presentation to evaluate: - Project proposal and research plan - Literature review and problem definition - Midterm project report and presentation - Final project report and presentation - Reflective report - Faculty advisor's assessment and feedback.
Assessment criteria(s) :	**Knowledge and understanding and skills:** Students necessary knowledge and skills to engage effectively in their research project while maintaining a balanced workload.
Assessment measure(s):	Assigning a final average: Indirect assessment, exercises, peer to peer assessment, weekly assignments , project presentation.
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Developing **research skills** necessary for effective critical thinking and problem definition. 2. Developing **technical Skills** such as Programming and Development, Data Analysis and Interpretation as well as Advanced Technical Knowledge (machine learning, artificial intelligence, cybersecurity, or other specialized areas). 3. Developing **Project Management skills** for effective Planning and Organization. 4. **Resource Management:** Managing resources effectively, including hardware, software, and data. 5. **Risk Assessment skills :** Identifying potential risks and mitigating strategies for project success. 6. **Communication Skills:** Technical Writing (comprehensive and well-structured technical reports), Oral Presentation and Visual Communication: Creating clear and informative visual aids for presentations and documentation. 7. **Adaptability:** The ability to adapt to unexpected challenges and changes during the project based on Problem-Solving and Collaboration. 8. **Ethical and Professional Awareness skills:** Research Ethics Professional Responsibility. 9. **Reflection and Improvement skills:** Self-Reflection, Continuous Learning.

Teaching Manual:	• PowerPoint presentation • Lecturing • Tutorials • Courses project Lecturing
References:	[1] "Scientific Writing and Communication: Papers, Proposals, and Presentations" by Angelika H. Hofmann. [2] "Research Methodology: A Step-by-Step Guide for Beginners" by Ranjit Kumar - A practical guide to research methodologies and writing. [3] "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig - If the project involves AI, this is a comprehensive guide to AI concepts. [4] "Data Science for Business" by Foster Provost and Tom Fawcett - Useful for data science and machine learning projects. [5] "Engineering Fundamentals: An Introduction to Engineering" by Saeed Moaveni [6] "Mechatronics: Principles and Applications" by Godfrey C. Onwubolu [7] "Introduction to Environmental Engineering" by Mackenzie L. Davis and David A. Cornwell.

Department of Computer Science
C-3.1.1-Artificial Intelligence

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Advanced Topic 1</td><td>Course</td><td colspan="2">Artificial Intelligence</td><td>Code</td><td>C-3.1.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="3">Semester</td><td rowspan="3">3</td></tr>
<tr><td>Workshop</td><td>21h 00</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>42h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Mounir Zrigui</td><td>Involved Teacher(s)</td><td colspan="3">Mr Mounir Zrigui</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1 & Obj 3</td></tr>
<tr><td>PLO</td><td colspan="3">P2 : Implementation Techniques
P3: Innovation, Research and Emerging Technologies
P7: Analytical and Critical Thinking</td></tr>
<tr><td colspan="2">Prerequisites:</td><td colspan="6">Basic knowledge of computer science and programming</td></tr>
<tr><td colspan="2">Corequisites:</td><td colspan="6"></td></tr>
<tr><td colspan="2">Learning outcomes</td><td colspan="6">Upon completion of this teaching module, the student will be able to:

1. Define and explain the fundamental concepts and techniques in artificial intelligence.
2. Apply machine learning algorithms to solve real-world problems.
3. Design and implement intelligent agents and systems.
4. Analyze and evaluate AI models and algorithms.
5. Discuss the ethical and societal implications of artificial intelligence.</td></tr>
<tr><td colspan="2">Content:</td><td colspan="6">Chapter 1: Introduction to Artificial Intelligence
1. Definition and history of AI
2. AI problem-solving techniques
3. Ethical considerations in AI
Chapter 2: Machine Learning
1. Types of machine learning (supervised, unsupervised, reinforcement)
2. Algorithms and models (linear regression, decision trees, neural networks, etc.)
3. Model evaluation and optimization</td></tr>
</table>

	Chapter 3: Intelligent Agents 1. Agent architecture and design 2. Search algorithms and problem-solving 3. Knowledge representation and reasoning **Chapter 4: Natural Language Processing** 1. Understanding and generating human language 2. Text processing, sentiment analysis, and chatbots 3. Applications in NLP **Chapter 5: Computer Vision** 1. Image processing and feature extraction 2. Object recognition and image classification 3. Computer vision applications
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,2 and 3.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 4 and 5.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain fundamental concepts in artificial intelligence, apply machine learning algorithms, and design intelligent agents. By the end of the final exam, students should be able to analyze and evaluate AI models and algorithms and discuss the ethical and societal implications of artificial intelligence.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Critical thinking and problem-solving 2. Data analysis and interpretation 3. Machine learning and AI model implementation 4. Ethical reasoning and decision-making
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks

References:	[1] Intelligence artificielle 4e édition Une approche moderne, Stuart Russell, Peter Norvig, Pearson Edition 2021 [2]Intelligence artificielle Impact sur les entreprises et le business, Jean-Michel Rodriguez, Eni Edition 2022 [3] Introduction au machine learning, Chloé-Agathe Azencott, Dunod Edition 2022 [4] L'intelligence artificielle, Margaret A. Boden, EDP Sciences Edition 2021

Department of Computer Science

C-3.1.2-IoT Foundations (and Embedded Systems)

Last Update Date: June 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Advanced Topic 1</td><td>Course</td><td colspan="2">IoT Foundations (and Embedded Systems)</td><td>Code</td><td>C-3.1.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">3</td></tr>
<tr><td>Workshop</td><td>10,5h</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>25h</td><td colspan="2"></td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Riadh Hourrig</td><td>Involved Teacher(s)</td><td colspan="2">Mr Riadh Hourrig</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="2">P1 : Systems Specification and Design Methodologies
P2 : Implementation Techniques
P3: Innovation, Research and Emerging Technologies.</td></tr>
<tr><td colspan="2">Prerequisites:</td><td colspan="6">Computer science, networking</td></tr>
<tr><td colspan="2">Corequisites:</td><td colspan="6"></td></tr>
<tr><td colspan="2">Learning outcomes</td><td colspan="6">Upon completion of this teaching module, the student will be able to:

1. Basic concepts and gradually advances to cover IoT architecture
2. Protocols, security, and practical implementation
3. In-depth exploration of the Internet of Things (IoT) and embedded systems</td></tr>
<tr><td colspan="2">Content:</td><td colspan="6">Chapter 1: Introduction to IoT and Embedded Systems (Basic Notions)
1. Understanding IoT and its significance
2. Embedded systems fundamentals
3. IoT use cases and applications
4. Basic components and architecture
Chapter 2: IoT Architecture and Components
1. IoT architecture layers (perception, network, and application)
2. Sensors, actuators, and communication modules
3. IoT edge devices and gateways
4. IoT platforms and cloud integration
Chapter 3: IoT Communication Protocols</td></tr>
</table>

1. Overview of IoT communication protocols (e.g., MQTT, CoAP, HTTP, etc.)
2. Selection criteria for communication protocols
3. IoT data serialization formats (e.g., JSON, XML, etc.)
4. Practical exercises with IoT protocols

Chapter 4: IoT Security and Privacy

1. IoT security challenges and threats
2. Security measures for IoT devices and networks
3. Authentication, authorization, and encryption in IoT
4. IoT privacy considerations and compliance

Chapter 5: IoT Data Analytics and Edge Computing

1. Data analytics in IoT
2. Edge computing in IoT for real-time processing
3. IoT data storage and databases
4. Edge device programming and data processing

Chapter 6: IoT Device Management and Scalability

1. Managing IoT devices at scale
2. IoT device provisioning, updates, and monitoring
3. Scalability challenges and solutions
4. Case studies on IoT scalability

Chapter 7: Practical IoT Projects

1. IoT project development and implementation
2. Hands-on IoT device programming
3. IoT application development and deployment
4. Project presentations and documentation

Chapter 8: IoT Innovations and Emerging Trends

1. Emerging IoT technologies and trends
2. IoT in industry 4.0 and smart cities
3. Research and innovation in the IoT field

IoT Search Project:

- **Objective:** To research and explore current trends, challenges, and innovations in the IoT field.
- **Task Description:**
  - **Topic Selection:** Each student will select a unique topic related to IoT. Possible topics include but are not limited to:
    - Emerging IoT technologies and their potential impact
    - Security challenges and novel solutions in IoT
    - IoT applications in smart cities and industry 4.0
    - Comparative analysis of IoT communication protocols
    - Advances in edge computing and data analytics for IoT
    - Case studies on successful IoT implementations
  - **Research and Analysis:** Students will conduct thorough research on their chosen topic, utilizing academic papers, industry reports, and other credible sources.
  - **Report Writing:** Students will compile their findings into a comprehensive report, including an introduction, literature review, methodology, analysis, and conclusions.
  - **Presentation:** Each student will present their research findings to the class, highlighting key insights and discussing implications for the future of IoT.
- **Assessment Criteria:**
  - **Depth and quality of research**
  - **Clarity and coherence of the written report**
  - **Effectiveness of the presentation**
  - **Critical analysis and originality of insights**

	○ **Engagement with current literature and industry practices**
Teaching and learning method(s):	Face to face (face-to-face) 32 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1 and 2.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 3.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain the basic concepts and gradually advance to cover IoT architecture. By the end of the final exam, students should be able to present in-depth exploration of the Internet of Things (IoT) and embedded systems.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	**1.** The course provides an in-depth exploration of the Internet of Things (IoT) and embedded systems.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Cyber Physical Computing for IoT-Driven Services, Vladimir hahanov, Springer Edition 2017 [2] IoT Standards with Blockchain: Enterprise Methodology for Internet of Things, Venkatesh Upadrista, Apress Edition 2021 [3] Artificial Intelligence in IoT, Fadi Al-Turjman, Springer Edition 2019 [4] Raspberry Pi et Windows IoT Core, Alexandre Svetec, Eni Edition 2017

Department of Computer Science

C-3.2.1 Advanced Object-Oriented Programming

Last Update Date: February 17, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Advanced Computing 1</td><td>Course</td><td colspan="2">Object Oriented Programming (OOP)</td><td>Code</td><td>C-3.2.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">3</td></tr>
<tr><td>Workshop</td><td>21h00</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>-</td></tr>
<tr><td>Self Study</td><td>42h00</td><td colspan="2"></td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Mohamed Amine Omrani</td><td>Involved Teacher(s)</td><td colspan="3">Mr Mohamed Amine Omrani</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="3">P1 : Systems Specification and Design Methodologies
P2: Implementation Techniques
P3: Innovation, Research and Emerging Technologies</td></tr>
<tr><td colspan="2"><u>Prerequisites:</u></td><td colspan="6">Procedural Programming: Mastery of fundamental object oriented programming concepts, such as class and object, constructors, attributs, methods, class relationship, Interface, etc.
Basic Algorithmic Concepts: Understanding of simple algorithms, the structure of a program, and programming logic.</td></tr>
<tr><td colspan="2"><u>Corequisites:</u></td><td colspan="6">Programming Languages: Familiarity with at least one programming language. Even though Java is the language used for the course, having a foundation in any language facilitates the learning of OOP (Object-Oriented Programming) concepts.
Knowledge in discrete mathematics and algorithmics.</td></tr>
<tr><td colspan="2"><u>Learning outcomes</u></td><td colspan="6">Upon successful completion of this course, students will:

1. Advanced Object-Oriented Concepts: Demonstrate an in-depth understanding of advanced OOP concepts, including design patterns, advanced inheritance, and dynamic polymorphism, etc..
2. Graphical User Interface (GUI) Mastery: Develop complex and aesthetically pleasing user interfaces using AWT, Swing, and JavaFX, showcasing advanced UI design principles.
3. Database Integration Expertise: Effectively integrate and manipulate relational databases using JDBC, demonstrating proficiency in complex database queries, transactions, and database design.</td></tr>
</table>

	4. Design patterns: Analyze and implement design patterns such as MVP, MVVM, MVC, and MVW, applying them to real-world scenarios and complex application development. 5. Master advanced Java features like generics and lambda expressions, and gain expertise in concurrent programming. 6. Advanced Java Application Development: Develop highly scalable and efficient Java applications, including proficiency in handling large data sets, optimizing performance, and ensuring security.
Content:	**Chapter 1: Review of Object-Oriented Programming** 1. Why OOP 2. Review of the basic notions of OOP 3. Review of the JAVA execution environment. **Chapter 2: Advanced Notions of OOP** 1. Introduction 2. JAVA SE and JAVA EE 3. Types of attributes and methods (Default, Static, ..) 4. Static objects. 5. Interfaces, overloading, and APIs. 6. Advanced inheritance and polymorphism 7. Exception mechanisms. 8. Operating principle of exceptions. 9. Handling multiple exceptions. 10. Complex relationships between classes. 11. Data collections in JAVA. **Chapter 3: Advanced Graphical User Interfaces (GUI) Development** 1. Introduction 2. Custom UI components (AWT, SWING, JAVA FX) 3. Layout managers and event handling 4. Designing responsive and accessible interfaces **Chapter 4: Database Management** 1. JDBC Driver and techniques 2. Connection pooling and Request handling 3. DAO 4. JAVA and MySql 5. NoSQL database integration (optional) **Chapter 5: Software Architecture and Design Patterns** 1. Real-world application of MVP, MVVM, MVC, and MVW 2. Design patterns (e.g., Layered Architecture, Microservices) 3. Scalability and performance considerations **Chapter 6: Parallel Processing and Concurrency in Java** 1. Introduction to multithreading 2. Synchronization and thread safety 3. Java concurrency utilities (e.g., Executor framework, Concurrent collections) 4. Asynchronous programming with CompletableFuture 5. Best practices and patterns for multithreaded programming **Chapter 7: Advanced Java Features and Techniques** 1. Generics and parameterized types

	2. Reflection and annotations 3. Lambda expressions and functional programming in Java 4. Method references and functional interfaces 5. Advanced I/O operations and NIO.2 **Project and Self-study details : Development of a complete JAVA application and best practices.** **Project Topic**: Develop a desktop java application for example "Student Management System". **Project Scope:** Develop a comprehensive Student Management System that enables educational institutions to efficiently manage student records, enrollment, course registration, and grading. The system should be designed with a user-friendly GUI (AWT/SWING), utilize a relational database for data storage, and follow a software architectural pattern (e.g., MVC) to ensure modularity and maintainability. **Key Features and Requirements:** User Authentication: Implement user authentication and authorization for administrators, instructors, and students. Student Records Management: Create, read, update, and delete (CRUD) operations for student profiles, including personal information, contact details, and enrollment status. Course Registration: Allow students to register for courses and instructors to assign students to courses. Grading System: Implement a grading system that records and calculates student grades for each course, including the ability to view and print transcripts. User-Friendly GUI: Design an intuitive and aesthetically pleasing GUI using JavaFX or Swing, providing an efficient and enjoyable user experience. Database Integration: Utilize JDBC to connect to a relational database for data storage and retrieval. Implement data validation and error handling. Software Architecture: Choose and justify a software architectural pattern (e.g., MVC) for the project. Organize the codebase accordingly to ensure maintainability and extensibility.
Teaching and learning method(s):	Face to face 42 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lecture with workshops and projects
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1 and 2.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 3, 4, 5 and 6.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain advanced Object-Oriented Programming (OOP) concepts with precision, encompassing design patterns, advanced inheritance mechanisms, and dynamic polymorphism. Articulate the intricacies of reflective programming and introspection, showcasing a comprehensive grasp of these intricate OOP facets. Apply these conceptual foundations to practical coding scenarios, ensuring a holistic understanding of advanced OOP principles.

	By the end of the final exam, students should design and construct a complete object-oriented application in Java, effectively implementing object-oriented techniques such as handling collections, exception management, and the proficient use of encapsulation, inheritance, and polymorphism. Demonstrate a deep understanding of the advantages and challenges of object-oriented programming. Showcase their ability to develop efficient Java desktop applications, proficiently handling substantial data sets.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Proficiency in advanced Object-Oriented Programming (OOP) concepts, including design patterns, advanced inheritance, and dynamic polymorphism. 2. Ability to apply advanced OOP principles in practical coding scenarios, demonstrating in-depth understanding. 3. Mastery of advanced Java programming concepts, syntax, and libraries. 4. Competence in designing and architecting complex software systems using advanced software architectural patterns (e.g., MVP, MVVM, MVC). 5. Expertise in seamlessly integrating advanced Java features like generics and lambda expressions, and gain expertise in concurrent programming. 6. Proficiency in managing relational databases using JDBC. 7. Ability to develop Java application performance, including efficient handling of relational databases.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Course support
References:	[1] Granet, Vincent, and Regourd. "Java: Aide-mémoire." Paris: Dunod, 2019. [2] Gervais, Luc. "Apprendre la programmation orientée objet avec le langage Java: Avec exercices pratiques et corrigés." St Herblain: Éditions ENI, 2018. [3] "Éditions ENI Java Learning Resources." [Online] Available: https://www.editions-eni.fr/open/mediabook.aspx?idR=1b69cad39002ebda5534e8406946c701 [4] "Java Documentation by Oracle." [Online] Available: http://docs.oracle.com/javase/ [5] "The Java Language Specification, Third Edition." [Online] Available: http://java.sun.com/docs/books/jls/ [6] "The Java Virtual Machine Specification, Second Edition." [Online] Available: http://java.sun.docs/books/jvms [7] Martin Fowler, "Patterns of Enterprise Application Architecture," Addison-Wesley, 2002. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software," Addison-Wesley, 1994.

Department of Computer Science

C-3.2.2-Advanced Database and DBMS

Last Update Date: June 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Advanced Computing 1</td><td>Course</td><td colspan="2">Advanced Database and DBMS</td><td>Code</td><td>C-3.2.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>10,5h</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">3</td></tr>
<tr><td>Workshop</td><td>21h</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>10,5h</td><td rowspan="2">Coefficient</td><td rowspan="2">2</td></tr>
<tr><td>Self Study</td><td>14h</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h / Semester</td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Mohsen Maraoui</td><td>Involved Teacher(s)</td><td colspan="2">Mr Mohsen Maraoui</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="2">P1 : Systems Specification and Design Methodologies
P2: Implementation Techniques
P3: Innovation, Research and Emerging Technologies</td></tr>
<tr><td colspan="2"><u>Prerequisites:</u></td><td colspan="6">SQL Language and Database Management Systems (DBMS)</td></tr>
<tr><td colspan="2"><u>Corequisites:</u></td><td colspan="6">Database Course.</td></tr>
<tr><td colspan="2"><u>Learning outcomes</u></td><td colspan="6">At the end of this course, the student should master the concepts related to the administration of a relational database, including:
1. Understanding the architecture of logical and physical databases.
2. Managing logical spaces (tablespaces) and users.
3. Managing privileges and profiles.
4. Backing up and restoring the database.
5. Hybrid databases.
6. Advanced Databases</td></tr>
<tr><td colspan="2"><u>Content:</u></td><td colspan="6">Chapter 1: Fundamental Database Concepts</td></tr>
</table>

	1. Data Models 2. Normalization 3. Data Manipulation Languages (SQL) **Chapter 2: Advanced SQL** 1. SQL optimization techniques. 2. Stored procedures, functions, and triggers for automation and data integrity. **Chapter 3: Transaction Management** 1. ACID properties (Atomicity, Consistency, Isolation, Durability). 2. Concurrency control mechanisms: locking, timestamp ordering, optimistic concurrency control. 3. Transaction isolation levels and their implications. **Chapter 4: Database Architecture and Internals** 1. Storage management and indexing strategies. 2. Buffer management and its role in performance optimization. 3. Query processing steps and optimization techniques. **Chapter 5: Distributed Databases** 1. Concepts of distributed databases and their advantages. 2. Distributed transaction management. 3. Data replication strategies and fragmentation techniques. **Chapter 6: Advanced Database Design** 1. Denormalization techniques for improving query performance. 2. Advanced normalization strategies for complex data models. 3. Design considerations for scaling and optimizing large-scale databases. **Chapter 7: NoSQL Databases** 1. Introduction to NoSQL databases: MongoDB and Cassandra. 2. Use cases and scenarios suitable for NoSQL solutions. 3. **Workshop Task**: Design and implement a social media analytics platform using MongoDB. **Objective**: Students will learn to model data in MongoDB to store social media interactions (e.g., posts, comments, likes) and perform queries to analyze trends and user engagement. **Skills:** Data modeling in a document-oriented database, querying MongoDB using aggregation framework, understanding scalability and performance considerations in NoSQL databases. **Chapter 8: NewSQL Databases**

	1. Overview of NewSQL databases: CockroachDB and Google Spanner. 2. Balancing scalability with traditional SQL features. 3. **Workshop Task:** Set up a distributed e-commerce platform using CockroachDB. **Objective:** Students will deploy a distributed database with CockroachDB, simulate transactional data (e.g., orders, inventory), and implement SQL queries to maintain data consistency across multiple nodes. **Skills:** Understanding distributed SQL databases, configuring and managing distributed transactions, ensuring high availability and fault tolerance in a NewSQL environment. **Chapter 9: Graph Databases** 1. Introduction to graph databases: Neo4j. 2. Managing complex relationships and interconnected data. 3. **Workshop Task:** Build a recommendation system using Neo4j. **Objective:** Students will model user preferences and item relationships in a graph database, develop algorithms (e.g., collaborative filtering, PageRank) to generate personalized recommendations, and visualize the graph structure for insights. **Skills:** Graph database modeling, implementing graph algorithms, leveraging traversal and pattern matching queries in Neo4j, interpreting and presenting graph-based insights. **Chapter 10: Cloud Database Deployment** 1. Hands-on experience with cloud-based database services: a. Amazon RDS b. Azure SQL Database c. Google Cloud SQL 2. Emphasis on deployment, management, and scaling in cloud environments. 3. **Workshop Task:** Deploy and manage a multi-region database using Amazon RDS or Azure SQL Database. **Objective:** Students will learn to provision a cloud-based database instance, configure replication across multiple regions for high availability, and implement scaling strategies (e.g., vertical and horizontal scaling) based on workload demands. **Skills:** Cloud database provisioning and management, configuring replication and failover mechanisms, optimizing performance and cost-effectiveness in a cloud environment.
Teaching and learning method(s):	Face to face (face-to-face) 42 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work

Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. • The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,2,3 and 5** • The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 4, 5, 6, 7, 8, 9 and 10**
Assessment criteria(s) :	By the midterm exam, students should possess a comprehensive understanding of the foundational principles of DBMS and its architectural intricacies. Upon completion of the final exam, students should be proficient in designing and implementing a robust hybrid and advanced databases
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: • Mid-term written test (D.S): 20% • Practical work: 20% • Final written exam: 60%.
Target skill(s):	**1.** Database Administration Proficiency **2.** Architectural Understanding of Oracle Server **3.** Database Physical Structure Management **4.** Logical Structure Management **5.** Database Backup and Restoration Skills **6.** Object-Relational Model Mastery
Teaching Manual:	• Powerpoint presentation • Tutorial booklet • Workbooks
References:	[1] : Shio Kumar Singh, Database Systems Concepts, Designs and Application, Pearson Education, Second Edition, 2011. [2] :Ambler, Scott W. Agile Database Techniques: Effective Strategies for the Agile Software developer. New York, NY: John Wiley & Sons (2003). ISBN 0-471-20283-5 [3] :Allen, Sharon and Terry, Evan, Beginning Relational Data Modeling, 2nd edition, Berkeley, CA: APress (2005): ISBN 1-59059-463-0. [4] :Chrucher, Clare. Beginning Database Design: From Novice to Professional. New York, NY: APress (2007). ISBN 978-1-59059-769-9 [5] :Bradshaw, Shannon; Brazil, Eoin; and Chodorow, Kristina. MongoDB: The Definitive Guide, 3rd edition, Sebastopol, CA: O'Reilly Media (2020): ISBN 978-1491954461.

[6] :Kleppmann, Martin. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems, Sebastopol, CA: O'Reilly Media (2017): ISBN 978-1449373320.

[7] :Robinson, Ian; Webber, Jim; and Eifrem, Emil. Graph Databases: New Opportunities for Connected Data, 2nd edition, Sebastopol, CA: O'Reilly Media (2015): ISBN 978-1491930892.

[8] :Erl, Thomas; Mahmood, Zaigham; and Puttini, Ricardo. Cloud Computing: Concepts, Technology & Architecture, Upper Saddle River, NJ: Prentice Hall (2013): ISBN 978-0133387520.

Department of Computer Science

C-3.3.1-Web Development 3 Symfony/Laravel

Last Update Date: February 17, 2024

Syllabus Sheet

Unit	**Web and Mobile 1**		**Course**	**Web Development 3 Symfony/Laravel**		**Code**	**C-3.3.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**10,5h**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**3**
	Workshop	**21h**					
	Project	**10,5h**		**Mixed**	**XX**		
	Self Study	**14h**					
Total workload	**56h / Semester**				**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher	**Mr Mohamed Amine Omrani**			**Involved Teacher(s)**	**Mr Nafaa Haffar**		
Review committee	**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**			**Objective**	**Obj 1**		
				PLO	**P1 : Systems Specification and Design Methodologies P2: Implementation Techniques P4: Web and Software Development**		

Prerequisites:	Basic knowledge of PHP programming. Understanding of Object-Oriented Programming (OOP) concepts. Familiarity with web development basics (HTML, CSS, JavaScript).
Corequisites:	Database Management Systems. Server Management & Deployment
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Understand the structure and components of the Symfony framework. 2. Develop and deploy web applications using Symfony. 3. Integrate databases and manage data using Doctrine. 4. Implement authentication and authorization techniques in Symfony. 5. Customize and extend Symfony using bundles and plugins. 6. Develop APIs using Symfony. 7. Optimize and scale Symfony applications for production. 8. Understand best practices for secure web development with Symfony. 9. Test and debug applications built with Symfony. 10. Integrate third-party tools and services into Symfony applications.

Content:

To enhance the consistency of your course content and provide additional depth, you could add side sections to each chapter as follows:

Chapter 1: Introduction to Symfony

1. Basics of Symfony
2. Symfony versus other PHP frameworks
3. Installing Symfony
    a. Step-by-step installation guide
    b. Setting up a development environment
    c. Troubleshooting common installation issues
4. First Symfony serve project

Chapter 2: Symfony Structure and Components

1. Symfony directory structure
2. DKey Symfony components (HttpFoundation, Routing, etc.)
3. Exploring Symfony's core architecture
4. Understanding the role of bundles and configuration files
5. Overview of dependency injection and service container

Chapter 3: Developing Web Applications

1. Controllers, Routes, and Views
2. Twig templating engine
3. Building a basic Symfony application
4. Creating routes and controllers
5. Integrating Twig templates for dynamic content
6. Implementing layouts and reusable components

Chapter 4: Database Integration with Doctrine

1. Configuring databases
2. CRUD operations
3. Data relationships
4. Implementing entity mappings and associations
5. One-to-One, One-to-Many, Many-to-Many relationships
6. Doctrine Query Language (DQL) for advanced database queries

Chapter 5: Security in Symfony

1. User authentication
2. Authorization and roles
3. Secure coding best practices
4. Implementing security features in Symfony applications
    a. Configuring firewalls and access control lists (ACLs)
    b. Protecting against common security vulnerabilities (e.g., CSRF, XSS)
5. Advanced Security Measures with Symfony (Keycloak Integration)

Chapter 6: Extending Symfony

1. Bundles and plugins
2. Building, extending and using services
3. Developing custom Symfony bundles
4. Creating reusable components and functionalities
5. Leveraging Symfony's service container for dependency injection

Chapter 7: API Development

1. RESTful web services

	2. Serialization and data transformation 3. Building RESTful APIs with Symfony 4. Implementing CRUD operations for API resources 5. Serializing data using Symfony's serializer component **Chapter 8: Performance and Scaling** 1. Caching techniques 2. Profiling and optimization 3. Improving Symfony application performance 4. Utilizing caching strategies for improved response times 5. Profiling and identifying performance bottlenecks **Chapter 9: Testing and Debugging** 1. Unit and functional testing 2. Debugging tools in Symfony 3. Writing tests for Symfony applications 4. Using PHPUnit for unit and functional testing 5. Debugging Symfony applications with built-in tools (e.g., Symfony Profiler) **Chapter 10: Advanced Symfony Integration** 1. Integrating third-party tools and libraries 2. Continuous Integration/Continuous Deployment (CI/CD) for Symfony 3. Streamlining development workflows with Symfony 4. Integrating popular libraries and frameworks (e.g., Guzzle, Monolog) 5. Setting up CI/CD pipelines for automated testing and deployment
	Workshops: 1. Setting up a new Symfony project. 2. Building a basic CRUD web application. 3. Implementing user registration and authentication. 4. Building a RESTful API. 5. Optimizing a Symfony application. 6. Testing a Symfony application.
	Self-Study Research/Project Subject: Build a Secure Web App with Symfony and Keycloak Integration. Create a web app using Symfony, focusing on security and scalability. Integrate Keycloak for advanced user authentication and authorization. Ensure protection against common security risks like CSRF and XSS. Optimize performance with caching and profiling. Develop RESTful APIs for seamless integration. Document your process and explore advanced Symfony features like CI/CD.
Teaching and learning method(s):	Face to face 42 hours.

	A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,2 and 3.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 4, 5, 6, 7, and 8.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain the fundamental structure, components, and application development basics of Symfony. By the end of the final exam, students should be able to demonstrate proficiency in advanced features, security best practices, API development, and optimization techniques in Symfony.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Symfony application development proficiency. 2. Mastery of web application security in Symfony. 3. Proficiency in API development and integration. 4. Optimization and scaling of web applications. 5. Advanced troubleshooting and debugging. 6. Efficient use of Symfony plugins and extensions.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] The Official Symfony Documentation Team, Symfony: The Fast Track, 6th ed. SensioLabs, 2023. ISBN: 978-2-491452-55-0. DOI: 10.1007/978-2-491452-55-1. [2] M. Elkahlouti and R. Weaver, Mastering Symfony 6: Building High-Performance Web Applications, 1st ed. Packt Publishing, 2022. ISBN: 978-1-83850-606-3. DOI: 10.1007/978-1-83850-607-0. [3] D. Buchmann and D. Leech, Symfony 6 Cookbook, 4th ed. Packt Publishing, 2023. ISBN: 978-1-83922-548-5. DOI: 10.1007/978-1-83922-549-2. [4] C. Pitt and C. Stultiens, Symfony 6 Essentials, 2nd ed. Manning Publications, 2022. ISBN: 978-1-4919-9043-5. DOI: 10.1007/978-1-4919-9044-2. [5] Official Symfony Documentation. Available at https://symfony.com/doc/current/index.html

Department of Computer Science

C-3.3.2-Native Mobile Development (Android/IOS)

Last Update Date: February 17, 2024

Syllabus Sheet

Unit	**Web and Mobile 1**		**Course**	**Native Mobile Development (Android/IOS)**		**Code**	**C-3.3.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**10,5h**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**3**
	Workshop	**21h00**					
	Project	**10,5h**		**Mixed**	**XX**		
	Self Study	**14h00**					
Total workload	**56h00 / Semester**				**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher	**Mr Mohamed Amine Omrani**			**Involved Teacher(s)**	**Mr**		
Review committee	**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**			**Objective**	**Obj 1**		
				PLO	**P1 : Systems Specification and Design Methodologies P2: Implementation Techniques P4: Web and Software Development**		

Prerequisites:	Understanding of object-oriented programming concepts
Corequisites:	Programming Languages: Familiarity with at least one object oriented language. Even though KOTLIN is the language used for the course.
Learning outcomes	Upon completion of this teaching module, the student will be able to: **1.** Understand the Android architecture and ecosystem. **2.** Design and implement user interfaces for Android devices. **3.** Develop efficient and responsive applications using Android Studio. **4.** Integrate Android apps with web services and databases.. **5.** Implement multimedia features, including audio, video, and image processing **6.** Implement internationalization and localization for multiple languages. **7.** Test and debug Android applications. **8.** Understand advanced Android features, such as background services, sensors and notifications.
Content:	**Chapter 1: Introduction to Android Development** 1. State of the Art of Mobile Applications 2. Specificities of mobile development 3. App monetization 4. Android platform architecture

Workshop: Android Studio installation and setup

Chapter 2: KOTLIN Basics

1. Introduction
2. Variables and constants
3. Functions
4. Conditional structures
5. Loops
6. Arrays
7. Classes

Workshop: Developing a simple management application using KOTLIN.

Chapter 3: User Interface Design

1. Android layouts and views
2. XML layout design
3. User interface best practices
4. Adapters and Dynamic Layout
5. Fragments
6. Jetpack compose

Workshops:

- Create a simple discount calculation application using basic graphic elements (constraintLayout, button, seekBar, EditText...) and DataBinding
- Create the same application using Jetpack Compose technique

Chapter 4: Activity and Intent

1. Activities and their lifecycle
2. Implicit and explicit intents
3. Passing data between activities
4. Activity navigation
5. Inflate menu

Workshop : Create an application that allows to create a quiz game containing multiple choice question regarding the flags of different countries

Chapter 5: Data persistence

1. Shared Preferences
2. Internal Storage
3. External Storage
4. SQLite Databases
5. Network storage
6. Broadcast, ContentProvider

Workshop: Create a directory application allowing management and backup information (last name, first name, email, telephone number, address, etc.) concerning people.

Chapter 6: Networking and Web Services

1. HTTP requests with HttpURLConnection or Retrofit
2. Parsing JSON data
3. Handling network operations asynchronously
4. Error handling and network security

Workshop: Develop the MeteoApp application which uses the free REST web services of the OpenWeatherMap.org site to get the weather forecast

Chapter 7: Services and Threads

	1. Types of Services 2. Service Life Cycle 3. Threads on Android 4. Coroutines **Chapter 8: Working with Sensors** 1. Introduction to Android libraries 2. Integrating Google Maps API 3. Using third-party libraries **Practical Work:** Creating a Location-Based Service App. —---------- **Self-Study Research/Project Subject:** ● Development of a Social Media Integration App on Android: Challenges and Best Practices : Students can explore the various challenges faced while integrating popular social media platforms into an Android app and determine the best practices to overcome these challenges. ● Participate in Android developer forums, communities, and groups. Ask questions and seek advice from experienced developers. ● Get familiar with version control systems like Git. GitHub or GitLab can be used to store and manage your code. ● Learn how to debug your apps and write unit tests. This is crucial for identifying and fixing issues. ● Utilize online resources, such as tutorials, courses, and documentation from https://developer.android.com/
Teaching and learning method(s):	Face to face 42 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,2 and 3** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 4, 5, 6, 7 and 8**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain the Android architecture, basic UI design, application components, and basic persistence mechanisms. By the end of the final exam, students should be able to design and build complex Android apps, integrate advanced features like broadcast, services, coroutines, sensors,, test and debug their applications, and understand the publication process.
Assessment measure(s):	Assigning of a final average

Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Proficient use of Android Studio and its tools. 2. UI/UX design for mobile interfaces. 3. Backend integration and data management for mobile apps. 4. Multimedia processing in mobile environments. 5. App optimization and debugging techniques.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Documentation and tutorials on the official android website : developer.android.com [2] Michael Fazio, Kotlin and Android Development featuring Jetpack, The Pragmatic Programmers, 2021. [3] B. Goetz, B. Burdon, D. Kelsey, and C. Stewart, Android Studio Hedgehog Essentials: Fully Updated for Android Studio Hedgehog (2023.1.1) and the New UI, O'Reilly Media, 2023.

Department of Computer Science

C-3.4.1-System and Network Administration

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Network, System and Information Security</td><td>Course</td><td colspan="2">System and Network Administration</td><td>Code</td><td>C-3.4.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="4">Semester</td><td rowspan="4">3</td></tr>
<tr><td>Workshop</td><td>10,5h</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>25h</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mme Elhem Blel</td><td>Involved Teacher(s)</td><td colspan="3">Mme Elhem Blel</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="3">P2: Implementation Techniques
P5: Network and Information Systems Management</td></tr>
<tr><td>Prerequisites:</td><td colspan="7">History of systems architecture and components and OSI/TCP Models and ip subnetting</td></tr>
<tr><td>Corequisites:</td><td colspan="7"></td></tr>
<tr><td>Learning outcomes</td><td colspan="7">involve the management, maintenance, and optimization of computer systems, networks, and infrastructure to ensure their reliability, security, and performance.</td></tr>
<tr><td>Content:</td><td colspan="7">1. System Architecture: Understand the architecture of a Linux system, including the boot process, kernel, init system,
2. Linux Installation and Package Management: Install a Linux system, manage software packages, and work with package managers like apt and yum.
3. GNU and Unix Commands: Familiarity with essential Linux commands and their usage.
4. Devices, Linux Filesystems, Filesystem Hierarchy Standard: Managing storage devices, filesystems, and adhering to the standard filesystem hierarchy.
5. Linux Shells, Scripting, and Data Management: Work with Linux shells, write scriptls, and manage data through commands and utilities.
6. Administrative Tasks: Perform common administrative tasks like managing users and groups, managing printers, and more
7. Essential System Services: Configure and manage system services like SSH, HTTP, and email.
8. Networking Fundamentals: Understand networking concepts, configure network interfaces, and troubleshoot network issues.</td></tr>
</table>

	9. System Maintenance: Implement backups, manage user accounts, and troubleshoot system problems. 10. Domain Name Server: Set up and configure DNS servers. 11. Web Services: Install and configure web servers (Apache…) . 12. Network Client Management: Configure network clients and services,
Teaching and learning method(s):	Face to face (face-to-face) 32 hours. Course LPIC official courses and labs.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S), a practice test and a final written test (Exam) are planned.
Assessment criteria(s) :	By the end of the final exam, students should be able to involve the management, maintenance, and optimization of computer systems, networks, and infrastructure to ensure their reliability, security, and performance.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	Student will be familiar with Linux os in management, maintenance, and optimization of computer systems, networks, and infrastructure
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	LPIC 101 LPIC 102 LPIC 201 LPIC 202

Department of Computer Science
C-3.4.2-Information Systems Security

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Network, System and Information Security**		**Course**	**Information Systems Security**		**Code**	**C-3.4.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Software Engineering**
Type of education	**Lecture**	**21h**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**3**
	Workshop	**10,5h**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**25h**					
Total workload	**56h / Semester**				**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher	**Mr Wael Sellami**		**Involved Teacher(s)**	**Mr Wael Sellami**			
Review committee	**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1**			
			PLO	**P1 : Systems Specification and Design Methodologies P5: Network and Information Systems Management**			

Prerequisites:	Familiarity with basic programming concepts. Knowledge of fundamental computer networking. Understanding of operating systems.
Corequisites:	System and Network Administration
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Recognize the importance of foundational security principles 2. Distinguish between foundational threats, categorizing various attack types, and understanding the landscape of malicious software. 3. Analyze and apply security measures, including operating system security, virtualization concepts, and physical security controls. 4. Explore network security technologies and tools for securing both wired and cloud-based environments, encompassing firewalls, IDS, and honeypots. 5. Implement strategies for addressing application vulnerabilities 6. Understand cryptography principles, digital signatures, certificates, and potential vulnerabilities and attacks
Content:	**Chapter 1: Introduction to Information Systems Security** 1. Definitions and Associated Fields 2. Significance and Pertinence 3. Foundational Security Principles and Objectives 4. Training Programs and Certification Paths

5. Policies, Legal Compliance, and Regulatory Frameworks

Chapter 2: Security Risks and Threats

1. Foundational Threats: Vulnerabilities and Risk Assessment
2. Categories of Attacks: Social Engineering, Network-Based, Application/Service, and Other Application Vulnerabilities and Attack Techniques
3. Malicious Software (Malware): Exploring Viruses, Ransomware, Worms, Trojans, Spyware/Adware, and Rootkits

Practical Work:

TP1: Threat Assessment and Analysis

Perform a web application security scan using OWASP ZAP, Analyze scan results, Discuss mitigation strategies based on findings.

Chapter 3: Infrastructure Security for Computers

1. Operating System (OS) Security (Applications and System) and Vendor Recommendations
2. Understanding Virtualization Concepts
3. Security Applications and Tools
4. Strategies for Securing Hardware and Peripherals
5. Physical Security Measures and Facility Controls

Practical Work:

TP2: Hardening Operating Systems

Scan a Linux system with OpenVAS to identify vulnerabilities and to analyze configuration

Chapter 4: Infrastructure Security for Networks

1. Network Design Elements and Considerations
2. Comprehensive Network Operations: IP Addresses, Ports, URLs, and Internet Connectivity
3. Safeguarding Cloud and Server Environments
4. Strategies for Securing Wired Networks and Associated Devices

Practical Work:

TP3: *Network Security Setup*

Configure network security components and analyze network traffic.

Chapter 5: Network Security Technologies, Tools, and Approaches

1. Delving into Firewalls, VPNs, Routers, Gateways, and Access Points
2. The Role of Honeypots and Honeynets in Network Security
3. Network Intrusion Protection Systems (NIPS) and Network Intrusion Detection Systems (NIDS)
4. Utilizing Command-Line Tools for Network Security

Practical Work: *Firewalls and*

TP4: Intrusion Detection

Install and configure Snort for intrusion detection, create custom Snort rules.

TP5: Firewalls

Set up IPTables rules for network security

Chapter 6: Risk Management

1. Addressing Application Vulnerabilities and Security Risks
2. Distinguishing Between Penetration Testing and Vulnerability Scanning
3. Effective Vulnerability Management Strategies
4. Identity and Access Management Principles
5. Strategies for Redundancy and Disaster Recovery
6. Implementing Monitoring and Auditing Protocols

Chapter 7: Cryptography and Public Key Infrastructure (PKI)

1. Real-World Use Cases, Including Authentication and Nonrepudiation

	2. Exploring Symmetric and Asymmetric Cryptographic Algorithms 3. Concealment Techniques: Steganography and Obfuscation 4. Understanding Digital Signatures, Certificates, and PKI 5. Vulnerabilities and Attacks in Cryptography **Practical Work:** TP6: Cryptographic Tools and Key Management Encrypt and decrypt data using OpenSSL, generate digital signatures and certificates and explore key management practices.
Teaching and learning method(s):	Face to face (face-to-face) 31.5 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2and 3** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes ., 4, 5 and 6**
Assessment criteria(s) :	During the midterm exam, students should be able to demonstrate a solid grasp of foundational concepts, including definitions, significance, principles, training programs, and legal compliance in information security. They should be able to categorize and differentiate security risks, threats, vulnerabilities, and various attack techniques effectively. They should exhibit their understanding of security aspects related to computer infrastructure, virtualization, security tools, hardware, and facility controls. By the end of the final exam, students should be able to showcase their proficiency in infrastructure security, network design, and the secure operation of network environments. Students should provide effective solutions for addressing application vulnerabilities. They should analyze and evaluate identity and access management, redundancy, disaster recovery, and monitoring protocols. Students should be able to exhibit a deep understanding of cryptographic principles, encompassing authentication, symmetric and asymmetric cryptography, digital signatures, certificates, and the identification of vulnerabilities in cryptography.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	**1.** Analyse a problem, identify, and define the computing requirements appropriate to its solution **2.** Design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs **3.** Recognize the need for and engage in continuous professional development. **4.** Apply current techniques, concepts, skills, tools, and best practices used in the core information technologies

Teaching Manual:	• PowerPoint presentation • Tutorial booklet • Workbooks
References:	[1] M. Stamp, Information Security: Principles and Practice" by Mark Stamp, 2016 [2] W. Stallings, Network Security Essentials: Applications and Standards, 2017 [3] W. Stallings, Cryptography and Network Security: Principles and Practice, 2020

Department of Computer Science
C-3.5.1-Project Management

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Transversal Unit 2</td><td>Course</td><td colspan="2">Project Management</td><td>Code</td><td>C-3.5.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>-</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="4">Semester</td><td rowspan="4">3</td></tr>
<tr><td>Workshop</td><td>21h00</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>-</td></tr>
<tr><td>Self Study</td><td>35h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mme Mouna Hjiri</td><td>Involved Teacher(s)</td><td colspan="2">Mme Mouna Hjiri</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 2 & obj 3</td></tr>
<tr><td>PLO</td><td colspan="2">P6 : Project Management and Leadership
P8: Communication and Collaboration
P9: Lifelong Learning and Adaptability</td></tr>
<tr><td colspan="2"><u>Prerequisites:</u></td><td colspan="6">Ensure that students have a basic understanding of their respective engineering disciplines and are comfortable with fundamental concepts.</td></tr>
<tr><td colspan="2"><u>Corequisites:</u></td><td colspan="6">Mathematics Courses: Engineering students often require a strong foundation in mathematics. Corequisites or recommended courses might include introductory calculus or linear algebra.
Computer Science Fundamentals: For computer science students, corequisites in introductory programming or data structures can be beneficial.
Basic Electrical Engineering: For electrical engineering students, corequisites in basic electrical circuits and electronics might be useful.
Technical Communication: A course in technical communication can help students develop written and oral communication skills, which are essential in project management.
General Education Requirements: Depending on your institution's curriculum, there might be general education courses or requirements that students need to fulfill during their first year.</td></tr>
<tr><td colspan="2"><u>Learning outcomes</u></td><td colspan="6">Upon completion of this teaching course, the student will be able to:
- Effectively plan, execute, and control engineering projects while adhering to ethical and professional standards.
- Develop a Work Breakdown Structure (WBS) and Gantt charts to better Project Control and Quality Management based on change management processes.</td></tr>
</table>

	- Apply risk management techniques to identify, assess, and mitigate project risks aiming to resolve issues and conflicts within the project. - Demonstrate professionalism and integrity in project management activities. - Consider the Legal and Regulatory Compliance unique requirements of the engineering discipline. - Explore opportunities for innovation and sustainable practices in engineering project management.
Content:	**Chapter 1:** Introduction to Project Management 1. Definition and importance of project management 2. Key project management processes 3. Project management in engineering disciplines **Chapter 2**: Project Planning 1. Project identification and selection 2. Stakeholder analysis and communication 3. Project charter and objectives 4. Work breakdown structure (WBS) 5. Gantt charts and project scheduling 6. Resource allocation and cost estimation 7. Identifying and assessing project risks 8. Risk mitigation and contingency planning 9. Risk monitoring and control **Chapter 3**: Project Execution, monitoring and control 1. Team formation and development 2. Project communication and reporting 3. Change management and quality assurance 4. Performance metrics and key performance indicators (KPIs) 5. Tracking project progress and making adjustments 6. Issue and scope change management 7. Project closure activities 8. Lessons learned and knowledge transfer 9. Final project presentations **Practical Work:** Students will work in teams to apply the concepts learned in class to a practical project relevant to computer science or electrical engineering. The project will involve planning, executing, and closing a small project, with documentation and a final presentation. **Self-study:** Students will: - Tailor their self-study plan to align with the specific concerned sub-discipline of engineering whether computer science, electrical, or another area. - Start with introductory books or online resources to grasp the fundamental concepts of project management. They can use resources like the Project Management Institute's (PMI) Project Management Body of Knowledge (PMBOK) or PRINCE2 (Projects IN Controlled Environments) guides.

	- Consider pursuing industry-recognized certifications like the Project Management Professional (PMP) or Certified associate in project management (CAPM) offered by PMI …
Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work, activities.
Assessment method(s):	A midterm written test (D.S) is planned to ensure the evaluation of the **learning outcomes** achievement level. A practical project / portfolio presentation to evaluate: - Class Participation and Engagement - Final project report and presentation - Reflective report - Faculty advisor's assessment and feedback.
Assessment criteria(s) :	**Knowledge and understanding:** Students should be able to: - Demonstrates understanding of key project management principles, processes, and terminology. - Define project scope, objectives, constraints, and stakeholders and create a detailed project plan, including a work breakdown structure (WBS) and Gantt chart. - Identify and assesses project risks and uncertainties. - Understands the role of leadership and team dynamics in project execution. - Can effectively monitor project progress, track milestones, and identify variances. - Can make data-driven decisions to control project performance. - Knowledge of quality assurance and control techniques. - Can identify and address quality issues in a project. **Skills:** - Effective written and verbal communication for project reporting and stakeholder engagement. - Can resolve conflicts and motivate team members. - Ability to identify and address issues and challenges that arise during project execution. - Analytical skills for root cause analysis and decision-making. - Can manage project schedules and timelines effectively. - Ability to critically evaluate project scenarios and make informed decisions. - Skill in creating clear and comprehensive project documentation, including project charters and closure reports. - Ability to reflect on personal strengths and weaknesses as a project manager. - Can provide constructive feedback to peers regarding their project management skills.
Assessment measure(s):	Assigning of a final average: Indirect assessment, exercises, peer to peer assessment, weekly assignments.
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final examination: 60%.

Target skill(s):	1. Understand the core principles of project management. 2. Define project scope, objectives, and constraints. 3. Develop a project plan, including tasks, resources, and timelines. 4. Apply project scheduling and cost estimation techniques. 5. Identify and manage project risks. 6. Monitor and control project progress. 7. Execute and close a project, including documentation and stakeholder communication.
Teaching Manual:	● PowerPoint presentation ● Lecturing ● Tutorials ● Courses project Lecturing
References:	[1] "Project Management: The Basics for Success" by Professional Education and Training Services. [2] "A Guide to the Project Management Body of Knowledge (PMBOK Guide)" by Project Management Institute (PMI). [3] "Project Management for Engineering, Business, and Technology" by John M. Nicholas and Herman Steyn. [4] "Engineering Project Management" by Nigel J. Smith and Mark A. Nicholas. [5] "The Art of Project Management" by Scott Berkun

Department of Computer Science
C-3.5.2-English TOEIC 2

Last Update Date: February 17, 2024

Syllabus Sheet

Unit	**Transversal Unit 3**		**Course**	**English TOEIC 2**		**Code**	**C-3.5.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**4**
	Workshop	**-**					
	Project	**-**		**Mixed**	**XX**		
	Self Study	**35h00**					
Total workload	**56h00 / Semester**			**Coefficient**		**2**	
				ECTS credit		**2**	
Responsible Teacher	**Mme Besma Ben Ismail**		**Involved Teacher(s)**	**Mme Besma Ben Ismail**			
Review committee	**-Prof. PHD. MOUNIR ZRIGUI** **-M.A. PHD. M.A. OMRANI** **-M.A. PHD. WAEL SELLAMI** **- Director of study A. SELMI**		**Objective**	**Obj 2 & Obj 3**			
			PLO	**P8 : Communication and Collaboration** **P9: Lifelong Learning and Adaptability**			

Prerequisites:	No formal prerequisites are usually required to take the TOEIC Listening and Reading test. It is primarily designed for individuals and organizations to assess and measure English language proficiency for professional or business purposes.
Corequisites:	None.
Learning outcomes	Upon completion of this teaching course, the student will be able to: **1.** Understand the main points of clear standard speech on familiar matters regularly encountered in work, school, leisure, etc. **2.** Understand texts that consist of high-frequency everyday or job-related language. This includes understanding the main ideas, factual information, and details in written texts. **3.** Build a moderate vocabulary range and demonstrating the ability to use it effectively in both written and spoken contexts. **4.** Understand and being able to apply a wide range of grammatical structures accurately, including complex sentence forms and verb tenses. **5.** Effectively express oneself in a variety of contexts, including social and work-related situations, by utilizing appropriate vocabulary and grammar. **6.** Demonstrate the capability to write clear, well-structured texts on various topics, including essays, formal letters, emails, and reports.

	7. Understand common business terms and language used in professional contexts, including meetings, presentations, negotiations, and other workplace-related situations.
Content:	**Chapter 1: Listening Comprehension** Section 1: Listening to Academic English 1. Practice listening to lectures, podcasts, and online courses in English related to computer science and electrical engineering. 2. Take notes while listening and summarize key points. Section 2: TOEIC Listening Skills 1. Practice listening to TOEIC-style conversations and lectures. 2. Focus on understanding details, main ideas, and inferences. 3. Take practice tests to assess progress. Section 3: Mock TOEIC Listening Tests 1. Take full-length mock TOEIC listening tests under timed conditions. 2. Review and analyze your mistakes to identify areas for improvement. **Chapter 2: Reading Comprehension** Section 1: Academic Reading 1. Read academic papers, journals, and textbooks related to your field in English. 2. Identify and understand key concepts, arguments, and technical terms. Section 2: TOEIC Reading Skills 1. Practice reading TOEIC-style business documents, reports, and emails. 2. Develop skills in locating information, understanding context, and making inferences. Section 3: Mock TOEIC Reading Tests 1. Take full-length mock TOEIC reading tests under timed conditions. 2. Analyze your performance and work on time management. **Chapter 3: Speaking (Optional)** 1. Engage in conversation with peers or language exchange partners to improve your spoken English skills. 2. Simulate a TOEIC Speaking Test by discussing various topics, expressing opinions, and giving short presentations. **Chapter 4: Writing (Optional)** 1. Write essays, reports, and summaries in English related to computer science and electrical engineering topics. 2. Focus on clarity, organization, and effective communication. **Practical Work:** - Practicing TOIEC 2 tests - Running groups and peer to peer conversations **Self-study:** Students should: - Familiarize themselves with the TOEIC format, which typically consists of two sections: Listening Comprehension and Reading Comprehension. - Purchase or download official TOEIC preparation materials from the ETS (Educational Testing Service) website.

	- Improve Listening Skills - Regularly take full-length practice TOEIC tests to become familiar with the format and to gauge your progress. Review your answers and understand the explanations for any questions you got wrong.
Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work , activities , T. B. L And P.B.L.
Assessment method(s):	A midterm written test (D.S) is planned to ensure the evaluation of the **learning outcomes** achievement level. A practical project / portfolio presentation to evaluate: - Class Participation and Engagement - Individual and Group Presentations - Written Assignments and Reports
Assessment criteria(s) :	Students proficiency should be at an advanced level, demonstrating the ability to understand complex technical materials, express your ideas clearly and concisely, and engage in professional discussions in English.
Assessment measure(s):	Assigning of a final average: Placement test, Indirect assessment, Exercises, Peer to peer assessment, Recording a conversation, Weekly assignments, and Quizzes.
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final examination assessing writing skills and comprehension : 60%.
Target skill(s):	- Advanced Listening Comprehension - Advanced Reading Comprehension - Specialized Vocabulary - Advanced Grammar and Syntax - Technical Writing Skills - Presentation and Speaking Skills - Critical Thinking and Analytical Skills - Time Management and Test Strategy
Teaching Manual:	● PowerPoint presentation ● Lecturing ● Tutorials ● Courses project Lecturing

References:	[1] "Barron's TOEIC: Test of English for International Communication" by Lin Lougheed: Barron's TOEIC prep book is a popular choice and covers both the listening and reading sections with ample practice tests and exercises. [2] "Longman Preparation Series for the New TOEIC Test: More Practice Tests" by Lin Lougheed: This series by Lin Lougheed is known for its practice tests that mimic the actual TOEIC format. [3] "TOEIC Listening and Reading Test Prep Plus" by Kaplan: Kaplan's TOEIC test prep book provides comprehensive coverage of the test content and includes practice questions and audio exercises. [4] "Delta's Key to the Next Generation TOEIC Test" by Nancy Lin and Lin Lougheed: This book is designed for students aiming for higher TOEIC scores and includes advanced practice materials. [5] Official ETS TOEIC Practice Tests: ETS offers official TOEIC practice tests that you can purchase on their website. These tests provide a realistic simulation of the actual exam.

Department of Computer Science

C-3.5.3-Personal Development and Soft Skills

Last Update Date: February 17, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Transversal Unit 3</td><td>Course</td><td colspan="2">Personal Development and Soft Skills</td><td>Code</td><td>C-3.5.3</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">4</td></tr>
<tr><td>Workshop</td><td>-</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>-</td></tr>
<tr><td>Self Study</td><td>35h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mme Mouna Hjiri</td><td>Involved Teacher(s)</td><td colspan="2">Mme Mouna Hjiri</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 2 & obj 3</td></tr>
<tr><td>PLO</td><td colspan="2">P3 : Innovation, Research and Emerging Technologies
P9: Lifelong Learning and Adaptability
P8: Communication and Collaboration</td></tr>
<tr><td><u>Prerequisites:</u></td><td colspan="7">None: Since this course primarily focuses on personal development and soft skills, it may not have strict academic prerequisites. However, it's important that students are in their second year, as they should have a foundational understanding of their respective technical disciplines.</td></tr>
<tr><td><u>Corequisites:</u></td><td colspan="7">Professional Development Course: This course can help students apply what they learn in the soft skills course to their technical field.
Technical Writing or Communication Course: A technical writing or communication course can complement the soft skills course by further developing students' written and verbal communication abilities.
Team Project Course or workshops: It can help students apply the teamwork and collaboration skills they learn in the soft skills course in a real-world context.</td></tr>
<tr><td><u>Learning outcomes:</u></td><td colspan="7">1. Upon completion of this teaching course, the student will be able to acquire:
2. should be able to apply critical thinking and analytical skills to complex problems, both in personal and professional contexts.
3. Develop a high level of self-awareness, understanding one's strengths, weaknesses, and personal values, and using this knowledge to make informed decisions.</td></tr>
</table>

	4. Develop a high level of emotional intelligence (EQ) which includes understanding and managing one's emotions, as well as effectively recognizing and managing the emotions of others. 5. Demonstrate advanced communication skills, both written and verbal, with an ability to adapt communication style to diverse audiences and situations. 6. Understand the principles of effective leadership and teamwork, with the ability to lead teams, coordinate tasks, and facilitate group processes. 7. Effectively manage time and prioritize tasks, demonstrating strong organizational skills and adaptability. 8. Make informed decisions, assess risks, and demonstrate good judgment in various personal and professional situations. 9. Develop advanced conflict resolution skills, including negotiation and mediation techniques.
Content:	**Chapter 1: Introduction to Personal Development and Soft Skills** 1. Overview of the course and its relevance. 2. Introduction to EQF Level 7. 3. Setting personal goals and expectations. **Chapter 2: Leadership and Management** 1. Leadership styles and qualities. 2. Project management and leadership. 3. Leadership and ethics in tech. 4. Team dynamics and roles. 5. Conflict resolution and effective collaboration. 6. Group project assignment. **Chapter 3: Effective Communication Skills** 1. Verbal and non-verbal communication. 2. Writing and presenting technical information. 3. Active listening and feedback. 4. Emotional Intelligence : Understanding and managing emotions. 5. Stress management and resilience. 6. Empathy and interpersonal skills. **Chapter 4: Time Management and Productivity** 1. Setting priorities and managing time. 2. Procrastination and time-wasting habits. 3. Tools and techniques for time management. **Chapter 5: Problem-Solving and Critical Thinking** 1. Creative problem-solving techniques. 2. Critical thinking and decision-making. 3. Case studies and problem-solving exercises. **Chapter 6: Ethical and Professional Responsibilities** 1. Ethical considerations in computer science and electrical engineering. 2. Professional development and networking.

	3. Course reflection and individual development plans. **Practical Work:** - Students will apply their communication skills in a real-world scenario, such as a group presentation or a technical report. - Peer and instructor evaluations. - Course reflection and feedback. **Self-study:** Students will: - Define goals for personal development and identify the specific soft skills you want to improve, and set realistic, measurable targets. - Create a structured plan that includes the topics they want to cover, the resources they will use, and the time they will allocate to each. - Practice Regularly: Find ways to integrate the skills you're learning into their daily life. - Networking and Collaboration: Engage with individuals who have expertise in the skills they want to develop. - Utilize Technology: Leverage online platforms, virtual communities, and applications specifically designed to enhance soft skills. - Receive Feedback: Seek feedback from mentors, peers, or professionals in their field. - Constructive criticism can provide valuable insights and help you refine their skills. - Engage in activities that promote self-awareness and mindfulness, such as meditation, journaling, and self-reflection.
Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work.
Assessment method(s):	A midterm written test (D.S) is planned to ensure the evaluation of the **learning outcomes** achievement level. A practical project / portfolio presentation to evaluate: - Class Participation and Engagement - Individual and Group Presentations - Written Assignments and Reports - Individual development plan and reflection
Assessment criteria(s) :	- Self-awareness - Emotional intelligence - Adaptability - Team work and collaboration - Presentation and public speaking.
Assessment measure(s):	Assigning of a final average: Assignments, Quizzes, Exams, Assignments, Quizzes, Exams.

Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	- Understand the importance of personal development and soft skills in the fields of computer science and electrical engineering. - Develop effective communication skills, both written and verbal. - Enhance teamwork and collaboration abilities. - Improve leadership and management skills. - Learn time management techniques for better productivity. - Develop emotional intelligence to manage stress and work well with others. - Cultivate problem-solving and critical thinking abilities. - Explore ethical and professional responsibilities in the tech industry.
Teaching Manual:	● PowerPoint presentation ● Lecturing ● Tutorials ● Courses project Lecturing
References:	[1] Emotional Intelligence" by Daniel Goleman [2] "The 7 Habits of Highly Effective People" by Stephen R. Covey [3] "Crucial Conversations" by Al Switzler, Joseph Grenny, and Ron McMillan

Department of Computer Science
NESC-3.6.1-Local Area Networks

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Networks and Embedded System 1</td><td>Course</td><td colspan="2">Local Area Networks</td><td>Code</td><td>NESC-3.6.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Networks and Embedded System</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td rowspan="2">--</td><td rowspan="4">Semester</td><td rowspan="4">3</td></tr>
<tr><td>Workshop</td><td>21h 00</td></tr>
<tr><td>Project</td><td>-</td><td rowspan="2">Mixed</td></tr>
<tr><td>Self Study</td><td>42h00</td><td>XX</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="2"></td><td colspan="3"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Wael Sellami</td><td>Involved Teacher(s)</td><td colspan="3">Mme ElHEM Blel</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="3">P1 : Systems Specification and Design Methodologies
P2: Implementation Techniques
P3: Innovation, Research and Emerging Technologies</td></tr>
<tr><td colspan="2"><u>Prerequisites:</u></td><td colspan="6">Data Communication Basics
Computer Architecture
TCP/IP Networking Knowledge</td></tr>
<tr><td colspan="2"><u>Corequisites:</u></td><td colspan="6">Emerging Technologies in Networking
Network Simulation and Modeling
Advanced Networking Protocols</td></tr>
<tr><td colspan="2"><u>Learning outcomes</u></td><td colspan="6">Upon completion of this teaching module, the student will be able to:

1. Analyze and understand the principles of LAN design, including network topologies and architectures.
2. Evaluate LAN protocols and technologies, including Ethernet, VLANs, and wireless LANs.
3. Design and implement efficient and secure LAN solutions.
4. Troubleshoot LAN issues and optimize network performance.
5. Explore emerging LAN technologies and trends.</td></tr>
<tr><td colspan="2"><u>Content:</u></td><td colspan="6">Chapter 1: Introduction to Local Area Networks

1. Overview of LANs, their role in network infrastructure, and historical development.</td></tr>
</table>

2. LAN topologies, architectures, and design considerations.
3. Network cabling and physical layer standards.

Chapter2: Ethernet LANs

1. In-depth study of Ethernet technologies, including Ethernet frames and addressing.
2. LAN switching, spanning tree protocol, and VLANs.
3. Practical configuration and management of Ethernet LANs.

Chapter 3: Wireless LANs (WLANs)

1. Introduction to WLAN technologies, including Wi-Fi standards and operation.
2. WLAN security and encryption.
3. Configuring and securing WLANs.

Chapter 4: LAN Routing and Interoperability

1. LAN routing protocols and their role in LAN-to-LAN communication.
2. Interconnecting LANs with routers and routing protocols.
3. LAN segmentation and traffic management.

Chapter 5: LAN Security

1. LAN security challenges and vulnerabilities.
2. Implementing LAN security measures, including firewalls, intrusion detection systems, and access control.
3. Security best practices for LANs.

Chapter 6: LAN Troubleshooting and Optimization

1. Diagnosing and troubleshooting LAN issues.
2. Performance optimization and monitoring.
3. Implementing Quality of Service (QoS) in LANs.

Chapter 7: Emerging LAN Technologies

1. Exploration of emerging LAN technologies, such as Software-Defined Networking (SDN) and Network Function Virtualization (NFV).
2. Trends in LAN design and infrastructure.

—----------

Project : Students undertake a substantial project related to LAN design, implementation, or optimization, demonstrating their ability to apply knowledge to real-world scenarios**.**

—----------

Self-study: Basic Networking Concepts

Review fundamental networking concepts, including the OSI model, TCP/IP, and LAN topologies. Familiarize yourself with Ethernet standards and basic LAN protocols.

Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2 and 3** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes , 4 and 5**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain the principles of LAN design, including network topologies and architectures. By the end of the final exam, students should be able to design and implement efficient and secure LAN solutions
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. an advanced understanding of Local Area Networks (LANs) and their role in modern network infrastructure. 2. It covers LAN design, protocols, security, and practical implementation.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] "Computer Networks" by Andrew S. Tanenbaum and David J. Wetherall [2] "Ethernet: The Definitive Guide" by Charles E. Spurgeon

Department of Computer Science
NES-C-3.6.2-Service Administration

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Networks and Embedded System 1</td><td>Course</td><td colspan="2">Service Administration</td><td>Code</td><td>NES-C-3.6.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Networks and Embedded System</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">3</td></tr>
<tr><td>Workshop</td><td>21h 00</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>--</td><td rowspan="2">Coefficient</td><td rowspan="2">3</td></tr>
<tr><td>Self Study</td><td>42h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">84h00 / Semester</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Wael Sellami</td><td>Involved Teacher(s)</td><td colspan="3">Mr Hatem Jarboua</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="3">P1 : Systems Specification and Design Methodologies
P2: Implementation Techniques
P3: Innovation, Research and Emerging Technologies</td></tr>
<tr><td colspan="2"><u>Prerequisites:</u></td><td colspan="6">Familiarity with basic UNIX/Linux concepts
Basic scripting skills</td></tr>
<tr><td colspan="2"><u>Corequisites:</u></td><td colspan="6">Operating Systems Fundamentals
Network Administration
Security in Information Technology</td></tr>
<tr><td colspan="2"><u>Learning outcomes</u></td><td colspan="6">Upon completion of this teaching module, the student will be able to:

1. Analyze and evaluate the architecture and design of UNIX-based server systems.
2. Implement advanced security measures to safeguard UNIX servers and services.
3. Optimize the performance of UNIX services through tuning and resource management.
4. Automate routine administrative tasks using shell scripting and other relevant tools.
5. Plan and execute server migrations, updates, and backup strategies.
6. Diagnose and resolve complex issues related to UNIX services and servers.
7. Design and deploy high-availability solutions for critical services.
8. Collaborate effectively with cross-functional teams and communicate technical information clearly.</td></tr>
</table>

Content:

Chapter 1: UNIX Server Fundamentals

1. Introduction to UNIX operating systems
2. UNIX server architecture and components
3. UNIX file systems and permissions
4. Basic server administration tasks

Chapter 2: Security and Access Control

1. Security principles and best practices
2. User and group management
3. SSH configuration and secure remote access
4. Firewalls and intrusion detection

Chapter 3: Performance Optimization

1. Performance monitoring and analysis
2. Resource management and tuning
3. Disk I/O optimization
4. Network optimization

Chapter 4: Scripting and Automation

1. Shell scripting fundamentals
2. Task automation with cron and systemd
3. Managing system configurations
4. Package management and updates

Chapter 5: Backup and Disaster Recovery

1. Data backup strategies
2. Implementing regular backups
3. Disaster recovery planning
4. Data restoration and verification

Chapter 6: Advanced Troubleshooting

1. Troubleshooting methodologies
2. Debugging and diagnostics tools
3. Handling system crashes and errors
4. Incident response and resolution

Chapter 7: High Availability

1. Concepts of high availability
2. Clustering and failover solutions
3. Load balancing for web services
4. Ensuring service continuity

Chapter 8: Collaboration and Communication

1. Team collaboration and best practices
2. Documenting administrative tasks
3. Effective communication and reporting

—----------

Project :

students can explore a research topic like "Scalability and Performance Optimization in UNIX Service Administration." This project can involve optimizing UNIX service configurations for performance and scaling, analyzing real-world scenarios, and proposing solutions for better service administration.

—----------

Self-study:

"Enhancing Security in UNIX Service Administration"

	In this self-study project, students will delve into the intricacies of enhancing security in UNIX service administration. The project aims to provide a comprehensive understanding of advanced security practices within the context of UNIX-based server environments
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,2,3 and 4** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 5, 6, 7 and 8.**
Assessment criteria(s) :	Assessment Type: Final Examination and Project Presentation 1. Knowledge and Understanding: 2. Critical Analysis: 3. Practical Skills: 4. Communication and Documentation actical Skills:
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	**1.** Server configuration and administration. **2.** Security assessment and implementation. **3.** Troubleshooting and problem-solving. **4.** Data backup and recovery planning. **5.** Technology evaluation and selection. **6.** Effective communication and documentation.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] "UNIX and Linux System Administration Handbook" by Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley [2] "UNIX Network Programming" by W. Richard Stevens [3] Online documentation for specific UNIX distributions (e.g., Red Hat, Ubuntu).

Department of Computer Science

SE-C-3.6.1-Advanced Software Engineering

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Software Engineering 1</td><td>Course</td><td colspan="2">Advanced Software Engineering</td><td>Code</td><td>SE-C-3.6.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">3</td></tr>
<tr><td>Workshop</td><td>21h 00</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>42h00</td><td colspan="2"></td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mme Safa Ben Salem</td><td>Involved Teacher(s)</td><td colspan="2">Mme Safa Ben Salem</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="2">P1: Systems Specification and Design Methodologies
P2: Implementation Techniques
P3: Innovation, Research and Emerging Technologies</td></tr>
<tr><td colspan="2">Prerequisites:</td><td colspan="6">Information Systems</td></tr>
<tr><td colspan="2">Corequisites:</td><td colspan="6"></td></tr>
<tr><td colspan="2">Learning outcomes</td><td colspan="6">1. Participants should be able to design complex software systems, identifying and solving advanced design problems.
2. Participants are expected to master the concepts and techniques of advanced software architecture, including the creation of distributed and scalable systems.
3. Participants should be able to implement agile methodologies and other advanced software development approaches.
4. Participants should understand and implement configuration management and continuous integration processes to ensure software stability.</td></tr>
<tr><td colspan="2">Content:</td><td colspan="6">Chapter 1: Software Engineering
Information Systems
Definition of Software Engineering
software crisis
Software quality
Chapter 2: Software life cycle models
Waterfall lifecycle model
V-shaped life cycle model
Spiral life cycle model
Iterative life cycle model</td></tr>
</table>

	Incremental life cycle model **Chapter 3: Agile methods** Prototyping RAID method UP method XP method Scrum method **Chapter 5: Configuration Management and Continuous Integration** **Chapter 4: IT project management using a Scrum tool** **Practical case**
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1 and 2.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 3 and 4.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain the concept of software engineering, present software life cycle models, address the agile software development approach, list the principles and values on which the agile methods and study agile methods: UP, XP and Scrum. At the end of the module, the engineering student will be able to apply the agile development process for the analysis, design and creation of computer applications.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Development of advanced skills in software engineering 2. Project management, ethics, security and communication.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks

References:	[1] Strohmeier A., Buchs D., Software engineering: principles, methods and techniques, Lausanne, Presses polytechniques et universitaire romandes, 1996. [2] Jacobson, Ivar., Booch, Grady and Rumbaugh, James, "The unified software development process", Edition Eyrolles, 2000. [3] Jean-Louis Bénard, Laurent Bossavit, Régis Médina and Dominic Williams, "Extreme Programming: With two case studies", Edition Eyrolles, 2002. [4] Kenneth S. Rubin. Pierre-Alain Muller and Nathalie Gaertner, "Scrum: Agile project management". Pearson Edition 2013. [5] Ken Schwaber and Jeff Sutherland, The Scrum Reference Guide: The Rules of the Game. November 2017 Version and November 2020 Version.

Department of Computer Science
SE-C-3.6.2-Development Platform (.Net)

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Software Engineering 1**		**Course**	**Development Platform (.Net)**		**Code**	**SE-C-3.6.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Software Engineering**
Type of education	**Lecture**	**21h 00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**3**
	Workshop	**21h 00**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**42h00**					
Total workload	**84h00 / Semester**			**Coefficient**	**3**		
				ECTS credit	**3**		
Responsible Teacher	**Mr Mohamed Amine Omrani**		**Involved Teacher(s)**	**Mr Walid Chafroud**			
Review committee	**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1**			
			PLO	**P1 : Systems Specification and Design Methodologies P2: Implementation Techniques P3: Innovation, Research and Emerging Technologies**			

Prerequisites:	Basic Algorithmic Concepts: Understanding of simple algorithms, the structure of a program, and programming logic.
Corequisites:	Programming Languages: Familiarity with at least one programming language Students should have completed an introductory programming course or demonstrate proficiency in at least one programming language. This is essential for understanding the basics of coding before diving into .NET development. Familiarity with database concepts and SQL is beneficial since database operations using Entity Framework
Learning outcomes	On completion of this teaching module, students will be able to: **1.** Understand the fundamentals of the .NET framework. **2.** Develop applications using C#. **3.** Work with Visual Studio, the primary IDE for .NET development. **4.** Create and manage databases using Entity Framework. **5.** Implement web applications with ASP.NET. **6.** Develop desktop applications with Windows Forms and WPF. **7.** Master debugging and error handling in .NET. **8.** Deploy and maintain .NET applications.
Content:	**Module 1: Introduction to .NET Development** 1. Understanding the .NET framework

2. Overview of Common Language Runtime (CLR)
3. Visual Studio and .NET ecosystem

Practical Work: Setting up a development environment with Visual Studio

Module 2: C# and VB.NET Programming

1. Data types and variables
2. Control structures
3. Object-oriented programming concepts

Practical Work:

Writing and debugging basic C# programs

Module 3: .NET Applications Development

1. Building Console Applications
2. Windows Forms Applications
3. Event and user interaction management
4. Windows Presentation Foundation (WPF)

Practical Work:

Developing a Windows Forms application

Module 4: Working with Databases

1. Introduction to Entity Framework
2. Database design and modeling
3. CRUD operations with databases

Practical Work:

Creating a database-driven application using Entity Framework

Module 5: ASP.NET Web Development

1. Introduction to ASP.NET
2. Web Forms and MVC
3. Building web applications

Practical Work:

Creating a basic ASP.NET web application

Module 6: Error Handling and Debugging

1. Exception handling in .NET
2. Debugging tools and techniques
3. Logging and monitoring

Practical Work:

Debugging and error handling in a sample application

Module 7: Deployment and Maintenance

1. Deployment strategies
2. Version control and continuous integration
3. Application maintenance and updates

Practical Work:

Deploying and updating a .NET application

Project: Inventory management application
Self Study : building Executable application and printing report ,use some sound effect and theme design

Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2 and 3.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 5, 6, and 8.**
Assessment criteria(s) :	During the midterm exam, students should be able to Explain what the .NET framework is and its key components and Describe the role of the Common Language Runtime (CLR) in .NET By the end of the final exam, students should be able to Create a Windows Forms application that allows the user to input data and display the result in a graphical interface and Explain the differences between Web Forms and MVC in ASP.NET. Provide use cases for each.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 45%. ● Final Project: 25%
Target skill(s):	**1.** Students will develop a strong proficiency in C# and VB.NET, with the ability to write, debug, and maintain code effectively **2.** Students will gain the skills to create web applications using ASP.NET and desktop applications with Windows Forms and WPF **3.** Students will master debugging techniques and error handling, enabling them to identify and resolve issues in .NET applications. **4.** Students will learn to design and manage databases using Entity Framework, including performing CRUD (Create, Read, Update, Delete) operations. **5.** Students will develop strong problem-solving skills in application development and debugging, enhancing their ability to resolve complex issues.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Albahari, J., & Albahari, B. (2021). "C# 9.0 in a Nutshell: The Definitive Reference." O'Reilly Media. [2] Miles, R. (2021). "C# Yellow Book." Available online. [3] Johnson, B., & Reif, J. (2020). "Professional Visual Studio 2019." Wrox. [4] "Pro ASP.NET Core MVC" by Adam Freeman [5] Lerman, J., & Miller, R. (2020). "Programming Entity Framework: DbContext." O'Reilly Media. [6] APPRENTISSAGE DU LANGAGE C# Serge Tahé - ISTIA - Université d'Angers. [7] Lock, A., & Richards, B. (2018). "ASP.NET Core in Action." Manning Publications.

	[8] Freeman, A. (2021). "Pro ASP.NET Core MVC." Apress.

Department of Computer Science
BDBI-C-3.6.1- Analyze et Reporting

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Big Data and Business Intelligence 1**		**Course**	**Analyze et Reporting**		**Code**	**BDBI-C-3.6.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Big Data and Business Intelligence**
Type of education	**Lecture**	**21h 00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**3**
	Workshop	**21h 00**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**42h00**					
Total workload		**84h00 / Semester**			**Coefficient**		**3**
					ECTS credit		**3**
Responsible Teacher		**Mr Mounir Zrigui**		**Involved Teacher(s)**	**Mr Mohamed Manaa**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1**		
				PLO	**P1 : Systems Specification and Design Methodologies P2: Implementation Techniques P3: Innovation, Research and Emerging Technologies**		

Prerequisites:	
Corequisites:	
Learning outcomes	At the end of this course, the student will be able to: − To understand Data Analysis Concepts − To situate reporting in the life cycle of a project − Identify different tools to use for reporting − Prepare a report on a given project
Content:	**Chapter 1 : Introduction to Data Analysis** Overview of data analysis process Types of data (quantitative, qualitative) Data sources and collection methods

	Chapter 2: Data Cleaning and Preparation Data preprocessing techniques Handling missing data Dealing with outliers Data transformation and normalization **Chapter 3: Descriptive Statistics** Measures of central tendency (mean, median, mode) Measures of dispersion (variance, standard deviation) Frequency distributions and histograms **Chapter 4: Statistical Analysis** Hypothesis testing (t-tests, chi-square tests) Regression analysis (linear regression, logistic regression) Analysis of Variance (ANOVA) **Chapter 5: Advanced Analytical Techniques** Cluster analysis Decision trees and random forests Machine learning algorithms for classification and prediction **Chapter 6: Practical Applications and Case Studies** Real-world examples and case studies from various domains (business, finance, healthcare, etc.) Hands-on exercises and projects to apply analytical techniques to solve problems
Teaching and learning method(s):	Face to face (face-to-face) 42 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2 and 3** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 3, 4 and 5**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain the reporting and reporting tools. By the end of the final exam, students should be able to create an interim report and a final report.

Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Mastery of Data Analysis 2. Creation of Advanced Visual Reports 3. Application of Reporting to Specific Areas
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Analyse et reporting RH, Pierre-Alexandre Grolleau, Editions ENI 2019, ISBN : 978-2409020131 [2] Le reporting financier, Eric Tort, Edition Dunod 2006, ISBN : 978-2-10-048898-8 [3] Comprendre tout le reporting financier, Elisabeth Albertini, Stéphane Lefrancq, Edition Vuibert 2019, ISBN : 978-2311406351 [4] Power BI Desktop De l'analyse de données au reporting, André Meyer Roussilhon, Editions ENI 2019, ISBN :978-2-409-03005-5

Department of Computer Science
BDBI-C-3.6.2-Business Intelligence

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Big Data and Business Intelligence 1**		**Course**	**Business Intelligence**		**Code**	**BDBI-C-3.6.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Big Data and Business Intelligence**
Type of education	**Lecture**	**21h 00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**3**
	Workshop	**21h 00**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**42h00**					
Total workload	**84h00 / Semester**				**Coefficient**		**3**
					ECTS credit		**3**
Responsible Teacher	**Mr Mohamed Amine Omrani**		**Involved Teacher(s)**	**Mr Mohamed Manaa**			
Review committee	**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1**			
			PLO	**P1 : Systems Specification and Design Methodologies P2: Implementation Techniques P3: Innovation, Research and Emerging Technologies**			

Prerequisites:	Databases, Data analysis concepts
Corequisites:	
Learning outcomes	Upon completion of this teaching module, the student will be able to: – Define and explain the fundamental concepts and principles of Business Intelligence. – Apply BI tools and techniques to collect, process, and analyze data for decision support. – Design and create effective data visualizations and dashboards. – Evaluate the performance of BI solutions and recommend improvements. – Discuss ethical and security considerations in BI implementations. – Apply BI in real-world scenarios and solve business problems. – Work collaboratively in teams to develop BI solutions. – Communicate BI insights effectively to technical and non-technical stakeholders. – Manage BI projects and assess their impact on organizational decision-making. – Stay updated with emergiDng trends and technologies in BI.

Content:	**Chapter 1: Introduction to Business Intelligence (Basic Notions)** 1. Definition and importance of BI 2. Historical overview of BI 3. Key concepts: data, information, knowledge 4. The role of BI in decision-making **Chapter 2: Data Warehousing and ETL Processes** 1. Data warehousing concepts and architecture 2. ETL (Extract, Transform, Load) processes 3. Data modeling for BI 4. Data quality and cleansing **Chapter 3: BI Tools and Technologies** 1. Types of BI tools: Reporting, OLAP, Data Mining, and more 2. Comparison of popular BI tools (e.g., Tableau, Power BI, QlikView, MicroStrategy) 3. Data visualization tools and techniques 4. Licensing and cost considerations **Chapter 4: Data Visualization and Dashboards** 1. Principles of data visualization 2. Creating interactive dashboards 3. Designing effective data visualizations 4. Report generation and distribution **Chapter 5: BI Performance Evaluation** 1. Key performance indicators (KPIs) in BI 2. Performance metrics and benchmarking 3. Continuous improvement in BI solutions 4. Case studies in BI performance evaluation
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,2,3,4 and 5.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 6, 7,8,9 and 10.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain fundamental BI concepts and apply BI tools for data collection and analysis. By the end of the final exam, students should be able to evaluate the performance of BI solutions, discuss ethical and security considerations, and apply BI in real-world scenarios to solve business problems.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20%

	● Final written exam: 60%.
Target skill(s):	**1.** Data analysis and interpretation **2.** Data visualization and dashboard creation **3.** Critical thinking and problem-solving **4.** Effective communication of insights
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Business Intelligence et management, Alphonse Carlier, AFNOR Edition 2013 [2] Business Intelligence : le recueil des besoins, Eva POLINI, ENI Edition 2023 [3] Business intelligence Avec Excel, Power BI et Office 365, Jean-Pierre Girardot, ENI Edition 2015

Department of Computer Science

C-4.1.1-Virtualization and Cloud Computing

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Advanced Topic 2</td><td>Course</td><td colspan="2">Virtualization and Cloud Computing</td><td>Code</td><td>C-4.1.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">4</td></tr>
<tr><td>Workshop</td><td>10,5h</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>24h</td><td colspan="2"></td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Wael Sallami</td><td>Involved Teacher(s)</td><td colspan="3">Mr Wael Sallami</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="3">P1: Systems Specification and Design Methodologies
P2 : Implementation Techniques
P3: Innovation, Research and Emerging Technologies
P5: Network and Information Systems Management</td></tr>
<tr><td colspan="2"><u>Prerequisites:</u></td><td colspan="6">Master the concepts of information system, distributed systems and operating systems.
A working knowledge of distributed systems and multi-tier architectures.</td></tr>
<tr><td colspan="2"><u>Corequisites:</u></td><td colspan="6">- Understanding operating systems, their components, and how they manage resources is crucial for cloud computing. An operating systems course can be a valuable corequisite.
- Familiarity with Linux or Unix operating systems is often essential for cloud computing, particularly in cloud server management.</td></tr>
<tr><td colspan="2"><u>Learning outcomes</u></td><td colspan="6">Upon completion of this teaching module, the student will be able to:
1. Understand the fundamental concepts of virtualization, including hypervisor technology, virtual machine management, and resource allocation strategies.
2. Analyze the benefits and challenges of cloud computing, including scalability, resource optimization, security concerns, and cost management.</td></tr>
</table>

	3. Implement virtualization solutions using industry-standard tools and platforms, such as VMware, Hyper-V, or KVM, to optimize hardware utilization and streamline IT operations. 4. Design resilient and scalable cloud architectures, considering factors such as redundancy, fault tolerance, and workload distribution to ensure high availability and performance. 5. Understand the principles of containerization, focusing on Docker technology, including image creation, containerization, and orchestration using Docker Compose and Kubernetes.
Content:	Chapter 1 : Introduction to computer systems 1. Operating system 2. Computer servers 3. Computer Security 4. Distributed systems Chapter 2 : Virtualisation 1. Definition of Virtualization 2. Domains of virtualization 3. Advantages & disadvantages of virtualization 4. Different types of virtualization **Practical Work:** - TP1: Master the different types of virtualization hypervisor, the principle of type 2 virtualization. - TP2: Master the basic concepts of type 1 virtualization. Chapter 3 : Cloud Computing 1. IT Challenges 2. Definition and characteristics of the Cloud 3. Cloud architecture, models and services 4. Security and privacy 5. Economic issues 6. Research axes **Practical Work:** - TP3: Become familiar with Cloud Computing IaaS environments (Virtualbox & OpenStack integration) - TP4: Study and manipulate the advanced features of the OpenStack environment. Chapter 4 : Cloud Services Security 1. Cloud Security 2. National Institute of Standard for Technology (NIST) recommendations 3. Cloud Security Alliance (CSA) recommendations Chapter 5 : Docker 1. Concepts 2. Setup 3. Create and use containers 4. Use and share images 5. Network management 6. Volume management 7. Docker Composer **Practical Work:**

	- TP5: Understand and test the main uses of dockers, master the creation of images and dockerfiles. - TP6: Master the management of networks and volumes of dockers. **Project :** Project Title: "Deployment of a Web Application Using Docker on a Cloud Platform" Project Description: In this project, students will explore Cloud Computing and Docker technologies by deploying a web application on a cloud platform, integrating the concepts of anl hybrid Model for Docker Container Workload Prediction. The primary objective is to analyze the effectiveness and performance implications of this novel hybrid model within cloud environments, particularly focusing on its impact on container management and application deployment processes. This research offers a unique opportunity to investigate the practical applications and benefits of this innovative approach in optimizing Docker container workload prediction for enhanced cloud-based application deployment strategies. **Self-study:** 1. Research Understand virtualization Basics: Understand Cloud Basics: Select a Cloud Platform: Explore Cloud Concepts: Study Use Cases: 2. Proposal Estimate the costs associated with your project. Plan the technical architecture, including the choice of cloud services, scalability, security, and data management. 3. Implementation Sign up for an account with the cloud provider you've selected. Install necessary tools like the command-line interface (CLI) for your chosen cloud platform and Docker (if relevant). 5. Presentation (Optional) Share project findings with the class if desired. 6. Doing Exercise and QCM published in easylearn.ovh platform.
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,.2.. and …3..**

	• The final written test is devoted to the evaluation of the level of achievement of **learning outcomes ., 4,5,6. and 7**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain the concept of virtualization in the context of cloud computing and its role in resource optimization. By the end of the final exam, students should be able to design a cloud-based solution for a given use case, including the selection of appropriate cloud services, architectural components, and scalability considerations.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: • Mid-term written test (D.S): 20% • Practical work: 20% • Final written exam: 60%.
Target skill(s):	1. Differentiating between Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), and understand when to use each model. 2. Understanding public, private, and hybrid cloud deployment models and the ability to select the most suitable option based on specific requirements. 3. Proficiency in container technologies such as Docker
Teaching Manual:	• Powerpoint presentation • Tutorial booklet • Workbooks
References:	[1] Cloud Computing: From Beginning to End, By Ray J. Rafaels, April 2015 [2] Cloud Computing: Concepts, Technology & Architecture and Cloud Computing Design Patterns, Thomas Erl and others, May 2013 [3] Cloud Computing, Sécurité, stratégie d'entreprise et panorama du marché, Guillaume Plouin, Edition DUNOD, 2013 [4] Cloud Computing, Maîtrisez la plateforme AWS, Amazon Web Services, Mathieu Zarrouk, Edition ENI, 2012 [5] VMware vSphere Virtualisation, Eric MAILLE, Edition ENI, 2011 [6] Cloud Computing Applications, Christopher Moyer, Edition Pearson, 2011 [7] Virtualisation des systèmes d'information avec VMware Architecture, Philippe GILLET, Edition ENI, 2010

Department of Computer Science
C-4.1.2-Advanced IoT Systems & Integration

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Advanced Topic 2</td><td>Course</td><td colspan="2">Advanced IoT Systems & Integration</td><td>Code</td><td>C-4.1.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">4</td></tr>
<tr><td>Workshop</td><td>10,5h</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>--</td><td rowspan="2">Coefficient</td><td rowspan="2">2</td></tr>
<tr><td>Self Study</td><td>24h00</td><td colspan="2">56h00 / Semester</td></tr>
<tr><td colspan="3">Total workload</td><td colspan="2">ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Walid ben mabrouk</td><td>Involved Teacher(s)</td><td colspan="2">Mr Walid ben mabrouk</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="2">P1: Systems Specification and Design Methodologies
P2 : Implementation Techniques
P3: Innovation, Research and Emerging Technologies
P5: Network and Information Systems Management</td></tr>
<tr><td><u>Prerequisites:</u></td><td colspan="7">● Proficiency in programming languages such as Python, Java, or C/C++.
● Understanding of basic networking concepts including TCP/IP, HTTP, and MQTT.
● Knowledge of hardware components and sensors commonly used in IoT systems.
● Familiarity with data structures and algorithms.
● Previous coursework or experience in Internet of Things (IoT) fundamentals.</td></tr>
<tr><td><u>Corequisites:</u></td><td colspan="7">- Concurrent enrollment in courses covering advanced topics in networking, such as network security or wireless communication protocols.
- Practical experience with IoT development platforms and tools, such as Arduino, Raspberry Pi, or similar embedded systems.
- Understanding of cloud computing principles and platforms, as IoT systems often rely on cloud-based services for data storage and processing.
- Knowledge of data analytics and visualization techniques, as IoT systems generate large volumes of data that require analysis and interpretation.</td></tr>
</table>

Learning outcomes	Upon completion of this course, students will be able to: ● Proficiently apply advanced networking concepts to optimize data transmission, latency reduction, and reliability enhancement in IoT systems. ● Design and implement secure and scalable IoT solutions by integrating diverse hardware devices, sensors, actuators, and communication protocols. ● Utilize cloud computing platforms for efficient data storage, processing, and analytics in IoT environments, ensuring scalability and cost-effectiveness. ● Evaluate emerging trends and technologies in IoT, such as edge computing and machine learning, to anticipate their impact on future IoT deployments and applications.
Content:	**Lectures :** 1. **Introduction to the Internet of Things (IoT)** ● Overview of IoT ● Historical Development of IoT ● Significance of IoT in Electrical Engineering 2. **Hardware Components of IoT: Sensors, Actuators, and Microcontrollers** ● Sensors in IoT ● Actuators for Control ● Microcontrollers for IoT 3. **IoT Communication Protocols: MQTT, CoAP, LoRa, NB-IoT, etc.** ● Introduction to IoT Protocols ● MQTT for IoT ● CoAP for IoT ● LoRa and NB-IoT Technologies 4. **Design and Programming of IoT Devices** ● Designing IoT Devices ● Programming Platforms (e.g., Arduino, Raspberry Pi) ● IoT Programming Languages (C/C++, Python) 5. **IoT Data Management: Storage, Analysis, and Visualization** ● Managing IoT Data ● Data Storage Solutions ● Data Analysis Tools ● Data Visualization Techniques 6. **IoT Applications in Electrical Engineering** ● Smart Home Automation ● Industrial Automation ● Energy Management ● Remote Monitoring 7. **Security and Privacy in IoT** ● IoT Security Challenges ● Privacy Concerns ● Encryption and Access Control ● Secure Data Transmission **Tutorial Sessions (TD):** ● In TD sessions, students will engage in in-depth discussions and analysis. The following topics will be covered: 1. Understanding IoT Concepts 2. Analyzing IoT Use Cases

	3. Evaluating Real-World IoT Applications 4. Assessing IoT Protocol Suitability for Different Scenarios 5. Identifying IoT Security Vulnerabilities 6. Proposing Security Measures **Lab Sessions (TP):** ● During Lab sessions (TP), students will gain practical hands-on experience. The following topics will be covered: 1. Interfacing with Sensors and Data Acquisition 2. Controlling Actuators and Receiving Feedback 3. Programming Microcontrollers for IoT Devices 4. Building IoT Prototypes and Systems **Self-Study (Auto-Apprentissage):** ● As part of self-study, students are encouraged to explore further readings, research papers, and online resources related to IoT, especially in the context of electrical engineering. This self-study will help students: 1. Deepen their understanding of IoT concepts 2. Stay updated with the rapidly evolving IoT field 3. Conduct research on specific IoT applications and case studies relevant to electrical engineering
Teaching and learning method(s):	The course will be taught using a combination of lectures, discussions, laboratories, and projects. Students will have the opportunity to learn about the IoT concepts and technologies through hands-on experience in the laboratory.
Teaching technique(s):	● In-person lecture course or other to be specified: Tutorials, Practical work, Accelerated practical training for mastering the software and hardware tool, Individual work
Assessment method(s):	A mid-term written test (DS) and a final written test (Exam) are planned. • The mid-term written test will be devoted to evaluating the level of achievement of learning outcomes 1, 2, 3 and 4. ● • The final written test is devoted to evaluating the level of achievement of learning outcomes on the content of the teaching module.
Assessment criteria(s) :	● Midterm exam: Students will be assessed on their knowledge of the IoT concepts and technologies covered in the first half of the course. ● Final exam: Students will be assessed on their knowledge of the IoT concepts and technologies covered in the entire course. ● Laboratory experiments: Students will be assessed on their ability to design, implement, and test IoT systems in the laboratory. ● Project: Students will be assessed on their ability to develop and deploy an IoT application for a real-world problem.
Assessment measure(s):	Allocation of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated according to the following formula: ● Mid-term written test: 20% ● TP grade: 30% (20% Attendance/30% Formative Assessment/50% TP Exam ● Final written exam: 60%.

Target skill(s):	1. Mastery of networking concepts including optimizing data transmission, minimizing latency, and ensuring reliability in IoT systems. 2. Ability to design and implement secure IoT solutions, integrating various hardware components and communication protocols while addressing cybersecurity concerns. 3. Competence in utilizing cloud computing platforms for efficient data storage, processing, and analytics in IoT environments, ensuring scalability and cost-effectiveness. 4. Capability to evaluate emerging trends and technologies in IoT, such as edge computing and machine learning, to anticipate their impact on future IoT deployments and applications. 5. Skill in collaborating effectively within multidisciplinary teams to plan, develop, and deploy IoT solutions, demonstrating effective communication, problem-solving, and project management abilities.
Teaching Manual:	• Course manual • Guided work manual • Practical work support
References:	**Books:** • Bahga, A., & Madisetti, V. (2014). Internet of Things: Architecture and design principles. Artech House. • Buyya, R., & Dastjerdi, A. V. (2016). Internet of Things: Principles and paradigms. Morgan Kaufmann. • McEwen, A., & Cassimally, H. (2014). Designing the Internet of Things. Pearson Education. • Al-Fuqaha, A., Salah, K., & Mohamed, A. (2015). Internet of things: Architectures, protocols, and applications. CRC Press. • Gupta, R., & Kumar, V. (2018). Internet of things: Architectures, protocols, and applications. Springer. **Articles :** • Aazam, M., & Khan, I. (2016). Internet of things: A survey of enabling technologies, security challenges, and open research issues. IEEE Communications Surveys & Tutorials, 18(4), 2347-2376. • Chawla, S., Agrawal, P., & Kaushik, S. K. (2016). A survey on Internet of Things: Architecture, security, and privacy. IEEE Access, 4, 5889-6010. • Khan, M. A., & Khan, M. I. (2018). Internet of things (IoT): A survey of security challenges and solutions. IEEE Access, 6, 19359-19381.

Department of Computer Science

C-4.2.1-Software Architecture and Design Patterns

Last Update Date: February 19, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Advanced Computing 2</td><td>Course</td><td colspan="2">Software Architecture and Design Patterns</td><td>Code</td><td>C-4.2.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">4</td></tr>
<tr><td>Workshop</td><td>10,5h</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>--</td><td rowspan="2">Coefficient</td><td rowspan="2">2</td></tr>
<tr><td>Self Study</td><td>24h</td><td rowspan="2">Total workload</td><td colspan="2" rowspan="2">56h00 / Semester</td></tr>
<tr><td colspan="2">ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Hatem Sandid</td><td>Involved Teacher(s)</td><td colspan="2">Mr Hatem Sandid</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1 & Obj 3</td></tr>
<tr><td>PLO</td><td colspan="2">P1: Systems Specification and Design Methodologies
P2 : Implementation Techniques
P7: Analytical and Critical Thinking</td></tr>
<tr><td><u>Prerequisites:</u></td><td colspan="7">Background in computer science, which includes data structures and algorithms.
Proficiency in one or more programming languages is essential, as it underpins the course's practical aspects.</td></tr>
<tr><td><u>Corequisites:</u></td><td colspan="7">Familiarity with software engineering principles, such as software development lifecycle, requirements analysis, and testing methodologies, provides a context for applying software architecture and design patterns effectively.</td></tr>
<tr><td><u>Learning outcomes</u></td><td colspan="7">Upon completion of this teaching module, the student will be able to:
1. Analyze and evaluate software architecture principles and design patterns.
2. Apply design patterns to solve software design problems.
3. Understand the relationship between software architecture and design patterns.
4. Choose appropriate architectural styles and patterns for specific projects.
5. Implement creational, structural, and behavioral design patterns.
6. Evaluate software architecture for quality attributes.</td></tr>
<tr><td><u>Content:</u></td><td colspan="7">Chapter 1: Introduction to Software Architecture and Design Patterns
1. Foundations of Software Architecture
2. Introduction to Design Patterns
3. The Role of Software Architecture in Design Patterns</td></tr>
</table>

4. Case Study: Applying Software Architecture and Design Patterns

Tutorial:

TD1: Gain insights into the core principles and significance of software architecture.

Chapter 2: Architectural Styles and Patterns

1. Architectural Styles Overview
2. Evaluating and Choosing Architectural Styles
3. Common Architectural Patterns
4. Group Project: Applying Architectural Styles and Patterns

Tutorial:

TD2: Learn the criteria and decision-making process for selecting the most suitable architectural style for a project.

Chapter 3: Creational, Structural, and Behavioral Design Patterns

1. Creational Design Patterns
2. Structural Design Patterns
3. Behavioral Design Patterns
4. Real-world Applications of Design Patterns

Tutorial:

TD3: Learn to implement creational design patterns like Singleton, Factory Method, and Abstract Factory for object creation.

Chapter 4: Architectural Patterns and Frameworks

1. Layered Architecture Patterns
2. Microservices Architecture
3. Frameworks and Software Product Lines
4. Practical Exercise: Building an Application Using an Architectural Framework

Tutorial:

TD4: Gain insights into layered architecture patterns and their application in software design.

Chapter 5: Advanced Topics in Software Architecture and Design Patterns

1. Architectural Quality Attributes
2. Emerging Trends in Software Architecture
3. Architectural Patterns for Scalability and Resilience
4. Final Project: Comprehensive Software Design

Tutorial:

TD5: Delve into architectural quality attributes, their impact on system design, and methods for evaluating them.

Chapter 6: Evolutionary Architecture and Continuous Integration

1. Understanding Evolutionary Architecture Principles
2. Evolutionary Design vs. Big Design Up Front (BDUF)
3. Continuous Integration and Delivery in Architecture
4. Case Studies: Implementing Evolutionary Architecture in Real-world Projects

Tutorial:

TD6: Explore the concepts of evolutionary architecture and continuous integration, and practice implementing them in a simulated environment.

Chapter 7: Domain-Driven Design (DDD) and Hexagonal Architecture

1. Principles of Domain-Driven Design
2. Strategic Design with Bounded Contexts and Ubiquitous Language
3. Hexagonal Architecture and Ports-and-Adapters Pattern
4. Applying DDD and Hexagonal Architecture in Complex Systems

	Tutorial: TD7: Dive deep into Domain-Driven Design and Hexagonal Architecture, and learn how to apply them effectively in software projects to manage complexity and ensure maintainability. **Project :** Course Project: Software System Design and Implementation Project Description: For the course project, students will work in teams to design and implement a software system that showcases their understanding of software architecture and design patterns. This project will allow students to apply the concepts learned in the course to a practical, real-world scenario. The project should encompass multiple architectural styles and design patterns, demonstrating a holistic understanding of software design principles. **Self-study:** 1. Research Study programming fundamentals and its relevance in engineering. Collect relevant resources and references. 2. Proposal Choose an engineering problem. Explain how to Design Architectural Solutions 4. Project Report Create a report summarizing the project. Reflect on the experience and insights gained. 5. Presentation (Optional) Share project findings with the class if desired. 6. Engaging in exercises and quizzes on the easylearn.ovh platform provided by Polytech.
Teaching and learning method(s):	Face to face for 32 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2. and 3** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes ., 4,5 and 6**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain architectural styles, analyze the application of design patterns, and discuss the relationship between software architecture and design patterns. By the end of the final exam, students should be able to demonstrate the ability to choose suitable architectural styles and patterns for specific projects, evaluate software architecture for quality attributes, and create comprehensive software designs for real-world applications.

Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	● Critical analysis of software architecture and design patterns. ● Problem-solving using design patterns. ● Selection of appropriate architectural styles and patterns. ● Evaluation of software architecture for quality attributes. ● Staying updated on emerging software architecture trends.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides [2] Software Architecture in Practice, Len Bass, Paul Clements, Rick Kazman [3] Pattern-Oriented Software Architecture: A System of Patterns, Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad [4] Clean Architecture: A Craftsman's Guide to Software Structure and Design, Robert C. Martin

Department of Computer Science

C-4.2.2-Machine Learning and Deep learning

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Advanced Computing 2</td><td>Course</td><td colspan="2">Machine Learning and Deep learning</td><td>Code</td><td>C-4.2.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">4</td></tr>
<tr><td>Workshop</td><td>21h</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>42h</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="3">Total workload</td><td colspan="2">84h00 / Semester</td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Mounir Zrigui</td><td>Involved Teacher(s)</td><td colspan="2">Mr Mounir Zrigui</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1 & Obj 3</td></tr>
<tr><td>PLO</td><td colspan="2">P2 : Implementation Techniques
P3: Innovation, Research and Emerging Technologies
P7: Analytical and Critical Thinking</td></tr>
<tr><td colspan="2">Prerequisites:</td><td colspan="6">Strong foundation in linear algebra, calculus, and probability.
Proficiency in Python programming.
Basic understanding of data analysis and statistics.
Familiarity with fundamental algorithms and data structures.</td></tr>
<tr><td colspan="2">Corequisites:</td><td colspan="6">Advanced Data Analysis and Visualization.
Neural Networks and Their Applications.</td></tr>
<tr><td colspan="2">Learning outcomes</td><td colspan="6">Upon completion of this teaching module, the student will be able to:

1. Grasp the foundational principles and theories of machine learning and deep learning.
2. Implement and evaluate various machine learning algorithms.
3. Design and train deep neural networks for complex real-world applications.
4. Understand the principles of model selection, regularization, and optimization in the context of ML and DL.
5. Apply transfer learning and fine-tuning to leverage pre-trained models.
6. Evaluate model performance and interpret results in a statistically rigorous manner.
7. Understand ethical considerations, biases, and fairness in machine learning.
8. Implement advanced deep learning architectures like CNNs, RNNs, and Transformers.
9. Stay updated with the latest advancements and research trends in the field.</td></tr>
</table>

	10. Deploy and scale ML and DL models in production environments.
Content:	**Chapter 1: Introduction to Machine Learning and Deep Learning** 1. Overview and history 2. Types of learning: supervised, unsupervised, and reinforcement **Chapter 2: Supervised Learning Algorithms** 1. Linear Regression 2. Logistic Regression 3. Support Vector Machines 4. Decision Trees and Random Forests **Chapter 3: Unsupervised Learning and Dimensionality Reduction** 1. Clustering (K-Means, Hierarchical) 2. Principal Component Analysis (PCA) 3. t-SNE and UMAP **Chapter 4: Neural Networks and Deep Learning Fundamentals** 1. Feedforward Neural Networks 2. Backpropagation Algorithm 3. Activation Functions **Chapter 5: Convolutional Neural Networks (CNNs)** 1. Architecture and Principles 2. Application in Image Recognition **Chapter 6: Recurrent Neural Networks (RNNs) and LSTM** 1. Understanding Sequence Data 2. Applications in Natural Language Processing **Chapter 7: Advanced Deep Learning Architectures** 1. Transformer Architectures 2. Generative Adversarial Networks (GANs) 3. Autoencoders **Chapter 8: Model Evaluation and Improvement** 1. Metrics and Evaluation Strategies 2. Regularization Techniques 3. Hyperparameter Tuning **Chapter 9: Ethics, Fairness, and Bias in ML/DL** 1. Ethical considerations in data collection and usage 2. Detecting and mitigating biases 3. Fairness in machine learning **Chapter 10: Deployment, Scalability, and Future Trends** 1. ML/DL in production 2. Scalability challenges

	3. Recent advancements in the field **List Of Workshop:** 1. Building a simple classifier using scikit-learn. 2. Designing a deep neural network with TensorFlow/Keras. 3. Image classification with CNNs. 4. Text generation using LSTMs. 5. Implementing a Transformer for text summarization. 6. Building a GAN for image generation. 7. Bias detection and fairness workshop. 8. Hyperparameter tuning using GridSearch and RandomSearch. 9. Deploying a model using cloud platforms. 10. Workshop on the latest research papers and trends. **Self-Study Research or Project Subject:** "Interpretable Machine Learning: Developing Transparent Deep Learning Models" – Students can explore the challenge of making complex deep learning models more interpretable, ensuring they can be trusted and understood by humans.
Teaching and learning method(s):	Face to face 84 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2, 3 and 4.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 5, 6, 7, 8, 9, and 10.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain foundational concepts, supervised and unsupervised algorithms, and basic deep learning principles. By the end of the final exam, students should be adept at designing complex neural networks, evaluating models rigorously, understanding ethical aspects, and staying updated with advanced ML/DL trends.
Assessment measure(s):	Assigning of a final average

Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Proficiency in implementing and tuning ML/DL algorithms. 2. Deep understanding of neural network architectures. 3. Ability to handle large datasets and derive meaningful insights. 4. Model evaluation and improvement. 5. Ethical and fair model development. 6. Deployment and scaling of ML models.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] "Pattern Recognition and Machine Learning" by Christopher M. Bishop. [2] "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville. [3] "Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili. [4] Online courses from Coursera, Udacity, and other platforms on ML and DL topics.

Department of Computer Science
C-4.3.1-Hybrid Mobile Development

Last Update Date: February 17, 2024

Syllabus Sheet

Unit	**Web and Mobile 2**		**Course**	**Hybrid Mobile Development**		**Code**	**C-4.3.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h 00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**4**
	Workshop	**10,5h**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**25h00**					
Total workload		**56h00 / Semester**			**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher		**Mr Mohamed Amine Omrani**		**Involved Teacher(s)**	**Mr Nafaa Haffar**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1**		
				PLO	**P1: Systems Specification and Design Methodologies P2 : Implementation Techniques P4: Web and Software Development**		

Prerequisites:	Familiarity with mobile app concepts Understanding of object-oriented programming concepts
Corequisites:	Programming Languages: Familiarity with at least one object oriented language. Even though DART is the language used for the course.
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Demonstrate a strong understanding of Flutter and Dart programming language, and the ability to develop cross-platform mobile applications. 2. Create responsive and visually appealing user interfaces using Flutter's widgets and follow design guidelines for Material Design. 3. Apply user experience (UX) and user interface (UI) design principles to create intuitive and aesthetically pleasing mobile apps. 4. Implement navigation and routing in Flutter apps, allowing users to navigate between different screens and pass data between them. 5. Retrieve and display data from RESTful APIs and local databases. 6. Develop custom widgets and apply animations and transitions to enhance user experience. 7. Integrate device features such as camera, GPS, sensors, and permission handling using Flutter plugins. 8. Apply effective debugging techniques in Flutter.

	9. Understand the process of app submission to the Google Play Store and Apple App Store
Content:	**Chapter 1: Introduction to Flutter and Dart** 1. Overview of Flutter and its advantages 2. Setting up Flutter and Dart development environment 3. Writing a simple "Hello, Flutter" app 4. Basic Dart language concepts **Workshop:** Workshop to help students set up Flutter and Dart development environments **Chapter 2: Building User Interfaces with Flutter** 1. Widgets and widget tree 2. Creating responsive layouts 3. Material Design and Cupertino design in Flutter 4. State management in Flutter **Workshop:** ● Guide students through creating a basic Flutter app with a user interface, and buttons. ● Include elements like text input and data display **Chapter 3: Navigation and Routing** 1. Creating navigation structures 2. Implementing routing and navigation 3. Handling navigation parameters 4. Building a multi-page Flutter app **Workshop:** Creating a basic Flutter app with a user interface and navigation between screens. **Chapter 4: Data Management in Flutter** 1. Fetching and displaying data in Flutter apps 2. State management techniques (Provider, BLoC, etc.) 3. Local data storage using SQLite 4. Working with REST APIs **Workshop:** ● Teach students how to fetch data from a remote API (e.g., a weather service) and display it in their Flutter app. ● Build an e-commerce app that displays products, allows users to add items to their cart, and proceed to checkout. Introduce more complex data management and navigation. **Chapter 5: Custom Widgets and Animations** 1. Creating custom widgets 2. Animation basics in Flutter 3. Implementing animations and transitions 4. Gesture recognition and touch events **Workshop:** ● Workshop on designing and customizing Flutter widgets. ● Creating responsive layouts and applying Material Design or Cupertino design principles. **Chapter 6: Working with Device Features** 1. Integrating device features (camera, GPS, sensors) 2. Platform-specific code in Flutter

	3. Permission handling 4. Using plugins to access native features **Workshop:** ● Workshop on integrating device features like the camera, GPS, and sensors. ● Managing permissions and platform-specific code in Flutter. **Chapter 7: Deployment and App Publishing** 1. Building and packaging apps for Android and iOS 2. App distribution on Google Play Store and Apple App Store 3. Managing app updates and version control 4. Monetization options for Flutter apps **Workshop:** Build a travel booking app that integrates with third-party APIs for flight and hotel reservations. Cover complex user interfaces, and performance optimization. —---------- **Self-Study Research/Project Subject:** ● Freestyle Capstone Project: Allow students to propose and develop their own Flutter app project. This project should incorporate a range of advanced Flutter concepts, such as custom plugins, background processing, and offline functionality. Each project should include: ○ Project Requirements: Clearly define the project's objectives, features, and scope. ○ Design Guidelines: Encourage students to follow design principles for a polished user interface. ○ Documentation: Require students to document their code and provide an overview of the project. ○ Presentation: Have students present their projects to the class, showcasing the app's functionality, design, and any challenges they overcame. ● Read Flutter's official documentation. It provides in-depth information about Flutter's features and best practices. ● Learn about UI/UX design principles. ● Join Flutter developer communities on platforms like Reddit, Stack Overflow, and the official Flutter Discord server to ask questions and share your experiences. ● Collaborate on projects with other students. ● Explore open-source Flutter projects on GitHub. Learn a lot by examining other developers' code and contributing to open-source projects.
Teaching and learning method(s):	Face to face 32 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2, 3 and 4** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 5, 6, 7 and 8.**

Assessment criteria(s) :	During the midterm exam, students should be able to understand and explain Flutter basics, including the widget tree, state management, and Flutter's architecture. They should also be proficient in the Dart programming language, including variables, data types, functions, and object-oriented programming principles. Assess their grasp of user interface design principles and Flutter's support for creating responsive and visually appealing layouts. By the end of the final exam, students should be able to retrieve and display data from RESTful APIs and local databases, develop custom widgets and animations, integrate device features and permission handling using Flutter plugins.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Understanding of Flutter's widget-based UI framework. 2. Proficiency in the Dart programming language. 3. Implementation of navigation structures and transitions between screens. 4. Implementation of gesture recognition and touch event handling. 5. Integration of device features like camera, GPS, sensors, and handling of permissions.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Sanjib Sinha, Beginning Flutter 3.0 with Dart, Lean Publishing, 2022 [2] Thomas Bailey and Alessandro Biessek, Flutter for Beginners, Packt Publishing,2021 [3] D. Mourato and G. Siani, Flutter Apprentice: Master Dart and Flutter App Development with Hands-On Projects, 3rd ed. Packt Publishing, 2023. [Online]. Available: docs.flutter.dev. [4] Documentation and tutorials on the Flutter official website: https://flutter.dev/learn

Department of Computer Science

C-4.3.2-Web Development 4 (ClientSide-ReactJS/Angular)

Last Update Date: February 17, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Web and Mobile 2</td><td>Course</td><td colspan="2">Web Development 4 (ClientSide-ReactJS/Angular)</td><td>Code</td><td>C-4.3.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>10,5h</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">4</td></tr>
<tr><td>Workshop</td><td>21h</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>25h</td><td colspan="2"></td></tr>
<tr><td colspan="3">Total workload</td><td colspan="2">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="3"></td><td colspan="2"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Mohamed Amine Omrani</td><td>Involved Teacher(s)</td><td colspan="3">Mr Mohamed Amine Omrani</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="3">P1 : Systems Specification and Design Methodologies
P2: Implementation Techniques
P4: Web and Software Development</td></tr>
<tr><td colspan="2">Prerequisites:</td><td colspan="6">Proficiency in HTML, CSS, and JavaScript.
Basic web notions and development techniques and tool.
Experience with AngularJS or any modern JavaScript framework.
Familiarity with version control systems (e.g., Git).</td></tr>
<tr><td colspan="2">Corequisites:</td><td colspan="6">None</td></tr>
<tr><td colspan="2">Learning outcomes</td><td colspan="6">Upon completion of this teaching module, the student will be able to:

1. Angular Mastery: Demonstrate a deep understanding of Angular, spanning from AngularJS to the latest Angular framework.
2. Component-Based Development: Create complex web applications using Angular's component-based architecture.
3. State Management: Implement advanced state management with Angular services and observables.
4. Optimization and Performance: Optimize Angular applications for superior performance and user experience.
5. Testing and Debugging: Apply comprehensive testing and debugging techniques for robust Angular applications.
6. Security: Implement best practices for securing Angular applications against common vulnerabilities.</td></tr>
</table>

	7. Advanced Routing: Manage complex routing scenarios and lazy loading in Angular applications. **8.** Forms and Reactive Programming: Develop advanced forms and leverage reactive programming in Angular. **9.** Interactivity: Create interactive web components with Angular animations and real-time data updates. **10.** Integration with Backend: Integrate Angular with RESTful APIs and backend services.
Content:	**Chapter 1: Introduction to Angular** 1. Angular history and evolution. 2. Setting up the development environment. 3. Angular architecture and key concepts. 4. Project setup and Angular CLI. **Chapter 2: Components and Templates** 1. Building Angular components. 2. Templates and data binding. 3. Directives and structural directives. 4. Custom pipes and built-in pipes. **Chapter 3: Services and Dependency Injection** 1. Creating and using services. 2. Dependency injection in Angular. 3. Singleton services and providers. 4. Data sharing among components. **Chapter 4: State Management with RxJS** 5. Introduction to reactive programming with RxJS. 6. Observable patterns and operators. 7. State management with BehaviorSubject and NgRx. **Chapter 5: Optimization and Performance** 1. Change detection strategies. 2. Lazy loading and code splitting. 3. Optimizing for mobile and progressive web apps. 4. Performance profiling and optimization techniques. **Chapter 6: Testing and Debugging** 1. Unit testing with Jasmine and Karma. 2. End-to-end testing with Protractor. 3. Debugging Angular applications effectively. 4. Continuous integration and deployment. **Chapter 7: Security in Angular** 1. Cross-Site Scripting (XSS) prevention. 2. Cross-Site Request Forgery (CSRF) protection. 3. Authentication and authorization in Angular. 4. Best practices for securing Angular apps. **Chapter 8: Advanced Routing and Navigation** 1. Advanced routing configuration. 2. Route guards and lazy loading modules. 3. Handling nested routes and route parameters. 4. Navigation and router events. **Chapter 9: Forms and Reactive Programming** 1. Template-driven forms vs. reactive forms.

2. Form validation and custom validators.
3. Reactive programming with Angular Forms.
4. Dynamic forms and form arrays.

Chapter 10: Interactivity and Real-Time Updates

1. Creating interactive UI components.
2. Angular animations and transitions.
3. Real-time data updates with WebSockets.
4. Integrating Angular with real-time databases.

Chapter 11: Integration with Backend Services

1. Consuming RESTful APIs in Angular.
2. Using HTTP interceptors for global error handling.
3. Authentication and authorization with backend services.
4. Data synchronization and caching.

Project & Self Study:

Project: "Advanced E-commerce Platform with Angular"
Project Description:
In this project, students will design and develop an advanced e-commerce web platform using Angular. The goal is to create a feature-rich application that showcases their mastery of Angular concepts, including component-based development, state management, performance optimization, testing, security, and advanced routing.

Project Requirements:
User Authentication: Implement user authentication and authorization using Angular's authentication services.
Product Catalog: Create a dynamic product catalog with advanced search and filtering options.
Shopping Cart: Develop a responsive shopping cart that allows users to add and remove items.
Order Management: Implement order management, including viewing, editing, and confirming orders.
Payment Integration: Integrate a payment gateway for real transactions (sandbox mode).
Performance Optimization: Optimize the application for fast load times and smooth user experience.
Testing: Implement comprehensive unit testing and end-to-end testing for critical features.
Security: Apply security best practices to protect user data and transactions.
Advanced Routing: Set up complex routing, including lazy loading for different sections of the application.
Responsive Design: Ensure the platform is responsive and user-friendly on various devices.

Project Milestones:
Milestone 1: Project proposal with architecture and wireframes.
Milestone 2: Authentication and user management.
Milestone 3: Product catalog and shopping cart.
Milestone 4: Order management and payment integration.
Milestone 5: Performance optimization and testing.
Milestone 6: Final deployment with security measures.
Milestone 7: Presentation and demonstration of the e-commerce platform.

Teaching and learning method(s):	Face to face 32 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,2 and 3** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 4, 5, 6, 7 and 8.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain fundamental Angular concepts, including component architecture, data binding, services, and routing. They should also demonstrate proficiency in building Angular components and templates. By the end of the final exam, students should be able to design and develop complex Angular applications, implement advanced state management, optimize performance, conduct comprehensive testing, ensure security measures, handle advanced routing scenarios, create interactive forms and animations, and seamlessly integrate Angular with backend services, thus showcasing their mastery of advanced FrontEnd Web Development with Angular.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Angular Mastery: Deep understanding of Angular, from AngularJS to the latest framework. 2. Component-Based Development: Creating complex web apps with Angular components. 3. Effective State Management: Implementing advanced state management with Angular services. 4. Performance Optimization: Optimizing Angular apps for speed and user experience. 5. Robust Testing and Debugging: Comprehensive testing and debugging techniques. 6. Security Awareness: Implement best practices for securing Angular applications. 7. Backend Integration: Integrate Angular with RESTful APIs and backend services for full-stack development.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks

References:	[1] "Angular Development with TypeScript" by Yakov Fain and Anton Moiseev, July. 2017. [2] "Angular in Action" by Jeremy Wilken, March. 2018. [3] "*Pro Angular 16*" by A. Freeman, June. 2022. [4] Angular documentation and official guides (https://angular.io/docs).

Department of Computer Science

C-4.4.1-Scientific Research

Last Update Date: June 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Transversal Unit 4</td><td>Course</td><td colspan="2">Scientific Research</td><td>Code</td><td>C-4.4.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>31h50</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="4">Semester</td><td rowspan="4">4</td></tr>
<tr><td>Workshop</td><td>-</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>-</td></tr>
<tr><td>Self Study</td><td>25h00</td></tr>
<tr><td colspan="2" rowspan="2">Total workload</td><td colspan="3" rowspan="2">55h50 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Prof Mounir Zrigui</td><td>Involved Teacher(s)</td><td colspan="2">Prof Mounir Zrigui</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1</td></tr>
<tr><td>PLO</td><td colspan="2">P3, P8 and P9</td></tr>
<tr><td>Prerequisites:</td><td colspan="7">Students may need to have completed introductory courses in computer science and electrical engineering before taking a research methodologies course.</td></tr>
<tr><td>Corequisites:</td><td colspan="7">None: As this course typically serves as a foundational or introductory course.</td></tr>
<tr><td>Learning outcomes</td><td colspan="7">Upon completion of this teaching course, the student will be able to:

1. Understand the fundamental concepts and principles of scientific research methodologies.
2. Formulate research questions and hypotheses.
3. Design research experiments and investigations.
4. Gather and analyze data using appropriate tools and techniques.
5. Present research findings effectively.
6. Demonstrate ethical conduct in research.</td></tr>
<tr><td>Content:</td><td colspan="7">Chapter 1: Introduction to Scientific Research
1. Definition of research
2. Types of research
3. Importance of research in engineering</td></tr>
</table>

4. Ethics in Research (Ethical considerations in scientific research and, Informed consent and confidentiality)
5. General Process Model
6. Researcher Skills
7. Science and Scientific Knowledge (Applied Science)

Chapter 2: Research Design and Methodology

1. Research paradigms
2. Research Paradigms (Theoretical and empirical studies , Qualitative / quantitative research and Interpretive approach).
3. Design Science
4. Model, method and artifact
5. Theoretical / conceptual framework
6. Information Systems and Cybersecurity

Chapter 3: Investigation process

1. Identifying research gaps
2. Developing research questions and hypotheses (Central question, Hypotheses, Design of a plan)
3. Research Methods and Techniques (Qualitative and quantitative methods)
4. Data Collection Techniques (Surveys, questionnaires, and interviews, Observations and experiments and Data sources and data management)
5. Data Analysis Techniques (Statistical analysis, Qualitative data analysis and Data visualization)

Chapter 4: Writing Research Report /Paper

1. Conducting a comprehensive literature review
2. Identifying relevant sources and citing them properly
3. Writing an Academic Report / Ppaer : Adequacy and clarity
4. Research Ethics
5. Objectionable Scientific Conduct
6. Conflicts of Interest
7. Common Errors

Practical Work: Student Research Projects

1. Students choose research topics and work on their projects with guidance from the instructor
2. Students present their research findings to the class
3. Writing research reports and papers
4. Creating effective presentations
5. Crafting a research proposal
6. Peer review and feedback
7. Review of key concepts and techniques
8. Course reflection and feedback

Self-study:

Students will:

- Identify the specific field of science they want to research (e.g., biology, physics, psychology).
- Define their research objectives .

	- Review the basics of their chosen field by reading textbooks, taking online courses, or attending lectures. - Learn about different research methodologies, experimental designs, and data analysis techniques relevant to their field. - Think critically about the material they are studying. Question assumptions, evaluate evidence, and identify gaps in existing research.
Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners. Classes include theoretical and practical sessions with targeted discussions. Theoretical sessions will mainly use the expository method, using slide projection. The discussions will be oriented to the analysis of academic articles and research projects, where the active method will be used through their analysis and discussion. Most of the activities to be developed by the students will be supervised by the teachers.
Teaching technique(s):	Classroom lectures with tutorials and practical work , activities , T. B. L And P.B.L.
Assessment method(s):	A midterm written test (D.S) is planned to ensure the evaluation of the **learning outcomes** achievement level. A practical project / portfolio presentation to evaluate: - Class Participation and Engagement - Individual and Group Presentations - Written Assignments and Reports
Assessment criteria(s):	The application in practice of the contents taught such as the main concepts used in scientific research and the validation of the competences defined in the course as well as : - Research Proposal - Literature Review - Oral Presentations - Written Reports - Critical Analysis of Research Papers - Problem-Solving Assignments.
Assessment measure(s):	Assigning of a final average: - Class Participation - Assignments - Class participation and engagement - Assignments (literature review, research proposal) - Final research project (including proposal, research paper, and presentation).
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final examination: 60%.
Target skill(s):	1. Identify research problems and formulate research questions. 2. Design and conduct experiments or investigations. 3. Analyze and interpret research data.

	4. Prepare and present a research project proposal. 5. Define and execute a literature review strategy. 6. Write and present research findings effectively. 7. Apply ethical standards in research. 8. Systematically analyze scientific articles. 9. Explain some of the main methods and techniques of scientific research. 10. Describe the process of writing and scientific publication. 11. Discuss ethical issues that may occur in scientific research.
Teaching Manual:	• PowerPoint presentation • Lecturing • Tutorials • Courses project Lecturing
References:	[1] Berndtsson, Hansson, Olsson and Lundell (2008) Thesis Projects - A Guide for Students in Computer Science and Information Systems, Second Edition., Springer-Verlag. [2] Leedy, P, Ormrod, J. (2015), Practical Research: Planning and Design, 11th Edition, Pearson Education Limited. [3] Remenyi and Bannister (2012). Writing up your Research. Academic Publishing Limited. [4] Thiel, David (2014). Research Methods for Engineers. Cambridge University Press. [5] Wallwork, Adrian (2011). English for Writing Research Papers, Springer-Verlag. [6] Several Design Science academic papers to be deliver by teacher.

Department of Computer Science

C-4.4.2 Quality Management System (QMS)

Last Update Date: February 17, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Transversal Unit 4</td><td>Course</td><td colspan="2">Quality Management System (QMS)</td><td>Code</td><td>C-4.4.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="4">Semester</td><td rowspan="4">4</td></tr>
<tr><td>Workshop</td><td>-</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>-</td></tr>
<tr><td>Self Study</td><td>35h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mme Mouna Hjiri</td><td>Involved Teacher(s)</td><td colspan="2">Mme Mouna Hjiri</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 2 & obj 3</td></tr>
<tr><td>PLO</td><td colspan="2">P1 : Systems Specification and Design Methodologies
P2: Implementation Techniques
P5: Network and Information Systems Management
P3: Innovation, Research and Emerging Technologies</td></tr>
<tr><td><u>Prerequisites:</u></td><td colspan="7">Fundamental Knowledge of Computer Science and Electrical Engineering: Students should have a foundational understanding of computer science and electrical engineering principles, as they will be applying quality management concepts in these domains.
Basic Knowledge of Software Development: A basic understanding of software development processes is essential, as the course will cover quality management in software engineering.
Basic Knowledge of Hardware Development: Similarly, a basic understanding of hardware development processes is important, as quality management principles will also apply to electrical engineering and hardware design.
Statistics and Mathematics: Proficiency in basic statistics and mathematics is beneficial since the course will involve data analysis, quality metrics, and statistical process control.</td></tr>
<tr><td><u>Corequisites:</u></td><td colspan="7">Project Management: A course in project management can complement the understanding of quality management, as project management principles often intersect with quality management in engineering projects.</td></tr>
</table>

	Engineering Ethics: An ethics course can be taken concurrently to help students understand the ethical considerations of quality management in engineering. **Software Engineering and Hardware Engineering Courses:** While not strict corequisites, taking courses in software engineering and hardware engineering can provide practical context for quality management concepts in these areas.
Learning outcomes	Upon completion of this teaching course, the student will be able to acquire: 1. **Understand Quality Management:** Explain the fundamental concepts and principles of Quality Management Systems (QMS) and their relevance in the fields of computer science and electrical engineering. 2. **Apply Quality Assurance and Control Techniques:** Apply quality assurance and quality control techniques to the development and maintenance of software and hardware systems to ensure they meet established quality standards. 3. **Implement Quality Planning and Risk Management:** Develop quality plans and strategies, and effectively manage risks associated with computer and electrical system development. 4. **Measure and Analyze Quality:** Use appropriate quality metrics and measurement tools to evaluate and analyze the quality of computer and electrical systems. 5. **Create Quality Documentation:** Create and implement quality documentation and procedures in accordance with industry standards and best practices. 6. **Collaborate within Interdisciplinary Teams:** Engage in effective communication and collaboration within interdisciplinary teams to ensure the quality of computer and electrical systems and their components. 7. **Understand Ethical and Legal Aspects:** Understand the ethical and legal considerations of quality management in computer science and electrical engineering, including intellectual property and compliance with industry regulations.
Content:	**Chapter 1: Introduction to Quality Management Systems** 1. Introduction to Quality Management 2. Quality concepts and principles 3. Historical development of QMS 4. Relevance of QMS in computer science and electrical engineering 5. Ethical considerations in quality management **Chapter 2: Quality Models and Standards** 1. ISO 9001 and other relevant standards 2. CMMI (Capability Maturity Model Integration) 3. Six Sigma 4. TQM (Total Quality Management) 5. IEEE Software Engineering Standards 6. Assessment of various quality models **Chapter 3: Quality Planning and Control** 1. Quality planning and strategy development 2. Risk management and mitigation 3. Quality metrics and measurements **Chapter 4: Quality Assurance and Documentation** 1. Quality assurance processes and practices 2. Quality audits and reviews

	3. Quality documentation and standards compliance 4. Change management **Chapter 5: Process Improvement and Best Practices** 1. Process improvement methodologies (e.g., Six Sigma DMAIC) 2. Root cause analysis 3. Lean principles in quality improvement 4. Benchmarking and best practices 5. Case studies and real-world applications **Practical Work:** Group Project and Presentations 1. Collaborative project to apply QMS principles in real-world scenarios 2. Group presentations and discussions 3. Peer evaluation and feedback 4. Peer and instructor evaluations. 5. Course reflection and feedback. **Self-study:** Students will: - Conduct thorough research on the topic of their projects: Using reliable sources such as academic journals, textbooks, and reputable websites to gather information and deepen your understanding of the subject matter. - Consider the challenges you overcame and the skills you developed. - Ask for feedback from professors, mentors, or experienced engineers to provide insights and suggestions for improvement.
Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) is planned to ensure the evaluation of the **learning outcomes** achievement level. A practical project / portfolio presentation to evaluate: - Class Participation and Engagement - Individual and Group Presentations - Written Assignments and Reports

Assessment criteria(s) :	1. Understanding of Quality Management 2. Application of Quality Assurance and Control Techniques 3. Evaluation of Quality Models and Standards 4. Quality Planning and Risk Management 5. Measurement and Analysis of Quality 6. Creation of Quality Documentation (10%): 7. Collaboration within Interdisciplinary Teams (10%): 8. Understanding of Ethical and Legal Aspects (10%): 9. Project or Case Study
Assessment measure(s):	Assigning of a final average: Assignments, Quizzes, Exams, Assignments, Quizzes, Exams.
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam :60%
Target skill(s):	- Understand the fundamentals of Quality Management Systems and their relevance in computer science and electrical engineering. - Apply quality assurance and quality control techniques to software and hardware development processes. - Analyze and evaluate various quality models and standards used in the industry. - Demonstrate proficiency in quality planning, process improvement, and risk management. - Develop skills in measuring and analyzing the quality of computer and electrical systems. - Create and implement quality documentation and procedures. - Engage in effective communication and collaboration within interdisciplinary teams to ensure product quality. - Understand the ethical and legal aspects of quality management in computer science and electrical engineering.
Teaching Manual:	● PowerPoint presentation ● Lecturing ● Tutorials ● Courses project Lecturing
References:	[1] "Quality Management for Organizational Excellence" by David L. Goetsch and Stanley Davis. [2] "The Certified Six Sigma Green Belt Handbook" by Roderick A. Munro. [3] ISO 9001 : 2015 “International Standardization Organization “ : Quality management system requirements.

Department of Computer Science
C-4.5.1 End of Year Project

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Project unit 2</td><td>Course</td><td colspan="2">End of Year Project</td><td>Code</td><td>C-4.5.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>-</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td rowspan="2">--</td><td rowspan="4">Semester</td><td rowspan="4">4</td></tr>
<tr><td>Workshop</td><td>42h00</td></tr>
<tr><td>Project</td><td>-</td><td rowspan="2">Mixed</td></tr>
<tr><td>Self Study</td><td>42h00</td><td>XX</td></tr>
<tr><td colspan="3">Total workload</td><td colspan="2">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="3"></td><td colspan="2"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="3">Responsible Teacher</td><td colspan="2">Mr Mohamed Amine Omrani</td><td>Involved Teacher(s)</td><td colspan="2">--</td></tr>
<tr><td colspan="3" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1, Obj 2 and Obj3</td></tr>
<tr><td>PLO</td><td colspan="2">P1, P2, P3, P4, P5, P6, P7, P8, P9 and P10</td></tr>
<tr><td>Prerequisites:</td><td colspan="7">Completion of first-year courses in computer science / electrical engineering.</td></tr>
<tr><td>Corequisites:</td><td colspan="7">Mathematics and Physics Courses: Calculus and Physics II (if not already completed in the first year). These courses provide essential mathematical and physics foundations that are often required in electrical engineering projects.
Programming Courses: Data Structures and Algorithms. Data Structures and Algorithms are crucial for computer science projects. Students should have a strong programming background.
Electrical Engineering Courses: Electronics I and Digital Circuits. These courses cover essential electrical engineering principles that are often needed for electrical engineering projects.
Research Methodology or Technical Writing: Research Methods in Engineering or Technical Writing. These courses can help students develop research skills and effective technical communication, which are essential for the project.
Teamwork and Collaboration Skills: Preparing students for effective collaboration within project teams.</td></tr>
<tr><td>Learning outcomes</td><td colspan="7">Upon completion of this teaching course, the student will be able to:
1. Students will be able to create a detailed project plan that includes timelines, milestones, and resource allocation.
2. Students will demonstrate the ability to manage and adapt their project plan as necessary throughout the course.
3. Students will develop the skills to identify and define complex problems or challenges.</td></tr>
</table>

	4. Students will conduct in-depth research relevant to their project's subject matter. 5. Students will demonstrate the ability to collaborate and communicate effectively within a project team. 6. Students will acquire and apply technical skills or knowledge relevant to their specific project, which may include software tools, equipment operation, or other technical competencies. 7. Students will understand and adhere to ethical standards and professional responsibilities relevant to their field of study and the project. 8. Students will reflect on the project experience and evaluate their own performance and that of their team, identifying areas for improvement.
Content:	**Chapter 1:** Project Proposal and Planning - Introduction to the project. - Problem identification and project scope. - Literature review and research. - Project planning, including timelines and milestones. - Proposal presentation and approval. **Chapter 2: Design and Analysis** - System architecture and design. - Data collection and analysis. - Selection of hardware and software tools. - Risk assessment and mitigation strategies. **Chapter 3: Implementation** - Coding and software development (if applicable). - Hardware construction and testing (if applicable). - Regular progress reviews with advisors. **Chapter 4: Testing and Validation** - Unit testing. - System integration testing. - User testing (if applicable). - Quality assurance and debugging. - Documentation and Reporting: Technical documentation & progress reports and status updates. **Practical Work: End of Year Project** - Preparing a final project report - Creating a project presentation - Demonstrating the project to peers and instructors - Reflecting on lessons learned and future improvements - Students showcase their personal projects to the class - Peer evaluation and feedback - Assessment by the instructor.

	Self-study: Students will: - Conduct thorough research on the topic of their projects: Using reliable sources such as academic journals, textbooks, and reputable websites to gather information and deepen your understanding of the subject matter. - Consider the challenges you overcame and the skills you developed. - Ask for feedback from professors, mentors, or experienced engineers to provide insights and suggestions for improvement.
Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work , activities .
Assessment method(s):	A midterm written test (D.S) is planned to ensure the evaluation of the **learning outcomes** achievement level. A practical project / portfolio presentation to evaluate: - Class Participation and Engagement - Individual and Group Presentations - Project proposal and planning documentation. - Progress reports. - Final project report. - Project presentation and defense. - Peer and self-assessment. - Ethical reflection.
Assessment criteria(s) :	**Knowledge and understanding:** Students should be able to introduce themselves both formally and informally to others, understand different types of written texts and summarize them and understand graphics, visuals, tables, and diagrams and analyze them orally or transfer them into a written text. **Skills :** - Speaking fluently and confidently. - Acquiring new vocabularies related to the main topics. - Writing accurately and coherently.
Assessment measure(s):	Assigning of a final average: - Exhibition or demonstration of projects to the community or industry partners. - Peer evaluation and feedback. - Q&A session. - Evaluation of the project based on technical quality, project management, and communication. - Assessment of ethical considerations and social implications.

Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	- Apply computer science and electrical engineering concepts and principles to practical problems. - Demonstrate proficiency in project management, including planning, organization, and time management. - Develop innovative and creative solutions to engineering challenges. - Effectively communicate and present technical information in written and oral forms. - Collaborate with team members and adapt to different roles within a project team. - Evaluate the ethical and societal implications of engineering projects. - Apply research skills to investigate and address complex problems.
Teaching Manual:	● PowerPoint presentation ● Lecturing ● Tutorials ● Courses project Lecturing
References:	[1] "Research Methodology: A Step-by-Step Guide for Beginners" by Ranjit Kumar. [2] "Ethics for the Information Age" by Michael J. Quinn. [3]"Engineering Ethics: Concepts and Cases" by Charles E. Harris, Michael S. Pritchard, and Michael J. Rabins.

Department of Computer Science

C-4.6.1-Elective Course 1 - Advanced JAVA Edition

Last Update Date: February 17, 2024

Syllabus Sheet

Unit	**Elective Unit 1**		**Course**	**Elective Course 1 - Advanced JAVA Edition (EE / ME)**		**Code**	**C-4.6.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**4**
	Workshop	**10h30**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**25h00**					
Total workload	**56h00 / Semester**				**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher	**Mr Mohamed Amine Omrani**		**Involved Teacher(s)**	**Mr Mohamed Amine Omrani**			
Review committee	**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1**			
			PLO	**P1 : Systems Specification and Design Methodologies P2: Implementation Techniques P3: Innovation, Research and Emerging Technologies**			

Prerequisites:	Fundamental concepts of Object-Oriented Programming (OOP), JAVA, JAVA SE, HTML, CSS, and JS.
Corequisites:	The JAVA SE Runtime Environment
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Understand the evolution, scope, and significance of Java EE and Java ME in modern software development. 2. Analyze and apply fundamental concepts and principles of Java EE architecture, including servlets, JSP, JPA, and Hibernate. 3. Develop advanced web applications using Java EE technologies such as JSF, JAX-RS, and security mechanisms. 4. Integrate Java EE applications with microservices architecture for scalability and modularity. 5. Implement asynchronous communication and advanced data persistence strategies in Java EE applications. 6. Develop mobile applications using Java ME platform, integrating advanced features like Bluetooth and location-based services.

	7. Optimize performance and manage resources effectively in Java ME applications. 8. Implement security measures to protect Java EE and Java ME applications from vulnerabilities and ensure application integrity.
Content:	**Part 1: Java Enterprise Edition (Java EE)** **Chapter 1 : Introduction to Java EE** 1. Evolution and Overview of Java EE 2. Understanding the Scope and Applications of Java EE 3. Importance and Benefits of Java EE in Modern Software Development **Chapter 2 : Java EE Fundamentals** 1. Exploring the Fundamentals of Java EE Architecture 2. Client/Server Architectures and Communications (TCP/IP and UDP) 3. Advanced Concepts in Servlets and JSP 4. Java Standard Tag Library (JSTL) 5. Deep Dive into Java Persistence API (JPA) and Hibernate 6. Object-Relational Mapping (Hibernate and HQL) **Chapter 3 : Advanced Web Development with Java EE** 1. Advanced JavaServer Faces (JSF) Development 2. Developing RESTful Web Services with JAX-RS 3. Annotations and Mails 4. Utilizing Java EE Security Features **Chapter 4 : Advanced Java EE Techniques** 1. Advanced Data Persistence Strategies in Java EE 2. Integrating Java EE Applications with Microservices Architecture 3. Implementing Asynchronous Communication in Java EE **Part 2: Java Micro Edition (Java ME)** **Chapter 5 : Introduction to Java ME** 1. Overview of Java ME and its Role in Mobile and Embedded Development 2. Understanding the Architecture of Java ME **Chapter 6 : Java ME Development Basics** 1. Developing Mobile Applications with Java ME 2. Integrating Advanced Features like Bluetooth and Location-Based Services into Java ME Applications **Chapter 7 : Advanced Java ME Techniques** 1. Optimizing Performance and Resource Management in Java ME Applications 2. Implementing Security Measures in Java ME
	Self-study & Projects: **Project 1: Online Bookstore Website:** **Description**: In this project, students will develop an online bookstore website using JAVA EE technologies. The website should include features like user registration and authentication, browsing and searching for

	books, adding books to a shopping cart, and completing orders. Students will use Servlets for handling user requests and JavaServer Pages (JSP) for rendering the web pages. They should also apply Java Standard Tag Library (JSTL) to simplify the presentation logic and Java Persistence API (JPA) for managing book data and user profiles. This project will cover various aspects of JAVA EE, including servlets, JSP, JSTL, and JPA. **Skills Practiced**: ● Servlets ● JavaServer Pages (JSP) ● Java Standard Tag Library (JSTL) ● Java Persistence API (JPA) ● User authentication and authorization ● Database management **Project 2: RESTful Web Service for a Task Manager:** **Description**: In this project, students will create a RESTful web service for a task manager application using JAVA EE. The web service should allow users to perform CRUD (Create, Read, Update, Delete) operations on tasks. Students will implement the web service endpoints using JAX-RS (Java API for RESTful Web Services). They will also use JSON for data exchange and leverage annotations to define resource paths and HTTP methods. Additionally, students can implement a client-side application (e.g., a simple web page or mobile app) that consumes the RESTful API. This project will provide hands-on experience with JAX-RS and RESTful web services in the context of JAVA EE. **Skills Practiced:** ● JAX-RS (Java API for RESTful Web Services) ● JSON for data exchange ● RESTful API design and implementation ● Client-server interaction ● CRUD operations on resources **Project 3: RESTful Web Service for a Task Manager:** **Description**: In this project, students will Develop a location-based reminder application using Java ME that allows users to set reminders based on their current location. The application should enable users to define specific locations (e.g., home, office, grocery store) and set reminders associated with those locations. When the user enters or exits a predefined location, the application should trigger the corresponding reminder notification. **Skills Practiced:** ● Develop a robust Java ME application that leverages location-based services for practical use. ● Implement efficient location tracking and reminder notification mechanisms. ● Provide a user-friendly interface for managing reminders and locations effectively. ● This project will allow students to apply their knowledge of Java ME development and location-based services to create a practical and useful application for mobile devices.
Teaching and learning method(s):	Face to face 32 hours apx. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2, 3 and 4.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 5, 6, 7 and 8.**

Assessment criteria(s) :	During the midterm assessment, students understand the JAVA should demonstrate the ability to develop JAVA EE applications utilizing libraries such as JSTL and APIs like JPA and advanced APIs and tools like JAX-RS, JSON for data exchange, RESTful API design /implementation and Client-server interaction. By the end of the final examination, students should be proficient in both JAVA EE and JAVA ME.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Component Proficiency: Understand JAVA EE components. 2. Communication Skills: Master Client/Server communication and JAVA server usage. 3. Library and API Expertise: Utilize JSTL and JPA effectively. 4. Application Development: Independently create JAVA EE applications. 5. Component Proficiency: Understand Java ME components. 6. Communication Skills: Master Client/Server communication and Java ME usage. 7. Library and API Expertise: Utilize MIDP and CLDC APIs effectively. 8. Application Development: Independently create Java ME applications.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Oracle Corporation. "Java EE Documentation." Available at: https://www.oracle.com/java/technologies/javaee-documentation.html [3] Liang, Y. "Introduction to Java Programming, Comprehensive Version." Pearson, 2015. [6] Karthikeyan, M. A. "Java EE References and Designing." ISBN-13: 9798533631914. [7] Oracle Corporation. "Java EE 7 Documentation." Available at: https://docs.oracle.com/javaee/7/index.html [8] Lindholm, T., and Yellin, F. "The Java™ Virtual Machine Specification, Second Edition." Available at: http://java.sun.docs/books/jvms [9] Chen, L., & Wang, Y. (2022). "Optimizing Performance of Java ME Applications on Resource-Constrained Devices." IEEE Internet of Things Journal, 9(2), 456-469. DOI: 10.1109/JIOT.2022.987654321 [10] The Java Community Process (JCP) for Java ME: https://www.jcp.org/en/jsr/detail?id=360 [11] The Eclipse Foundation's Eclipse ME project: https://projects.eclipse.org/projects/tools.me [12] Official blogs and forums related to Java ME development from Oracle or other reputable sources.

Department of Computer Science

C-4.6.1-Elective Course 1 - ASP CORE

Last Update Date: February 17, 2024

Syllabus Sheet

Unit	**Elective Unit 1**		**Course**	**Elective Course 1 - ASP CORE**		**Code**	**C-4.6.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**4**
	Workshop	**10h30**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**25h00**					
Total workload	**56h00 / Semester**				**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher	**Mr Walid Chafroud**			**Involved Teacher(s)**	**Mr Walid Chafroud**		
Review committee	**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**			**Objective**	**Obj 1**		
				PLO	**P1 : Systems Specification and Design Methodologies P2: Implementation Techniques P3: Innovation, Research and Emerging Technologies**		

Prerequisites:	Basic knowledge of web programming languages such as HTML, CSS, and JavaScript. Familiarity with C# and the .NET Framework.
Corequisites:	Database design and SQL. Fundamentals of object-oriented programming with C#.
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Understand and apply the fundamental principles of ASP.NET Core. 2. Design and deploy web applications with ASP.NET Core. 3. Integrate databases with Entity Framework Core. 4. Master security and authentication with ASP.NET Core Identity. 5. Develop RESTful APIs with ASP.NET Core. 6. Implement unit and integration tests for ASP.NET Core applications. 7. Utilize middleware and services in ASP.NET Core. 8. Create real-time applications with SignalR. 9. Understand the deployment and performance of ASP.NET Core. 10. Delve into advanced practices of ASP.NET Core.

Content:

Chapter 1: Basic Concepts

- 1.1 Introduction to ASP.NET Core
  - What is ASP.NET Core?
  - History and Evolution
  - ASP.NET Core vs. ASP.NET
- 1.2 Configuration and services
  - App settings and configuration files
  - Dependency Injection and Service Lifetimes
- 1.3 Creating Your First ASP.NET Core Application
  - Setting up the development environment
  - Building and running the application

Chapter 2: MVC and Razor Pages

- 2.1 Introduction to MVC
  - Understanding the MVC pattern
  - Role of Model, View, and Controller
- 2.2 Controllers and actions
  - Defining controllers
  - Action methods and routing
- 2.3 Razor views and models
  - Introduction to Razor syntax
  - ViewModels and ViewData

Chapter 3: Databases and Entity Framework Core

- 3.1 Database Design
  - ER diagrams and database schema
  - Migrations
- 3.2 CRUD Operations with Entity Framework Core
  - DbSet and DbContext
  - LINQ queries

Chapter 4: Security and Authentication

- 4.1 ASP.NET Core Identity
  - Setting up Identity
  - User management and roles
- 4.2 Authentication and Authorization
  - Different authentication methods
  - Policy-based authorization

Chapter 5: API and Web Services

- 5.1 Creating a RESTful API
  - Principles of REST
  - Building RESTful services using ASP.NET Core
- 5.2 Consuming an API
  - HTTP Client
  - Third-party API integration

Chapter 6: Testing and Quality

- 6.1 Unit Tests
  - xUnit and Test project setup
  - Mocking with Moq
- 6.2 Integration Tests
  - Testing API endpoints

- Database testing strategies

Chapter 7: Middleware and Services

- 7.1 Using Middleware
  - Built-in middleware components
  - Creating custom middleware
- 7.2 Dependency Injection
  - DI in ASP.NET Core
  - Scopes and service registration

Chapter 8: Real-time Applications with SignalR

- 8.1 Introduction to SignalR
  - WebSockets and SignalR
  - Push notifications
- 8.2 Building Real-time Features
  - Chat applications
  - Real-time data updates

Chapter 9: Deployment and Performance

- 9.1 Deployment Strategies
  - Deployment to IIS
  - Cloud deployment with Azure
- 9.2 Performance Optimization
  - Caching strategies
  - Asynchronous operations

Chapter 10: Advanced Practices

- 10.1 Advanced Routing
  - Attribute routing
  - Route constraints
- 10.2 Extending ASP.NET Core
  - Customizing Identity
  - Building and using middleware

Project Topic 1:

Title: Design and Deployment of an E-commerce Platform with ASP.NET Core: From Design to Production

Description:This project consists of creating a functional e-commerce platform using ASP.NET Core integrating best practices in web design, security, and database management.

Project Topic 2:

Title: Building a Blogging Platform with ASP.NET Core and Blazor

Description:This project consists of developing a blogging platform using ASP.NET Core for the backend and Blazor for the frontend. The platform will enable users to create accounts, write, edit, and delete their articles, and allow others to comment on them. Through this project, students will integrate many of the concepts learned during the ASP CORE course, providing hands-on experience and a deeper understanding of the content.

Teaching and learning method(s):	Face to face 32 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2, 3 and 4.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 5, 6, 7, 8, 9, 10 and 11.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain the fundamental principles o ASP.NET Core, the design of web applications, and the integration of databases with Entity Framewor Core. By the end of the final exam, students should be able to master security, develop RESTful APIs, and perform tests for ASP.NET Core applications
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Developing robust web applications. 2. Integrating databases. 3. Managing web security. 4. Designing APIs. 5. Testing and deployment.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] "Pro ASP.NET Core" by Adam Freeman. [2] Official ASP.NET Core Documentation. [3] "Entity Framework Core in Action" by Jon Smith. [4] ASP.NET Core forums: Official forums where students can interact with other learners and experts

Department of Computer Science

C-4.6.2-Elective Course 2 - Image Processing **Last Update Date: February 17, 2024**

Syllabus Sheet

Unit	**Elective Unit 1**		**Course**	**Elective Course 2 - Image Processing**		**Code**	**C-4.6.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**--**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**4**
	Workshop	**21h00**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**35h00**					
Total workload		**56h00 / Semester**			**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher		**Mr Mehrez Abdellaoui**		**Involved Teacher(s)**	**Mr Mehrez Abdellaoui**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1 & Obj3**		
				PLO	**P2: Implementation Techniques P3: Innovation, Research and Emerging Technologies P9: Lifelong Learning and Adaptability**		

Prerequisites:	- Signal Processing, Mathematics for Engineer 1, Mathematics for Engineer 2, Probability and Statistics, Advanced Object Oriented Programming (Python)
Corequisites:	- Industrial Vision Systems, Artificial Intelligence
Learning outcomes	At the end of this teaching module, the student will be able to: 1. Understand Image Processing Fundamentals 2. Study Digital Image Basics 3. Study Image Enhancement Techniques 4. Study Image Restoration Techniques 5. Understand the Image Compression 6. Study the Image Segmentation and Object Detection techniques 7. Study the Feature Extraction algorithm for Image Analysis
Content :	**Chapter 1: Introduction to Image Processing** 1. What is Image Processing? 2. Image Processing Applications 3. Digital Image Basics 4. Image Enhancement vs. Image Restoration

5. Image Processing Systems

Chapter 2: Image Enhancement

1. Point Processing Techniques
2. Histogram Processing
3. Spatial Domain Filtering
4. Frequency Domain Filtering
5. Enhancement in Color Images

Chapter 3: Image Restoration

1. Degradation Model
2. Noise Models
3. Image Deconvolution
4. Inverse Filtering
5. Wiener Filtering
6. Constrained Least Squares Filtering

Chapter 4: Image Compression

1. Need for Image Compression
2. Lossless vs. Lossy Compression
3. Transform Coding (e.g., Discrete Cosine Transform)
4. Quantization
5. JPEG and JPEG 2000 Compression Standards

Chapter 5: Image Segmentation

1. Image Segmentation Concepts
2. Thresholding
3. Region-Based Segmentation
4. Edge Detection
5. Watershed Transform

Chapter 6: Feature Extraction

1. Shape Analysis
2. Texture Analysis
3. Color-Based Features
4. Wavelet Transform for Feature Extraction

Workshops:

W 1 : Enhancing image contrast

W 2 : Deblurring images with known degradation

W3 : Image compression using Discrete Cosine Transform (DCT)

W 4 : Image segmentation using thresholding

W5 : Wavelet-based feature extraction

Projects :

Different projects are offered to the students. The student has the choice to select one of the proposed topics or to choose a topic of their own proposal.

Examples of projects:

	P1: Image Enhancement Tool: Create a user-friendly image enhancement tool that allows users to apply various enhancement techniques P2: Image Deblurring: Build a tool to deblur images that have been degraded due to factors like motion or defocus. P3: Image Compression and Quality Assessment:Create an image compression tool that allows users to compress images while maintaining acceptable quality P4: Image Segmentation and Object Counting: Develop an application for image segmentation and object counting P5 :Feature Extraction and Object Recognition:Build a tool for extracting and recognizing features from images P6 :Real-Time Webcam Filters: Create a real-time image processing application that applies filters to video captured from a webcam P7 :Medical Image Analysis:Develop a tool for medical image analysis, focusing on a specific application like tumor detection **Self Study :** Different self study themes integrate into this subject such as: -Learn about the components and architecture of image processing systems, including hardware (sensors, cameras) and software (algorithms, libraries). -Explore point processing techniques, such as contrast stretching, histogram equalization, and gamma correction. -Explore various noise models (e.g., Gaussian, salt and pepper) that can affect image quality. -Explore constrained least squares filtering as a method for image restoration under specific constraints. -Understand the distinctions between lossless and lossy image compression and when to use each. -Study edge detection methods to identify boundaries and edges within an image. -Understand the watershed transform as a technique for image segmentation and object separation. -Explore techniques for analyzing and extracting shape-based features from images. -Learn about color-based feature extraction and its applications in image analysis.
Teaching and learning method(s):	Face to face 21 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching Technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s) :	1. A midterm written test (DS) and a final written test (Exam) are planned. 2. The mid-term written test will be devoted to evaluating the level of achievement of learning outcomes 1, 2, 3 and 4. 3. The final written test is devoted to evaluating the level of achievement of learning outcomes 5, 6 and7.

	4. A practical exam to assess the student's ability to use a python algorithm for image processing. 5. A note dedicated to the evaluation of the image processing mini-project for each student. 6. A grade will be distributed for self-study work. This work will be controlled by the teacher through the GIT HUB.
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain advanced design theories, utilize sophisticated design tools, storyboard complex interactions, and create responsive designs (Figma/ Adobe XD). By the end of the final exam, students should be able to integrate feedback into their designs, understand the psychological aspects influencing user behavior, craft inclusive designs, and anticipate future trends in UI/UX.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	**1.** Advanced Prototyping **2.** Design System Development **3.** Cross-platform Design Adaptability **4.** Advanced User Testing and Analysis **5.** Inclusive and Accessible Design Crafting **6.** Ethical Design Considerations **7.** Practical Skills (FIGMA and ADOBE XD)
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] "Don't Make Me Think" by Steve Krug: This is a classic in the field of usability and gives a great introduction to the principles of intuitive navigation and information design. [2] "The Design of Everyday Things" by Don Norman: While not exclusively about UI/UX design, this book teaches critical design thinking that can be applied in the digital space. [3] "Lean UX" by Jeff Gothelf: Focuses on the iterative design process. Official Figma Resources: [4] Figma's Help Center: Comprehensive guide covering almost all features. [5] Figma Community: A place to find user-generated content, including design files and plugins. Official Adobe XD Resources: [6] Adobe XD Tutorials: Available on Adobe's official website, covering a range of topics.

	[7] Adobe XD User Guide: Comprehensive documentation on all features.

Department of Computer Science

C-4.6.2-Elective Course 2 - UX/UI Design

Last Update Date: February 17, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Elective Unit 1</td><td>Course</td><td colspan="2">Elective Course 2 - UX/UI Design</td><td>Code</td><td>C-4.6.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>--</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td rowspan="2">--</td><td rowspan="4">Semester</td><td rowspan="4">4</td></tr>
<tr><td>Workshop</td><td>21h00</td></tr>
<tr><td>Project</td><td>--</td><td rowspan="2">Mixed</td></tr>
<tr><td>Self Study</td><td>35h00</td><td>XX</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mme Safa Ben Salem</td><td>Involved Teacher(s)</td><td colspan="3">Mme Safa Ben Salem</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1 & Obj3</td></tr>
<tr><td>PLO</td><td colspan="3">P2: Implementation Techniques
P3: Innovation, Research and Emerging Technologies
P9: Lifelong Learning and Adaptability</td></tr>
<tr><td colspan="2"><u>Prerequisites:</u></td><td colspan="6">Basic understanding of design principles
Familiarity with web and mobile application development
Prior coursework in graphic design or a related field</td></tr>
<tr><td colspan="2"><u>Corequisites:</u></td><td colspan="6">Interaction Design
Human-Computer Interaction</td></tr>
<tr><td colspan="2"><u>Learning outcomes</u></td><td colspan="6">Upon completion of this teaching module, the student will be able to:

1. Analyze and critique modern UI/UX design trends and methodologies.
2. Develop advanced UI/UX strategies aligned with user-centered design principles.
3. Utilize sophisticated tools and software for UI/UX prototyping and wireframing.
4. Design adaptive and responsive user interfaces for a variety of platforms.
5. Conduct advanced user testing methodologies, including A/B testing and heatmaps.
6. Integrate feedback loops and iterative design into the UI/UX process.
7. Understand the psychological and sociological aspects influencing user behavior.
8. Develop design systems and UI kits for large-scale applications.
9. Analyze accessibility and inclusivity in design.
10. Craft compelling user narratives and storyboarding for complex digital interactions.</td></tr>
</table>

Content:	**Chapter 1: Advanced Design Theories and Methodologies** 1. Evolution of UI/UX Design 2. Modern Design Systems and Patterns 3. Cross-cultural Design Considerations **Chapter 2: Tools of the Trade** 1. Advanced Features in Tools like Figma, Adobe XD, and Sketch 2. Integrating with Development Environments 3. Collaboration and Version Control in Design **Chapter 3: Crafting Advanced User Journeys** 1. Storyboarding Complex Interactions 2. Designing for Diverse User Personas 3. Cognitive Walkthroughs **Chapter 4: Responsive and Adaptive Design** 1. Fluid Grids and Flexible Images 2. Designing for Wearables and IoT Devices 3. Breakpoints and Media Queries in Design **Chapter 5: User Testing and Feedback Loops** 1. A/B Testing and Multivariate Testing 2. Eye-tracking and Heatmaps 3. Incorporating Feedback into Iterative Design **Chapter 6: Design Systems and Scalability** 1. Creating and Maintaining a UI Kit 2. Design Tokens and Theming 3. Integrating with Development Frameworks **Chapter 7: Psychological Principles in UI/UX Design** 1. Cognitive Load and Information Processing 2. Color Theory and Emotional Design 3. Micro-interactions and Gamification **Chapter 8: Accessibility and Inclusivity** 1. Designing for All Users 2. ARIA, WCAG, and Other Standards 3. Multimodal Interfaces **Chapter 9: Future Trends in UI/UX** 1. Designing for AR and VR 2. Voice and Conversational Interfaces 3. Ethical Considerations in Design **Workshops :** 1. Introduction to Adobe XD and Figma: ● Overview of the tools and their ecosystems ● Setting up a project: Workspace, Artboards, and Design Grids 2. Advanced Prototyping with Adobe XD: ● Crafting intricate user journeys using Auto-Animate and Overlays

	• Voice prototyping and responsive resize • Using Component States and Interactive Components 3. Design Collaboration and Feedback in Figma: • Sharing, commenting, and collaborating in real-time • Using the Figma Mirror for live previews • Version history and rollback features 4. Responsive Design in Adobe XD and Figma: • Designing for different device sizes and resolutions • Using Layout Grids and Constraints in Figma • Adaptive artboards in Adobe XD 5. Design Systems and Components: • Building a comprehensive design system in Figma • Using and sharing libraries in Adobe XD • Nested components and overrides in Figma 6. Advanced User Testing with Adobe XD: • Crafting interactive prototypes for user testing • Gathering feedback directly within Adobe XD • Integrating with third-party user testing platforms 7. Integrating with Development using Figma and Adobe XD: • Exporting assets and generating CSS code • Using the Figma API for automation and extensions • Collaboration with developers using Adobe XD plugins
	Project and Self-study:: Project Title: Designing an Omni-Channel Experience Objective: Craft a comprehensive UI/UX design strategy for a multi-platform application (web, mobile, wearables). This should include user personas, wireframes, interactive prototypes, and a design system. Ensure the design is inclusive and accessible, with a focus on a seamless user experience across platforms.
Teaching and learning method(s):	Face to face 56 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. • The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2, 3 and 4** • The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 5, 6, 7, 8, 9 and 10.**

Assessment criteria(s) :	During the midterm exam, students should be able to define and explain advanced design theories, utilize sophisticated design tools, storyboard complex interactions, and create responsive designs (Figma/ Adobe XD). By the end of the final exam, students should be able to integrate feedback into their designs, understand the psychological aspects influencing user behavior, craft inclusive designs, and anticipate future trends in UI/UX.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Advanced Prototyping 2. Design System Development 3. Cross-platform Design Adaptability 4. Advanced User Testing and Analysis 5. Inclusive and Accessible Design Crafting 6. Ethical Design Considerations 7. Practical Skills (FIGMA and ADOBE XD)
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] "Don't Make Me Think" by Steve Krug: This is a classic in the field of usability and gives a great introduction to the principles of intuitive navigation and information design. [2] "The Design of Everyday Things" by Don Norman: While not exclusively about UI/UX design, this book teaches critical design thinking that can be applied in the digital space. [3] "Lean UX" by Jeff Gothelf: Focuses on the iterative design process. Official Figma Resources: [4] Figma's Help Center: Comprehensive guide covering almost all features. [5] Figma Community: A place to find user-generated content, including design files and plugins. Official Adobe XD Resources: [6] Adobe XD Tutorials: Available on Adobe's official website, covering a range of topics. [7] Adobe XD User Guide: Comprehensive documentation on all features.

Department of Computer Science
NES-C-4.7.1-Routing and Switching

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Networks and Embedded System 2</td><td>Course</td><td colspan="2">Routing and Switching</td><td>Code</td><td>NES-C-4.7.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Networks and Embedded System</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="4">Semester</td><td rowspan="4">4</td></tr>
<tr><td>Workshop</td><td>21h 00</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>42h00</td></tr>
<tr><td colspan="3">Total workload</td><td colspan="2">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="3">Responsible Teacher</td><td colspan="2">Mr Hatem Jarboua</td><td>Involved Teacher(s)</td><td colspan="2">Mr Hatem Jarboua</td></tr>
<tr><td colspan="3" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1 & Obj2</td></tr>
<tr><td>PLO</td><td colspan="2">P1: Systems Specification and Design Methodologies
P3: Innovation, Research and Emerging Technologies
P6: Project Management and Leadership
P8: Communication and Collaboration</td></tr>
<tr><td colspan="3"><u>Prerequisites:</u></td><td colspan="5">- Showcase comprehension of OSI model layers, TCP/IP protocols, subnetting, and basic network topologies.
- Be acquainted with Windows Server, Linux, or Cisco IOS, demonstrating configuration and management skills.
- Display knowledge of routing protocols (e.g., OSPF, EIGRP, BGP) and switching concepts (e.g., VLANs, STP).
- Gain hands-on experience configuring routers and switches, troubleshooting network issues, and implementing best practices.</td></tr>
<tr><td colspan="3"><u>Corequisites:</u></td><td colspan="5">- Learn advanced security protocols and techniques, including encryption, authentication, intrusion detection, and firewall configuration.
- Dive into network architecture principles, mastering scalability, redundancy, load balancing, and fault tolerance.
- Investigate telecommunications technologies, including voice over IP (VoIP), MPLS, DWDM, and SONET/SDH.
- Gain expertise in complex routing protocols such as MPLS, IPv6, and multicast routing, and their applications in enterprise networks.</td></tr>
</table>

Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Understanding Networking Fundamentals 2. Ethernet LAN Technology 3. Covers the fundamentals of routing, including different routing protocols (such as RIP and OSPF) 4. Have Basic network security concepts, access control mechanisms, and security best practices to protect network assets 5. Doing Hands-on experience in configuring and managing Huawei networking equipment, including routers and switches. 6. Troubleshooting Skills 7. Preparation for HCIA Exam
Content:	**Course:** **Chapter 1**: Fundamentals of Routing and Switching ● Overview of networking basics ● Introduction to routing protocols ● Switching concepts and VLANs **Chapter 2**: Advanced Routing Protocols ● In-depth study of OSPF, EIGRP, and BGP ● Route summarization and route redistribution ● Advanced routing scenarios and troubleshooting techniques **Chapter 3**: Network Security in Routing and Switching ● Security threats and vulnerabilities in networks ● Access control lists (ACLs) and security policies ● Virtual Private Networks (VPNs) and secure network communication **Chapter 4**: Network Design and Optimization ● Network design principles and methodologies ● Scalability, redundancy, and high availability ● Performance optimization and Quality of Service (QoS) **Tutorials:** **Tutorial 1**: Configuring Basic Routing and Switching ● Setting up a small network with routers and switches ● Configuring IP addresses, VLANs, and basic routing protocols **Tutorial 2**: Implementing Advanced Routing Protocols ● Configuring OSPF, EIGRP, and BGP in a network environment ● Troubleshooting routing issues and optimizing routing tables **Tutorial 3**: Enhancing Network Security ● Configuring access control lists (ACLs) and security policies ● Implementing VPNs for secure remote access **Tutorial 4**: Designing and Optimizing Networks ● Designing network architectures for scalability and redundancy

	● Implementing QoS to optimize network performance **Practical Work:** **Practical Work 1**: Hands-on Labs ● Lab exercises covering basic routing and switching configurations ● Troubleshooting common network issues **Practical Work 2**: Advanced Routing Labs ● Hands-on labs focusing on implementing OSPF, EIGRP, and BGP ● Simulation of complex routing scenarios and troubleshooting exercises **Practical Work 3**: Network Security Implementation ● Configuring security features such as ACLs and VPNs in lab environments ● Simulated attacks and vulnerability assessments **Practical Work 4**: Network Design Projects ● Designing network architectures to meet specific requirements ● Optimization projects aimed at improving network performance and reliability
Teaching and learning method(s):	Face to face (face-to-face) 42 hours. A course space is created on the Huawei cloud platform . This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	**A midterm written test (D.S), a practice test and a final written test (Exam) are planned**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain basic and intermediate knowledge of Huawei equipment configurations By the end of the final exam, students should be able pass the HCIA routing and switching Exam
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Learn to configure routers and switches, setting up IP addresses, VLANs, and routing protocols to establish network connectivity.

	2. Optimize routing protocols and configurations to improve network efficiency and minimize packet loss. 3. Implement security policies such as access control lists (ACLs) and VPNs to protect network infrastructure from unauthorized access and security threats. 4. Develop the ability to troubleshoot network issues, diagnosing routing and switching problems and implementing solutions to ensure smooth network operation.
Teaching Manual:	• Huawei Palteform • Huawei Simulator. • Powerpoint presentation • Tutorial booklet • Huawei Workbooks
References:	[1] Huawei e learning Platform [2] Tanenbaum, A. S., & Wetherall, D. J. (2011). Computer Networks (5th ed.). Pearson Education. [3] Cisco. (2019). Cisco CCNA Routing and Switching 200-125 Official Cert Guide Library. Cisco Press. [4] Kurose, J. F., & Ross, K. W. (2017). Computer Networking: A Top-Down Approach (7th ed.). Pearson. [5] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communications Surveys & Tutorials, 17(4), 2347-2376.

Department of Computer Science
NES-C-4.7.2-Antenna and Propagation

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Networks and Embedded System 2</td><td>Course</td><td colspan="2">Antenna and Propagation</td><td>Code</td><td>NES-C-4.7.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Networks and Embedded System</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">4</td></tr>
<tr><td>Workshop</td><td>21h 00</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">4</td></tr>
<tr><td>Self Study</td><td>42h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mme Elhem</td><td>Involved Teacher(s)</td><td colspan="3">Mme Elhem</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1 & Obj2</td></tr>
<tr><td>PLO</td><td colspan="3">P1: Systems Specification and Design Methodologies
P3: Innovation, Research and Emerging Technologies
P6:Project Management and Leadership
P8: Communication and Collaboration</td></tr>
<tr><td><u>Prerequisites:</u></td><td colspan="7">• Understand Maxwell's equations and wave propagation theory to comprehend antenna behavior effectively.
• Acquire knowledge of Fourier transforms and modulation techniques to grasp antenna performance in communication systems.</td></tr>
<tr><td><u>Corequisites:</u></td><td colspan="7">• Gain additional insights into antenna design and analysis by enrolling in courses covering electromagnetic field theory.
• Connect antenna theory with real-world applications by studying wireless communication systems in parallel.
.</td></tr>
</table>

Learning outcomes	• Develop the ability to analyze the radiation patterns, impedance matching, and gain of antennas using theoretical models and simulation tools. • Acquire the skills to design and optimize antenna systems for specific applications, considering factors such as frequency bands, polarization, and environmental constraints. • Gain insight into propagation mechanisms in various environments (e.g., free space, urban, indoor) and model path loss, shadowing, and multipath fading effects. • Examine emerging antenna technologies such as metamaterials, reconfigurable antennas, and MIMO systems, and explore their applications in next-generation wireless communication systems.
Content:	**Content:** **Chapter 1**: Explore Antenna Fundamentals • Introduce antenna parameters and radiation mechanisms • Discuss types of antennas and their applications in different scenarios **Chapter 2**: Study Antenna Arrays and Beamforming • Learn principles of antenna arrays and beamforming techniques • Consider design considerations for phased array antennas and their performance analysis **Chapter 3**: Model Propagation and Channel Characterization • Understand propagation mechanisms in different environments (free space, urban, indoor) • Model path loss, shadowing, and multipath fading effects **Chapter 4**: Investigate Advanced Antenna Technologies • Examine emerging antenna technologies such as metamaterials, reconfigurable antennas, and MIMO systems • Explore applications of advanced antennas in next-generation wireless communication systems **Tutorials**: **Tutorial 1**: Simulate Antenna Characteristics using Software Tools • Guide on simulating antenna characteristics using software packages like MATLAB, HFSS, or CST Microwave Studio. **Tutorial 2**: Design Patch Antennas • Step-by-step instructions on designing and fabricating patch antennas, including impedance matching techniques and practical considerations. **Tutorial 3**: Measure and Test Antenna Performance • Hands-on sessions covering antenna measurement techniques in an anechoic chamber or through field testing, including radiation pattern measurements and impedance measurements. **Tutorial 4**: Deploy Antennas in Wireless Networks • Case studies and simulations on deploying antennas in practical wireless network scenarios, considering factors like coverage area, interference, and capacity. **Practical Work:** **Practical Work 1**: Fabricate and Test Dipole Antennas • Fabricate dipole antennas and measure their radiation patterns and impedance characteristics in a laboratory environment.

	Practical Work 2: Simulate and Optimize Antenna Arrays ● Conduct simulation-based exercises on designing and optimizing antenna arrays using software tools, focusing on beamforming and spatial filtering techniques. **Practical Work 3**: Measure Propagation Characteristics in the Field ● Conduct field trips to measure propagation characteristics in different environments and compare them with theoretical models. **Practical Work 4**: Innovate Antenna Designs ● Collaborate on group projects proposing and developing innovative antenna designs to address specific communication challenges, culminating in prototype implementation and performance evaluation.
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S), a practice test and a final written test (Exam) are planned
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain basic and intermediate knowledge of Huawei Wifi(Accesscontroller and AccessPoint) equipments configurations By the end of the final exam, students should be able pass the HCIA WLAN Exam
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Develop the ability to analyze and evaluate the performance of different types of antennas, including radiation patterns, impedance matching, and gain characteristics. 2. Learn to design antenna systems tailored to specific requirements, considering factors such as frequency bands, polarization, and environmental conditions. 3. Gain proficiency in modeling propagation environments and predicting signal propagation characteristics, including path loss, shadowing, and multipath fading effects. 4. Explore advanced antenna technologies such as phased arrays, metamaterials, and MIMO systems, understanding their principles and applications in modern communication systems.
Teaching Manual:	● Huawei Palteform ● Huawei Simulator.

	• Powerpoint presentation • Tutorial booklet • Huawei Workbooks
References:	[1] Balanis, C. A. (2016). Antenna Theory: Analysis and Design (4th ed.). Wiley. [2] Kraus, J. D., & Marhefka, R. J. (2002). Antennas for All Applications (3rd ed.). McGraw-Hill Education. [3] Stutzman, W. L., & Thiele, G. A. (2012). Antenna Theory and Design (3rd ed.). Wiley. [4] Cheng, D. K. (1989). Field and Wave Electromagnetics (2nd ed.). Addison-Wesley.

Department of Computer Science

SE-C-4.7.1-Collaborative development and E-Commerce

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Software Engineering 2</td><td>Course</td><td colspan="2">Collaborative development and E-Commerce</td><td>Code</td><td>SE-C-4.7.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="4">Semester</td><td rowspan="4">4</td></tr>
<tr><td>Workshop</td><td>21h 00</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>42h00</td></tr>
<tr><td colspan="3">Total workload</td><td colspan="3">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="3"></td><td colspan="3"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="3">Responsible Teacher</td><td colspan="2">Hatem Sandid</td><td>Involved Teacher(s)</td><td colspan="2">Hatem Sandid</td></tr>
<tr><td colspan="3" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1 & Ob2</td></tr>
<tr><td>PLO</td><td colspan="2">P1: Systems Specification and Design Methodologies
P3: Innovation, Research and Emerging Technologies
P6:Project Management and Leadership
P8: Communication and Collaboration</td></tr>
<tr><td colspan="2">Prerequisites:</td><td colspan="6">Web development 1 & 2</td></tr>
<tr><td colspan="2">Corequisites:</td><td colspan="6">Agile and Hybrid Methods</td></tr>
<tr><td colspan="2">Learning outcomes</td><td colspan="6">Upon completion of this teaching module, the student will be able to:

1. Understand the foundational principles of collaborative development and e-commerce, including software architecture, design patterns, content management systems (CMS), and e-commerce platforms.
2. Evaluate and select appropriate architectural styles and technologies for e-commerce projects, considering factors such as scalability, flexibility, and maintainability.
3. Demonstrate proficiency in setting up and configuring CMS for e-commerce websites, integrating them seamlessly with e-commerce platforms.
4. Apply collaborative development tools and practices effectively, including version control systems, issue tracking, and continuous integration, to facilitate teamwork and project management.
5. Design and develop a fully functional e-commerce website, incorporating frontend and backend technologies to provide a seamless user experience.</td></tr>
</table>

	6. Implement security measures and ensure compliance with relevant regulations (e.g., GDPR, PCI DSS) to safeguard user data and transactions in e-commerce environments. 7. Utilize digital marketing strategies to promote e-commerce websites, enhance visibility, attract customers, and drive sales. 8. Analyze emerging trends in collaborative development and e-commerce, and propose innovative solutions to address future challenges and opportunities in the field.
Content:	**Chapter 1: Introduction to Collaborative Development and E-Commerce** 1. Understanding the Concepts of Collaborative Development 2. Overview of E-Commerce and its Importance 3. Trends in Collaborative Development and E-Commerce 4. Case Studies: Successful Collaborative Development and E-Commerce Platforms **Chapter 2: Content Management Systems (CMS) Fundamentals** 1. Introduction to Content Management Systems (CMS) 2. Types of CMS: Open Source vs. Proprietary 3. Evaluating and Choosing a CMS for E-Commerce 4. Hands-on: Setting Up a CMS for E-Commerce Website **Chapter 3: E-Commerce Platforms and Technologies** 1. Overview of E-Commerce Platforms (e.g., Shopify, WooCommerce, Magento) 2. Features and Capabilities of E-Commerce Platforms 3. Integrating CMS with E-Commerce Platforms 4. Case Studies: E-Commerce Success Stories **Chapter 4: Collaborative Development Tools and Practices** 1. Introduction to Collaborative Development Tools (e.g., Git, GitHub, Bitbucket) 2. Collaborative Development Workflows (e.g., Agile, Scrum, Kanban) 3. Version Control and Branching Strategies for Collaborative Projects 4. Hands-on: Collaborative Development Setup and Workflow **Chapter 5: Designing and Developing an E-Commerce Website** 1. E-Commerce Website Design Principles 2. Frontend Development Technologies (e.g., HTML, CSS, JavaScript) 3. Backend Development Technologies (e.g., PHP, Python, Node.js) 4. Building an E-Commerce Website with CMS Integration **Chapter 6: Security and Compliance in E-Commerce** 1. Importance of Security in E-Commerce 2. Common Security Threats and Vulnerabilities 3. Implementing Security Measures in E-Commerce Websites

4. Compliance Requirements (e.g., GDPR, PCI DSS) for E-Commerce

Chapter 7: Marketing and Analytics for E-Commerce

1. Digital Marketing Strategies for E-Commerce
2. Implementing SEO and SEM Techniques
3. Utilizing Analytics Tools for E-Commerce Optimization
4. Case Studies: Effective Marketing Campaigns in E-Commerce

Chapter 8: Future Trends in Collaborative Development and E-Commerce

1. Emerging Technologies Impacting Collaborative Development and E-Commerce
2. Predictions for the Future of E-Commerce
3. Innovations in Collaborative Development Practices
4. Ethical Considerations in Collaborative Development and E-Commerce

Content of the Practical Work :

1. Installation and Configuration of WordPress CMS
2. Creating a Showcase Website with WordPress
3. Exploring and Utilizing WordPress Plugins
4. Installation and Configuration of WooCommerce Plugin
5. Reviewing Electronic Payment Methods and Security Issues in E-Commerce
6. Project Presentation: Exploring and Configuring E-Commerce-oriented CMS (Shopify, WooCommerce, Magento)
7. Understanding Collaborative Work Tools: Presentation and Roles
8. Utilizing Trello for Project Management and GIT for Version Control
9. Implementing Security Measures for E-Commerce Transactions
10. Designing and Implementing a Marketing Campaign for an E-Commerce Website

—----------

Self-study project and search topics:

Project: Building a Collaborative E-Commerce Marketplace
Develop a web-based marketplace platform that allows multiple vendors to sell their products/services collaboratively. Users should be able to create accounts, list products, manage orders, and process payments. Implement collaborative features such as vendor collaboration, user reviews, and real-time communication. Integrate popular CMS and e-commerce platforms, along with collaborative development tools, to create a comprehensive solution.

Search topic: Impact of Blockchain Technology on Collaborative E-Commerce
Investigate how blockchain technology can revolutionize collaborative e-commerce platforms. Explore the potential benefits such as enhanced security, transparent transactions, and decentralized control. Analyze case studies of blockchain-powered e-commerce marketplaces and examine the challenges and

	opportunities in implementing blockchain in collaborative development projects within the e-commerce domain.
Teaching and learning method(s):	Face to face (face-to-face) 42 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2 and 3** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes , 4 , 5, 6, 7 and 8**
Assessment criteria(s) :	During the midterm exam, students should be able to comprehend the foundational concepts of collaborative development and e-commerce, including software architecture, design patterns, CMS, e-commerce platforms, and collaborative development tools. By the end of the final exam, students should be proficient in designing and developing a functional e-commerce website integrated with a CMS. They should also demonstrate competence in implementing security measures, complying with regulations, utilizing marketing strategies, and analyzing emerging trends in collaborative development and e-commerce.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Understand the concepts and types of e-commerce 2. Develop Software Installation and Configuration Skills 3. Master Web Development and Design Abilities 4. Able to Manage and Integrate Plugins Effectively 5. Develop Platform Setup and Configuration Proficiency 6. Master Payment Systems and Security Awareness 7. Develop Project Management and Presentation Skills 8. Master Collaboration Tools (Trello, Git, etc.) 9. Able to Implement Version Control and Task Management Effectively 10. Develop Security Implementation Skills 11. Master Marketing Strategy Development Abilities
Teaching Manual:	PW booklet of Mr. Hatem Sandid
References:	[1] R. S. Pressman, "Software Engineering: A Practitioner's Approach," McGraw-Hill Education, 2015.

[2] J. F. Rayport and B. J. Jaworski, "Introduction to E-Commerce," McGraw-Hill Education, 2016.

[3] R. C. Martin, "Agile Software Development: Principles, Patterns, and Practices," Pearson, 2002.

[4] T. Felke-Morris, "Web Development and Design Foundations with HTML, CSS, and JavaScript," Pearson, 2021.

[5] Git official website https://git-scm.com/

[6] Wordpress Official website wordpress.org/support

Department of Computer Science
SE-C-4.7.2-Agile and hybrid methods

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Software Engineering 2</td><td>Course</td><td colspan="2">Agile and hybrid methods</td><td>Code</td><td>SE-C-4.7.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">4</td></tr>
<tr><td>Workshop</td><td>21h 00</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>42h00</td><td colspan="2"></td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Wael Sellami</td><td>Involved Teacher(s)</td><td colspan="2">Mr Wael Sellami</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1 & Obj2</td></tr>
<tr><td>PLO</td><td colspan="2">P1: Systems Specification and Design Methodologies
P3: Innovation, Research and Emerging Technologies
P6:Project Management and Leadership
P8: Communication and Collaboration</td></tr>
</table>

Prerequisites:	Nothhing
Coequisites:	Nothhing
Learning outcomes	Upon completion of this teaching module, the student will be able to • understand some agile methodology like scrum to help them to be integrated in the Professional world (communications soft skills and …) • choose the right approach for a specific project, considering factors like project size, complexity, industry, and stakeholder needs.
Content:	**Introduction to Project Management:** • Understanding what a project is. • Differentiating between projects, programs, and portfolios. • Exploring the role of a project manager. • Overview of project management processes. **Project Environment:** • Factors influencing a project's success. • Stakeholders and their roles.

	● Organizational structures and their impact on projects. **Role of the Project Manager:** ● Project manager's responsibilities. ● Leadership vs. management in project management. ● Ethics and professional conduct. **Project Integration Management:** ● Developing a project charter. ● Developing a project management plan. ● Directing and managing project work. ● Monitoring and controlling project work. ● Closing a project or phase. **Project Scope Management:** ● Defining project scope. ● Creating a Work Breakdown Structure (WBS). ● Scope validation and control. **Project Resource Management:** ● Identifying and acquiring project resources. ● Developing the project team. ● Managing the project team. **Project Communications Management:** ● Planning for project communications. ● Managing and controlling project communications. ● Communication channels and methods. **Project Stakeholder Management:** ● Identifying project stakeholders. ● Stakeholder engagement and communication. **Professional and Social Responsibility:** ● Ethical considerations in project management. ● PMI's Code of Ethics and Professional Conduct. **Practice Exams and Review:** ● Practice exams to assess your knowledge.
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh platform, based on the PMP book “PMBOCK V7”, to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with activities, use cases and workshops
Assessment method(s):	A midterm written test (D.S), a practice test and a final written test (Exam) are planned.

Assessment criteria(s) :	During the midterm exam, students should be able to define and explain …resolve . By the end of the final exam, students should be able to understand how to be a good project manager and to choose the right approach for a specific project, considering factors like project size, complexity, industry, and stakeholder needs.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	Students will be able to understand some agile methodology like scrum to help them to be integrated in the Professional world (communications soft skills and …)
Teaching Manual:	● PMBOCK v7 guide
References:	[1] PMBOCK v7 guide

Department of Computer Science
BDBI-C-4.7.1-Data Mining 1

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Big Data and Business Intelligence 2</td><td>Course</td><td colspan="2">Data Mining 1</td><td>Code</td><td>BDBI-C-4.7.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Big Data and Business Intelligence</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="4">Semester</td><td rowspan="4">4</td></tr>
<tr><td>Workshop</td><td>21h 00</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>42h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Wael Sellami</td><td>Involved Teacher(s)</td><td colspan="2">Mr Mohamed Manaa</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1 & Obj2</td></tr>
<tr><td>PLO</td><td colspan="2">P1: Systems Specification and Design Methodologies
P3: Innovation, Research and Emerging Technologies
P6:Project Management and Leadership
P8: Communication and Collaboration</td></tr>
<tr><td>Prerequisites:</td><td colspan="7"></td></tr>
<tr><td>Corequisites:</td><td colspan="7"></td></tr>
<tr><td>Learning outcomes</td><td colspan="7">Upon completion of this teaching module, the student will be able to:

– understand Data Mining Concepts
– understand advanced concepts of Data Mining
– apply sophisticated techniques for advanced data preparation</td></tr>
<tr><td>Content:</td><td colspan="7">Chapter 1: Advanced Understanding of Data Mining
1. Classification
2. Regression
3. Clustering
4. association methods
Chapter 2: Mastery of Data Mining Tools
1. advanced data mining tools such as R, Python (scikit-learn, pandas)</td></tr>
</table>

	2. commercial solutions for data analysis **Chapter 3: Advanced Data Preparation** 1. collection 2. Cleaning 3. Transformation 4. integration **Chapter 4: Advanced Data Mining Algorithms** 1. support vector machines (SVM) 2. neural networks 3. advanced decision trees 4. ensemble methods **Chapter 5: Evaluation of Data Mining Models** 1. advanced metrics 2. area under the ROC curve (AUC) 3. recall 4. F1-score
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 2.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain Data Mining and mastery of Data Mining Tools. By the end of the final exam, students should be able to manage complex projects, apply sophisticated methods.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Understand Data Mining and data preparation 2. Manage complex projects, apply sophisticated methods
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks

References:	[1] Data Mining et Statistique décisionnelle, Stéphane Tufféry, Editions Technip 2017 [2] Data mining Découverte de connaissances dans les données, Daniel T. Larose, Chantal D. Larose, Vuibert Edition 2018 [3] Exploration de Données et Méthodes Statistiques Data Analysis & Data Mining avec le Logiciel R, Lise Bellanger, Richard Tomassone, Edition ELLIPSES 2014

Department of Computer Science
BDBI-C-4.7.2-BI Tools

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Big Data and Business Intelligence 2**		**Course**	**BI Tools**		**Code**	**BDBI-C-4.7.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Big Data and Business Intelligence**
Type of education	**Lecture**	**21h 00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**4**
	Workshop	**21h 00**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**42h00**					
Total workload	**84h00 / Semester**				**Coefficient**		**3**
					ECTS credit		**3**
Responsible Teacher	**Mr Wael Sellami**		**Involved Teacher(s)**	**Mr Mohamed Manaa**			
Review committee	**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1 & Obj2**			
			PLO	**P1: Systems Specification and Design Methodologies P3: Innovation, Research and Emerging Technologies P6:Project Management and Leadership P8: Communication and Collaboration**			

Prerequisites:	DataBases, data analysis concepts
Corequisites:	Data Mining, Machine Learning
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Demonstrate a deep understanding of BI concepts, principles, and its strategic significance in modern organizations. 2. Design and implement complex data warehousing solutions, including data modeling, ETL processes, and data integration. 3. Evaluate and select appropriate BI tools based on specific business needs, considering factors like data volume, user requirements, and licensing. 4. Master various BI tools, such as Tableau, Power BI, QlikView, or MicroStrategy, including their advanced features and functionalities.
Content:	**Chapter 1: Introduction to Business Intelligence** 1. Understanding Business Intelligence and its importance 2. Historical overview of BI

	3. Key concepts: data, information, knowledge 4. The role of BI in decision-making **Chapter 2: Data Warehousing** 1. Introduction to data warehousing 2. Data modeling and design 3. Data integration and ETL (Extract, Transform, Load) processes 4. Data quality and cleansing **Chapter 3: BI Tools Overview** 1. Types of BI tools: Reporting, OLAP, Data Mining, and more 2. Comparison of popular BI tools (e.g., Tableau, Power BI, QlikView, MicroStrategy) 3. Selection criteria for choosing the right BI tool 4. Licensing and cost considerations **Chapter 4: Data Visualization and Reporting** 1. Data visualization principles 2. Creating effective data visualizations 3. Designing interactive dashboards 4. Report generation and distribution **Chapter 5: Data Analytics** 1. Introduction to data analytics 2. Descriptive, diagnostic, predictive, and prescriptive analytics 3. Statistical analysis and data exploration 4. Machine learning for business **Chapter 6: Advanced BI Concepts** 1. Big data and BI 2. Real-time BI and streaming data 3. Mobile BI and accessibility 4. Security and data governance in BI **Chapter 7: Business Decision Support** 1. Using BI for strategic decision-making 2. Case studies and business scenarios 3. Business Intelligence best practices 4. Preparing for BI implementation and adoption **Chapter 8: Practical BI Project** 1. Group or individual BI project 2. Hands-on experience with BI tools 3. Project presentation and documentation
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1 and 2.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 3 and 4.**

Assessment criteria(s) :	During the midterm exam, students should be able to define and explain BI Tools. By the end of the final exam, students should be able to encompass advanced knowledge and skills related to business intelligence tools, data analysis, and decision support .
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	Encompass advanced knowledge and skills related to business intelligence tools, data analysis, and decision support
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Business Intelligence Tools for Small Companies: A Guide to Free and Low-Cost Solutions, Albert Nogués, Juan Valladarew, Apress Edition 2017 [2] Microsoft Business Intelligence Tools for Excel Analysts, Michael Alexander, Wiley Edition 2014 [3] Data Visualisation, De l'extraction des données à leur représentation graphique, Xavier Guesnu, Eyrolles Edition 2013

Department of Computer Science
C-5.1.1-Cloud-Native Development

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Advanced Topic 3**		**Course**	**Cloud-Native Development**		**Code**	**C-5.1.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**--**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**5**
	Workshop	**21h 00**					
	Project	**21h 00**		**Mixed**	**XX**		
	Self Study	**14h00**					
Total workload		**56h00 / Semester**			**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher		**Mr Wael Sellami**		**Involved Teacher(s)**	**Mr Wael Sellami**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1**		
				PLO	**P1 : Systems Specification and Design Methodologies P2: Implementation Techniques P3: Innovation, Research and Emerging Technologies P5: Network and Information Systems Management**		

Prerequisites:	Fundamental software development principles, including design patterns, version control, and software testing methodologies. Knowledge of key cloud computing concepts such as virtualization, scalability, and cloud service models (IaaS, PaaS, SaaS).
Corequisites:	Basics of cloud computing to provide students with a strong foundation in the subject. This could be taken concurrently with the Cloud-Native Development course to ensure that all students have a similar background.
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Develop the ability to design and architect cloud-native applications, considering microservices, scalability, and resilience principles. 2. Master the use of containerization tools like Docker and container orchestration with Kubernetes for efficient application deployment. 3. Acquire skills to create, deploy, and manage microservices-based applications, understanding communication patterns and service discovery. 4. Apply Cloud-Native Tools: Utilize cloud-native development tools and frameworks, including serverless computing and monitoring, for enhanced application performance and security.

	5. Develop the ability to identify and resolve challenges in cloud-native development through practical experience and critical thinking.
Content:	**Chapter 1: Introduction to Cloud-Native Development** 1. Introduction 2. Overview of cloud-native principles and methodologies 3. Cloud-native vs. traditional development approaches 4. Importance of cloud-native development in modern software engineering **Practical Work:** TP1: Hands-on exercises and simulations to explore cloud-native development fundamentals. **Chapter 2: Containerization and Orchestration** 1. Understanding containerization using Docker 2. Kubernetes: The leading container orchestration platform 3. Deploying and managing applications in containers **Practical Work:** TP2: Practical exercises involving Docker and Kubernetes to deploy containerized applications. **Chapter 3: Microservices Architecture** 1. Introduction to microservices and their benefits 2. Designing and building microservices applications 3. Communication patterns, API gateways, and service discovery **Practical Work:** TP3: Building a microservices-based application and implementing communication patterns. **Chapter 4: Scalability and Resilience** 1. Introduction 2. Horizontal and vertical scaling of cloud-native applications 3. Load balancing and auto-scaling strategies 4. High availability and fault tolerance in cloud-native systems **Practical Work:** TP4: Practical scenarios and exercises related to scaling and ensuring application resilience. **Chapter 5: Tools and Frameworks** 1. Introduction 2. Exploring cloud-native development tools and frameworks 3. Serverless computing and event-driven architectures 4. Monitoring, logging, and security in cloud-native applications **Practical Work:** TP5: Utilizing various cloud-native development tools and frameworks to build and secure cloud-native applications. **Project :** Project Title: Cloud-Native Application Development Project Description:

	Students will collaboratively develop a cloud-native application from inception to deployment, applying course principles. Research topics include serverless computing, microservices orchestration, containerization best practices, pipelines automation, and cloud-native data management. Through hands-on development and exploration of these topics, students will gain practical skills and deepen their understanding of cloud-native technologies. **Self-study:** 1. Research Understand Cloud Basics: Select a Cloud Platform: Explore Cloud Concepts: Study Use Cases 2. Proposal Estimate the costs associated with your project. Plan the technical architecture, including the choice of cloud services, scalability, security, and data management. 3. Implementation Sign up for an account with the cloud provider you've selected. Install necessary tools like the command-line interface (CLI) for your chosen cloud platform and Docker (if relevant). 5. Presentation (Optional) Share project findings with the class if desired. 6. Doing Exercise and QCM published in easylearn.ovh platform.
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1 and 2** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes ., 3,4 and 5**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain fundamental cloud-native development principles, technologies, and advantages. By the end of the final exam, students should be able to demonstrate comprehensive proficiency in designing, building, and deploying cloud-native applications, as well as critically evaluate their architecture and performance.
Assessment measure(s):	Assigning of a final average

Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	● Proficiency in designing cloud-native applications with a focus on microservices, scalability, and resilience. ● Competence in using containerization (e.g., Docker) and container orchestration (e.g., Kubernetes) for application deployment. ● Skill in implementing communication patterns, API gateways, and service discovery within microservices-based applications. ● Expertise in ensuring horizontal scaling, load balancing, high availability, and fault tolerance in cloud-native systems.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] John Author. "Cloud-Native Development: Principles and Practices." Publisher, 2023. [2] Emily Smith. "Docker and Kubernetes for Containerization and Orchestration." Containerization Journal, 5(2), 100-120, 2020. [3] Jane Doe. "Microservices: Design and Implementation." Publisher, 2021. [4] David Brown. "Scalability and Resilience in Cloud-Native Applications." In Proceedings of Cloud Engineering Conference, pp. 75-84, 2022.

Department of Computer Science
C-5.1.2-Big Data

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Advanced Topic 3**		**Course**	**Big Data**		**Code**	**C-5.1.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h 00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**5**
	Workshop	**21h 00**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**42h00**					
Total workload		**84h00 / Semester**			**Coefficient**		**3**
					ECTS credit		**3**
Responsible Teacher		**Mr Wael Sellami**		**Involved Teacher(s)**	**Mr Mohamed Manaa**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1**		
				PLO	**P1 : Systems Specification and Design Methodologies P2: Implementation Techniques P3: Innovation, Research and Emerging Technologies P5: Network and Information Systems Management**		

Prerequisites:	Databases, networking fundamentals, SQL
Corequisites:	Cloud Computing, Machine Learning
Learning outcomes	At the end of this teaching module, the student will be able to: 1. Understand the architecture of a Big Data platform 2. Use Big Data platform and methodologies to collect and analyze large amounts of data from different sources 3. Know and use Apache Hadoop, Ambari, Spark, Big SQL, HDFS, YARN, MapReduce, ZooKeeper, Knox, Sqoop and HBase. 4. Define the Hadoop ecosystem 5. Understand the concept of NoSql Databases
Content:	**Chapter 1: Introduction to Big Data and Data Analytics** a- Introduction to Big Data b- The 4 Vs of Big Data

	c- Types of Big Data d- Use cases for a Big Data platform **Chapter 2: Hadoop: HDFS/MapReduce** a- What is Hadoop? b- HDFS architecture c- MapReduce programming model d- MapReduce and YARN e- Apache Spark **Chapter 3: Hadoop Query Languages** a- Databases vs Hadoop b- Pig c- Hive d- JAQL **Chapter 4 : Big SQL** a- What is Big SQL? b- Big SQL Architecture c- Using Big SQL d- Creating Big SQL schemas and tables e- Data federation with Big SQL **Content of the Practical Work:** **TP1: Hadoop installation and configuration** **TP2: HDFS** **TP3: MapReduce** **TP4: MongoDB**
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,2 and 3** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 4 and 5.**

Assessment criteria(s) :	During the midterm exam, students should be able to define BigData, Hadoop, HDFS and MapReduce At the end of the final exam, students should be able to illustrate SPARK, HQL and choose one of the data collection and analysis methods and create Big Sql queries.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Define Big Data 2. Know Apache Hadoop, Ambari, Spark, Big SQL, HDFS, YARN, MapReduce, ZooKeeper, Knox, Sqoop and HBase 3. Choose a scientific method for data analysis.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Brian Godsey: Think Like a Data Scientist, Manning Publications, 2017 [2] IBM Big Data Engineer official certification course, IBM Academy Skills; 2018

Department of Computer Science
C-5.2.1-Programing paradigms

Last Update Date: February 17, 2024

Syllabus Sheet

Unit	**Advanced Computing 3**		**Course**	**Programing paradigms**		**Code**	**C-5.2.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h 00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**5**
	Workshop	**21h 00**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**42h00**					
Total workload		**84h00 / Semester**			**Coefficient**		**3**
					ECTS credit		**3**
Responsible Teacher		**Mr Mohamed Amine Omrani**		**Involved Teacher(s)**	**Mr Mohamed Amine Omrani**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1**		
				PLO	**P2: Implementation Techniques P3: Innovation, Research and Emerging Technologies**		

Prerequisites:	Proficiency in algorithms, classical and/or object-oriented programming, and mastery of the multiparadigm programming language (Python)
Corequisites:	Python
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Master the Python IDE development environment. 2. Understand classical programming paradigms (Imperative paradigm, Structural paradigm, Recursive paradigm). 3. Familiarize themselves with the Declarative programming paradigm. 4. Familiarize themselves with the Functional programming paradigm. 5. Gain proficiency in the Object-Oriented programming paradigm. 6. Understand sequential, parallel, and concurrent programming paradigms. 7. Gain knowledge of the Aspect-Oriented programming paradigm. 8. Acquire an understanding of the Logic programming paradigm. 9. Comprehend mathematical programming paradigms (such as Constraint Programming, CPC).

	10. Acquire and deepen their skills in multiparadigm programming and explore various paradigms. **11.** Solve complex problems using different paradigms.
Content:	**Chapter 1: Introduction to Programming Paradigms** 1. Historical Overview of Paradigms 2. Why Multiple Paradigms? **Chapter 2: The Python Environment** 1. Introduction 2. Python IDE / PyCharm and Python Virtual Environment 3. Integration of External Libraries 4. Python Syntax Review 5. Programming with Python **Workshop** **Chapter 3: Classical Paradigms** 1. Introduction 2. Imperative Programming 3. Structural Programming 4. Procedural Programming 5. Recursive Programming 6. Declarative Paradigm **Workshop** **Chapter 4: Functional Programming** 1. Introduction 2. Definitions 3. Limitations of Classical Paradigms 4. Strengths of Functional Programming 5. Limitations of Functional Programming 6. Problem Solving with Functional Programming **Workshop** **Chapter 5: Object-Oriented Programming with Python** 1. Introduction 2. Definitions 3. Limits of OOP (Object-Oriented Programming) 4. Advantages of Object-Oriented Programming 5. OOP vs. Functional Programming 6. Python as an Object-Oriented Language **Workshop** **Chapter 6: Parallel Programming** 1. Introduction 2. Definitions 3. Sequential Programming 4. Advantages of Parallel Programming 5. Parallelism (Multiprocessing & Multithreading) 6. Python Multithreading

	7. Python Multiprocessing **Workshop**: (Implementation of real-world image downloading case with Sequential, Multiprocessing, and Multithreading), analysis, and interpretations. **Chapter 7: Aspect-Oriented Programming** 1. Introduction 2. Definitions 3. Joinpoints & Pointcuts 4. Advice 5. Aspect 6. Weaving & Weaver 7. Inter-type Declarations 8. Python AOP / SpringBoot AOP **Workshop**
	Self-study & Project 1: Implement a Sudoku Solver: **Description**: In this project, students can build a Sudoku solver using Python, which will involve both logic programming and constraint programming. Sudoku puzzles can be represented as a logical constraint satisfaction problem. Students can use the python-constraint library to model and solve Sudoku puzzles. They will need to define the rules, facts, and constraints of Sudoku and create a program that can solve various Sudoku puzzles. **Skills Practiced: [**Logic programming, Constraint programming, Problem modeling Python programming **]** **Self-study & Project 2: Parallel Image Downloader:** **Description**: In this project, students can implement an image downloader that can download a large number of images from the internet concurrently using Python's multiprocessing and multithreading capabilities. They can start with a sequential implementation, then create parallel versions using multiprocessing and multithreading. Students can compare the performance of the three implementations in terms of download speed and resource utilization. They should also analyze and interpret the results to understand the advantages and disadvantages of parallel programming. **Skills Practiced: [** Parallel programming, Multiprocessing and multithreading Real-world problem-solving, Performance analysis **]**
Teaching and learning method(s):	Face to face 42 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2, 3, 4 and 5.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 6, 7, 8, 9, 10 and 11.**

Assessment criteria(s) :	At the midterm test, students should be able to define the fundamental concepts of the Python environment and understand the classical, functional, and object-oriented paradigms, including their limitations and strengths. By the end of the final exam, students should be capable of applying advanced paradigms such as logic programming, aspect-oriented programming, or constraint programming to solve complex problems.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Paradigm Understanding: Understand various programming paradigms, including their strengths and weaknesses. 2. Advanced Problem Solving: Apply advanced paradigms (e.g., logic, aspect-oriented) to solve complex problems. 3. Python Proficiency: Master the Python language for diverse programming tasks. 4. Parallel Programming Skills: Develop expertise in multiprocessing and multithreading for efficient program execution. 5. Critical Analysis: Analyze, compare, and choose the best paradigm for specific tasks based on problem characteristics.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Python Documentation. "Python 3 Tutorial." Available at: https://docs.python.org/fr/3/tutorial/ [2] Boucheny, Vincent. "Learning Object-Oriented Programming with the Python Language - (with practical exercises and solutions)." [3] Python Documentation. "Python multiprocessing module." Available at: https://docs.python.org/3/library/multiprocessing.html [4] Labix. "Python-Constraint Library." Available at: https://labix.org/python-constraint [5] Python Package Index (PyPI). "Kanren: A Declarative Logic Programming Library for Python." Available at: https://pypi.org/project/kanren/ [6] IBM. "IBM ILOG CPLEX Optimization Studio Documentation." Available at: https://www.ibm.com/docs/en/SSSA5P_12.8.0/ilog.odms.studio.help/pdf/gscplex.pdf [7] Python Package Index (PyPI). "Numberjack: A Constraint Programming Library." Available at: https://pypi.org/project/Numberjack/ **[8] ResearchGate. "On the Generation of Constrained Graphs." Available at: https://www.researchgate.net/publication/357539347_On_the_generation_of_constrained_graphs/** [9] M. Scott, "Programming Language Pragmatics," Morgan Kaufmann, 2005.

[10] B. Meyer, "Object-Oriented Software Construction," Prentice Hall, 1997.

[11] P. Hudak, "Conception, Evolution, and Application of Functional Programming Languages," ACM Computing Surveys, 1989.

Department of Computer Science
C-5.2.2-Data warehouse

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Advanced Computing 3**		**Course**	**Data warehouse**		**Code**	**C-5.2.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**5**
	Workshop	**10,5h**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**25h**					
Total workload		**56h00 / Semester**			**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher		**Mr Mounir Zrigui**		**Involved Teacher(s)**	**Mr Mounir Zrigui**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1**		
				PLO	**P1 : Systems Specification and Design Methodologies P2: Implementation Techniques P3: Innovation, Research and Emerging Technologies P5: Network and Information Systems Management**		

Prerequisites:	
Corequisites:	
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Focus on advanced mastery of skills in data management, 2. Master Data Warehouse design, 3. Define ETL (Extract, Transform, Load), 4. Implement tools of data analysis and reporting.
Content:	**Chapter 1: Introduction to Data Warehousing** Definition and importance of data warehousing Evolution and history of data warehousing Characteristics and architecture of data warehouses

	Comparison with operational databases **Chapter 2: Data Warehouse Design** Dimensional modeling (star schema, snowflake schema) Fact tables and dimension tables Data granularity ETL (Extract, Transform, Load) processes Data cleansing and integration **Chapter 3: Data Warehouse Implementation** Tools and technologies for building data warehouses (e.g., SQL Server, Oracle, Teradata, Snowflake) Data warehouse lifecycle management Performance tuning and optimization Security considerations **Chapter 4: Data Warehouse Administration** Backup and recovery strategies Monitoring and maintenance tasks User management and access control Disaster recovery planning **Chapter 5: Data Warehouse Querying and Reporting** SQL for querying data warehouses OLAP (Online Analytical Processing) concepts Data visualization tools (e.g., Tableau, Power BI) Creating dashboards and reports
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1 and 2.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 3 and 4.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain data management, Data Warehouse design, data analysis and project management.

	By the end of the final exam, students should be abledata management, Data Warehouse design, data analysis and project management. They must be adapted according to the specific objectives of the program.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	Participants should be able to solve complex problems related to data management and Data Warehouse operations.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Le data warehouse Guide de conduite de projet, Ralph Kimball, Laura Reeves, Margy Ross, Warrren Thornthwaite, Eyrolles Edition, 2005, ISBN : 978-2-212-11600-7 [2] Cycle de vie des entrepôts de données, Franck Ravat, Olivier Teste, Gilles Zurfluh, Hermès - Lavoisier 2009, ISBN : 978-2-7462-2352-3

Department of Computer Science
C-5.3.1-Web Development 5

Last Update Date: February 17, 2024

Syllabus Sheet

Unit	**Advanced Web**		**Course**	**Web Development 5**		**Code**	**C-5.3.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**10,5h**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**5**
	Workshop	**21h 00**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**25h00**					
Total workload		**56h00 / Semester**			**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher		**Mr. Mohamed Amine Omrani**		**Involved Teacher(s)**	**Mr Mohamed Amine Omrani**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1**		
				PLO	**P1 : Systems Specification and Design Methodologies P2: Implementation Techniques P3: Innovation, Research and Emerging Technologies P4: Web and Software Development P5: Network and Information Systems Management**		

Prerequisites:	Proficiency in Java programming language. Understanding of web development concepts (HTML, CSS, HTTP). Familiarity with SQL and relational databases.
Corequisites:	
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Spring Boot Mastery: Mastering Spring Boot from foundational to advanced levels. 2. RESTful API Development: Designing and building robust RESTful APIs using Spring Boot. 3. Database Integration: Integrating Spring Boot applications with relational databases effectively. 4. Security Implementation: Implementing advanced security measures for Spring Boot applications. 5. Integration with Frontend: Integrating Spring Boot backend with frontend frameworks for full-stack development. 6. Microservices Architecture: Designing and developing microservices-based systems with Spring Boot.

	7. Advanced mode : Deploying, Testing and Quality Assurance: Conducting comprehensive testing and ensuring software quality.
Content:	**Chapter 1: Introduction to Spring Boot** 1. Spring Boot features and advantages. 2. Project setup and initialization. 3. Spring Boot application structure. **Chapter 2: RESTful API Development with Spring Boot** 1. Designing RESTful endpoints. 2. Building RESTful controllers. 3. Request and response handling. **Chapter 3: Database Integration with Spring Boot** 1. Data access with Spring Data JPA. 2. Advanced JPA Features. 3. Database configuration and connection. 4. CRUD operations with Spring Boot and Hibernate. **Chapter 4: Security in Spring Boot** 1. Authentication and authorization. 2. OAuth 2.0 and JWT for secure APIs. 3. Implementing security best practices. **Chapter 5: Microservices with Spring Boot** 1. Microservices architecture and principles. 2. Building microservices with Spring Boot. 3. Service discovery and communication. **Chapter 6: Production Mode [Testing,Quality Assurance and Deployment]** 1. Unit testing and test-driven development (TDD). 2. Integration testing with Spring Boot. 3. Deployment 4. Containerization with Docker. 5. Overview of deployment to cloud platforms (e.g., AWS, Azure).
	Project : "E-commerce Platform Enhancement with Spring Boot" **Project Description:** In this project, students will have the opportunity to enhance an existing e-commerce platform using Spring Boot. They will work on various aspects of the platform to improve its functionality, performance, and security. This project provides hands-on experience in building a real-world web application and mastering Spring Boot development. **Project Objectives:** Enhance the existing e-commerce platform with new features and improvements. Implement advanced security measures to protect user data and transactions. Optimize the platform for better performance and scalability. Develop a RESTful API to enable integration with external services. Conduct thorough testing and debugging to ensure the application's robustness. **Self-Study Tasks:**

	Students will be responsible for self-study tasks, including: ● Spring Boot Documentation: Explore the official Spring Boot documentation to deepen your understanding of the framework's features and capabilities. ● Advanced Security Concepts: Research and learn about advanced security concepts such as OAuth 2.0 and JWT for implementing secure authentication and authorization. ● Performance Optimization Techniques: Study techniques for optimizing the performance of Spring Boot applications, including profiling and load testing. ● RESTful API Design: Explore best practices for designing and documenting RESTful APIs with Spring Boot. ● Testing and Test Automation: Learn about unit testing, integration testing, and test automation frameworks for Spring Boot applications. **Project Tasks:** ● Feature Enhancement: Collaborate to enhance specific features of the e-commerce platform, such as user profiles, product search, or order processing. ● Security Implementation: Implement advanced security measures, including OAuth 2.0-based authentication and JWT token validation. ● Performance Optimization: Identify and address performance bottlenecks in the application, making it more responsive and efficient. ● RESTful API Development: Develop a RESTful API for external partners to access the platform's data and services. ● Testing and Debugging: Write comprehensive unit tests and conduct integration testing. Debug and fix any issues that arise during testing. **Deliverables:** ● Enhanced e-commerce platform with added features and improved performance. ● Comprehensive documentation on security implementation, API design, and performance optimization. ● Test suites with detailed test cases and reports. ● A final project report summarizing the enhancements and lessons learned. **Evaluation Criteria:** ● Implementation of advanced security features (20%). ● Performance optimization results (20%). ● Quality of RESTful API design (20%). ● Effectiveness of testing and debugging (20%). ● Documentation quality (10%). ● Overall project presentation and communication (10%).
Teaching and learning method(s):	Face to face 56 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2 and 3.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes and 4, 5, 6 and 7.**

Assessment criteria(s) :	During the midterm exam, students should be able to define and explain key Spring Boot concepts, explain RESTful API principles, and describe Spring Boot's database integration. By the end of the final exam, students should be able to demonstrate proficiency in designing robust RESTful APIs, implementing advanced security measures, integrating with frontend frameworks, designing microservices-based systems, and conducting comprehensive testing and quality assurance in Spring Boot applications.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Advanced Spring Boot Development: Mastery of Spring Boot for complex projects. 2. Effective RESTful API Design: Design and build robust RESTful APIs. 3. Database Integration Expertise: Effective integration of Spring Boot with relational databases. 4. Security Implementation: Implement advanced security measures for Spring Boot applications. 5. Microservices Architecture: Design and develop microservices-based systems. 6. Performance Optimization: Optimize Spring Boot applications for maximum efficiency and scalability.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] "Spring Boot in Action" by Craig Walls. [2] "Spring Microservices in Action" by John Carnell and Ronnie Mitra. [3] "Mastering Spring Boot 2.0" by Greg Turnquist. [4] Spring Boot official documentation and guides (https://spring.io/guides).

Department of Computer Science
C-5.3.2-Microservices Architecture

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Advanced Web**		**Course**	**Microservices Architecture**		**Code**	**C-5.3.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h 00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**5**
	Workshop	**--**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**35h00**					
Total workload		**56h00 / Semester**			**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher		**Mr Wael Sellami**	**Involved Teacher(s)**		**Mr Wael Sellami**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI** **-M.A. PHD. M.A. OMRANI** **-M.A. PHD. WAEL SELLAMI** **- Director of study A. SELMI**	**Objective**		**Obj 1**		
			PLO		**P1 : Systems Specification and Design Methodologies** **P2: Implementation Techniques** **P3: Innovation, Research and Emerging Technologies**		

Prerequisites:	Strong programming skills are a prerequisite for this advanced microservices course, ensuring students can efficiently implement microservices solutions. A foundational understanding of data structures and algorithms is vital for designing efficient microservices that handle data effectively in distributed systems.
Corequisites:	Concurrent enrollment in Distributed Systems is recommended to explore the intricacies of distributed computing alongside microservices design. Taking a Software Architecture course in tandem ensures students grasp architectural patterns that contribute to robust microservices solutions.
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. By course completion, students will possess the skills to design, develop, and deploy microservices architectures for scalable and efficient distributed systems. 2. They will be able to make informed architectural decisions, implement security measures, and handle data in a distributed, microservices-based ecosystem. 3. The course ensures students grasp industry best practices, fostering a deep understanding of real-world microservices applications.

	4. Ultimately, graduates will emerge as proficient microservices engineers capable of contributing to cutting-edge software solutions in the industry.
Content:	**Chapter 1: Introduction to Microservices** 1. Introduction 2. Fundamentals of Microservices 3. Principles of Microservices 4. Microservices architecture **Practical Work:** TP1: Build a simple microservices-based application and assess its benefits and challenges. **Chapter 2: Design and Development of Microservices** 1. Microservices Design Principles 2. Implementing Microservices 4. Testing strategies for microservices 5. Continuous integration and continuous deployment (CI/CD) **Practical Work:** TP2: Design and implement a microservices architecture with CI/CD for a real-world problem. **Chapter 3: Microservices Communication and Orchestration** 1. Communication Protocols 2. Containerization and Orchestration 3. Service discovery and load balancing **Practical Work:** TP3: : Containerize and orchestrate a microservices app with Docker and Kubernetes, including service discovery and load balancing. **Chapter 4: Data Management in Microservices** 1. Data Storage in Microservices 2. Event-Driven Microservices 3. Implementing microservices using Apache Kafka **Practical Work:** TP4: Develop a microservices system using event sourcing and CQRS with Apache Kafka integration. **Chapter 5: Advanced Topics and Case Studies** 1. Microservices Security and Scaling 2. Monitoring and Troubleshooting 3. Scalability and performance optimization **Practical Work:** TP5: Strengthen microservices security, optimize performance, and create a monitoring and troubleshooting plan. **Project :** Project Title: Microservices-Based E-commerce Platform Project Description: In this comprehensive project, students will apply the knowledge and skills acquired throughout the course to design, develop, and deploy a microservices-based e-commerce platform. The platform will include

	various services responsible for user management, product catalog, shopping cart, order processing, and payment **Self-study:** 1. Research Study programming fundamentals and its relevance in engineering. Collect relevant resources and references. 2. Proposal Choose an engineering problem. Explain how to decompose a complex web application into microservices 4. Project Report Create a report summarizing the project. Reflect on the experience and insights gained. 5. Presentation (Optional) Share project findings with the class if desired. 6. Doing Exercise and QCM published in easylearn.ovh platform.
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1 and 2** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 3 And 4**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain microservices architecture, including its principles, communication methods, containerization, orchestration, and data management, demonstrating a comprehensive understanding of foundational microservices concepts. By the end of the final exam, students should be able to design, implement, and optimize a complete microservices-based application, demonstrating proficiency in microservices architecture, development, communication, data management, security, scalability, and troubleshooting, along with a critical understanding of industry best practices and real-world case studies.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.

Target skill(s):	1. Design modular microservices architectures for scalability. 2. Implement microservices using various programming languages. 3. Containerized microservices with Docker for efficient deployment. 4. Orchestrate microservices using Kubernetes for management. 5. Analyze real-world case studies for practical insights.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] Fowler, M. (2014). "Microservices: a definition of this new architectural term." [2] Newman, S. (2015). "Building Microservices: Designing Fine-Grained Systems." O'Reilly Media. [3] Wiggins, R., & Jones, N. (2018). "Mastering Microservices." Packt Publishing. [4] Tilkov, S., & Vinoski, S. (2015). "Microservices: From Design to Deployment." O'Reilly Media.

Department of Computer Science
C-5.3.3-Tests and Quality

Last Update Date: February 17, 2024

Syllabus Sheet

Unit	**Advanced Web**		**Course**	**Tests and Quality**		**Code**	**C-5.3.3**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**10,5h**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**5**
	Workshop	**21h 00**					
	Project	**10,5h**		**Mixed**	**XX**		
	Self Study	**14h00**					
Total workload	**56h00 / Semester**				**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher	**Prof. PHD. MOUNIR ZRIGUI**		**Involved Teacher(s)**	**Prof. PHD. MOUNIR ZRIGUI**			
Review committee	**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1**			
			PLO	**P2: Implementation Techniques P5: Network and Information Systems Management P3: Innovation, Research and Emerging Technologies**			

Prerequisites:	Basic programming knowledge Understanding of software development life cycle Basics of software engineering
Corequisites:	Software Engineering principles Database Management Systems
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Understand and differentiate between various testing methodologies and techniques. 2. Design and execute test cases and test scripts for software applications. 3. Utilize industry-leading tools for software testing and quality assurance. 4. Evaluate the quality of software through systematic testing processes. 5. Implement continuous integration and continuous testing methods. 6. Understand and apply principles of software quality metrics and measurements. 7. Manage and operate defect tracking systems. 8. Recognize and implement security testing techniques. 9. Understand performance, load, and stress testing strategies. 10. Analyze software artifacts and generate testing documentation.

Content:	**Chapter 1:** Introduction to Software Testing 1. Basics of software testing 2. Importance of quality assurance 3. Types of software defects **Chapter 2: Testing Methodologies** 1. White-box testing vs. black-box testing 2. Unit testing, Integration testing, System testing 3. Regression testing, UAT (User Acceptance Testing) **Chapter 3: Test Case Design** 1. Requirements traceability matrix 2. Test case design techniques 3. Writing effective test cases **Chapter 4: Automation and Tools** 1. Introduction to automation testing 2. Tools: Selenium, JUnit, TestNG, and others 3. Scripting and maintenance **Chapter 5: Continuous Integration and Testing** 1. Principles of CI/CD 2. Jenkins, CircleCI, and Travis CI 3. Automated build testing **Chapter 6: Quality Metrics and Measurements** 1. Introduction to software metrics 2. Collecting and interpreting metrics 3. Quality benchmarks **Chapter 7: Defect Tracking and Management** 1. Bug life cycle 2. Tools: Jira, Bugzilla, and Mantis 3. Reporting and analytics **Chapter 8: Security and Testing** 1. Common software vulnerabilities 2. Penetration testing basics 3. Tools and techniques **Chapter 9: Advanced Testing Techniques** 1. Performance testing 2. Load and stress testing 3. Tools: Apache JMeter, LoadRunner **Chapter 10: Documentation and Reporting**

1. Test plans and strategy
2. Test summary reports
3. QA documentation

List of Workshops:

1. Workshop 1: Writing and Executing Test Cases
2. Workshop 2: Automation Scripting with Selenium
3. Workshop 3: Implementing Continuous Testing with Jenkins
4. Workshop 4: Metrics Collection and Analysis
5. Workshop 5: Bug Reporting with Jira
6. Workshop 6: Basic Penetration Testing for Web Applications
7. Workshop 7: Performance Testing with Apache JMeter
8. Workshop 8: Documentation and Reporting Best Practices

Self-Study Project: Comprehensive Software Testing Strategy and Implementation

Topic (Goal): The primary goal of this project is to allow students to demonstrate their understanding of various software testing methodologies, tools, and techniques in a practical scenario. By selecting a software application of their choice, students will research, design, and implement a holistic testing strategy, incorporating multiple testing methodologies to ensure the quality and reliability of the software.
Description:
Software Selection:
Choose a software application. This can be a web application, mobile application, or desktop software. Ideally, this should be open-source or a project students have previously worked on, allowing complete access to the codebase.
Requirement Analysis:
Gather and document the functional and non-functional requirements of the chosen software. This will form the basis for your testing strategy.
Test Plan Development:
Create a comprehensive test plan detailing the scope, approach, resources, and schedule of the intended testing activities. It should specify the items to be tested, the testing tasks, and the personnel responsible for each task.
Testing Methodologies:
Implement at least three different testing methodologies:
Unit Testing: Test individual units or components of the software.
Integration Testing: Test the interfaces between components.
System Testing: Test the complete integrated system.
Automation:
Incorporate automated testing wherever feasible. Use tools like Selenium for web apps or JUnit for Java applications. Document the scripts and the outcomes.
Defect Management:
Identify and document defects. Use tools like Jira or Bugzilla to track and manage these defects. Each defect should include steps to reproduce, severity, and potential impact.
Performance and Security:
Conduct basic performance testing (e.g., using Apache JMeter) to check the application's responsiveness and stability.
Implement basic security testing to uncover potential vulnerabilities. This could involve penetration testing or vulnerability scanning, depending on the nature of the application.

	Documentation: Compile your findings, challenges faced, and the quality improvements noticed after testing. Detail the tools, techniques, and methodologies used. Highlight the value and importance of each testing method in your overall strategy. **Reflection**: Reflect on the entire testing process. What went well? What challenges did you face, and how did you overcome them? How would you improve your strategy if you had to redo the project? **Submission Guidelines:** The project should be submitted as a comprehensive report, accompanied by any scripts, code snippets, or testing tools used. The report should be structured, including a table of contents, introduction, main content, conclusion, and references. Relevant screenshots, diagrams, and flowcharts can be included to enhance the clarity and presentation of the report. All tools, third-party software, or resources used should be appropriately cited.
Teaching and learning method(s):	Face to face 42 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2, 3, 4. and 5** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 6, 7, 8, 9 and 10.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain various testing methodologies, design relevant test cases, and utilize basic testing tools. By the end of the final exam, students should be able to manage comprehensive testing processes, understand advanced testing techniques, and generate effective software testing documentation.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Proficiency in test case design and execution. 2. Mastery in software testing tools and automation. 3. Deep understanding of software quality metrics. 4. Ability to manage and maintain continuous testing environments. 5. Skill in defect management and tracking. 6. Competence in performance and security testing techniques.

Teaching Manual:	• Powerpoint presentation • Tutorial booklet • Workbooks
References:	[1] "Software Testing: A Craftsman's Approach" by Paul C. Jorgensen [2] "The Art of Software Testing" by Glenford J. Myers, Corey Sandler, Tom Badgett [3] "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by Jez Humble, David Farley.

Department of Computer Science

C-5.4.1 Business creation and innovation

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Transversal Unit 5**		**Course**	**Business creation and innovation**		**Code**	**C-5.4.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**5**
	Workshop	**-**					
	Project	**-**		**Mixed**	**XX**		
	Self Study	**34h00**					
Total workload		**55h00 / Semester**			**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher		**Mr Safa Ben Salem**		**Involved Teacher(s)**	**Mr Safa Ben Salem**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1, Obj2 and Obj 3**		
				PLO	**P3: Innovation, Research and Emerging Technologies P7: Analytical and Critical Thinking P8: Communication and Collaboration**		

Prerequisites:	**Foundations in Computer Science and Electrical Engineering:** Students are expected to have a solid understanding of core concepts in computer science and electrical engineering at the undergraduate level. **Business Fundamentals:** A foundational knowledge of business concepts, which might be obtained through previous coursework or experience. **Statistics:** An understanding of statistical analysis and data interpretation, which is often needed for market research and data-driven decision-making. **Programming Skills:** Depending on the course content, proficiency in programming languages relevant to the field may be required. **Mathematics:** A strong background in mathematics, particularly in areas like calculus and linear algebra, may be necessary for certain topics within the course.
Corequisites:	None: In some cases, a course may not have corequisites, meaning that students can take it independently without needing to enroll in any specific accompanying courses.

Learning outcomes	Upon completion of this teaching course, the student will be able to : 1. Demonstrate an understanding of the fundamental concepts and principles of entrepreneurship and innovation in the context of computer science and electrical engineering. 2. Explain the key components of a successful business model, including value proposition, customer segments, revenue streams, and cost structure. 3. Explain the role of intellectual property protection, including patents and trademarks, in innovation and entrepreneurship. 4. Apply critical thinking and problem-solving skills to evaluate the feasibility of a business idea and make informed decisions. 5. Develop a comprehensive business plan, including financial projections, marketing strategies, and operational plans. 6. Utilize techniques such as lean startup methodologies and agile development for effective product development and iteration. 7. Analyze case studies and real-world examples to extract lessons and best practices for entrepreneurial success. 8. Integrate knowledge from computer science and electrical engineering with business concepts to identify innovative solutions and market opportunities.
Content:	**Chapter 1: Introduction to Entrepreneurship and Innovation** 1. Definitions and concepts 2. Entrepreneurship ecosystems 3. Innovation and its role in business 4. Identifying market opportunities 5. Feasibility analysis 6. Market research and customer discovery **Chapter 2: Business Models and Planning** 1. Business model canvas 2. Lean startup methodology 3. Developing a business plan **Chapter 3: Intellectual Property and Legal Aspects** 1. Patents, trademarks, and copyrights 2. Licensing and protecting intellectual property 3. Legal considerations for startups **Chapter 4 : Funding and Financing** 1. Sources of funding (angel investors, venture capital, etc.) 2. Bootstrapping 3. Pitching to investors **Chapter 5: Product Development and Prototyping** 1. Business Innovation and Technological Trends : Emerging technologies and industry-specific innovations 2. Ethical considerations in entrepreneurship and social and environmental responsibility

	3. Agile development 4. Minimum Viable Product (MVP) 1. Prototyping tools and techniques **Chapter 6 : Marketing and Sales Strategies** 1. Marketing fundamentals 2. Sales techniques 3. Branding and promotion **Practical Work:** - Students will apply their communication skills in a real-world scenario, such as a group presentation or a technical report. - Peer and instructor evaluations. - Course reflection and feedback. - Final Projects and Presentations. - Students work on business creation or innovation projects. - Final project presentations. **Self-study:** Students will focus on : - Case Studies and Guest Lectures: Real-world examples of successful startups / Guest speakers from the industry.
Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) is planned to ensure the evaluation of the **learning outcomes** achievement level. A practical project / portfolio presentation to evaluate: - Class Participation and Engagement - Individual and Group Presentations - Written Assignments and Reports
Assessment criteria(s) :	**Business plan :** - Clarity and Completeness: Assess the clarity and completeness of the business plan, ensuring that it effectively communicates the business idea and strategy. - Feasibility Analysis: Evaluate the thoroughness of the feasibility analysis, including market research, financial projections, and risk assessment. - Innovation: Consider the level of innovation in the business idea and whether it addresses a real market need. **Alignment with Learning Outcomes:** Assess how well the business plan aligns with the course's learning outcomes related to business planning and innovation. **Presentation and Pitching Skills:** Communication, Clarity: and Engagement **Practical Projects and Prototyping:** Quality of Prototype, Innovation and Problem-Solving Ethical Considerations: Ethical Awareness and Social and Environmental Responsibility.

	Written Assignments: - Clarity and Structure: Assess the clarity, structure, and organization of written assignments related to the course. - Research and Analysis: Evaluate the quality of research and analysis in written assignments, especially when dealing with business models, market research, and industry trends.
Assessment measure(s):	Assigning of a final average: Assignments, Quizzes, Exams, Assignments, Quizzes, Exams.
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam : 60%
Target skill(s):	This course aims to equip students with the knowledge and skills required to understand and engage in business creation and innovation in the context of computer science and electrical engineering. It covers theoretical foundations, practical tools, and real-world case studies to foster an entrepreneurial mindset.
Teaching Manual:	● PowerPoint presentation ● Lecturing ● Tutorials ● Courses project Lecturing
References:	[1] "The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses" by Eric Ries - This book is a staple for understanding the lean startup methodology and its application to innovative technology ventures. [2] "Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers" by Alexander Osterwalder and Yves Pigneur - It provides insights into business model canvas and helps in creating, analyzing, and innovating business models. [3] "The Innovator's Solution: Creating and Sustaining Successful Growth" by Clayton Christensen and Michael Raynor - A companion to "The Innovator's Dilemma," it focuses on practical solutions for innovation and growth. [4] "Disciplined Entrepreneurship: 24 Steps to a Successful Startup" by Bill Aulet - A systematic guide to entrepreneurship, providing a step-by-step approach to building a startup.

Department of Computer Science

C-5.4.2-Technical English 2 (Scientific Writing)

Last Update Date: February 17, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Transversal Unit 5</td><td>Course</td><td colspan="2">Technical English 2 (Scientific Writing)</td><td>Code</td><td>C-5.4.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">2</td></tr>
<tr><td>Workshop</td><td>-</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>-</td></tr>
<tr><td>Self Study</td><td>35h00</td><td colspan="2"></td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mme Besma Ben Ismail</td><td>Involved Teacher(s)</td><td colspan="3">Mme Besma Ben Ismail</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 2 & Obj 3</td></tr>
<tr><td>PLO</td><td colspan="3">P8 : Communication and Collaboration
P9: Lifelong Learning and Adaptability</td></tr>
<tr><td><u>Prerequisites:</u></td><td colspan="7">English Proficiency: Students should have a demonstrated proficiency in English, either through standardized language exams (e.g., TOEFL, IELTS) or through an English language proficiency test administered by the institution. The minimum required English proficiency level should be defined based on the institution's policies.</td></tr>
<tr><td><u>Corequisites:</u></td><td colspan="7">Introduction to Computer Science and Engineering: It's recommended that students have completed or are concurrently enrolled in an introductory course in computer science and engineering. This corequisite ensures that students have a basic understanding of the subject matter they will be writing about and enables them to apply their technical knowledge to their writing assignments.
Academic Writing Skills: A corequisite course in academic writing, which focuses on general writing skills, can be beneficial. It provides students with foundational writing skills that are applicable across different academic contexts.
Technical Communication or Intro to Scientific Writing: Depending on the institution's offerings, a corequisite course in technical communication or an introductory scientific writing course can be beneficial. This will provide students with an understanding of the fundamentals of technical writing, which they can build upon in the more specialized Technical English - Scientific Writing course.</td></tr>
</table>

Learning outcomes	Upon completion of this teaching course, the student will be able to: **1.** Introduce themselves both formally and informally to others. **2.** Understand different types of written texts and summarize them. **3.** Understand graphics, visuals, tables, and diagrams and analyze them orally or transfer them into a written text. **4.** Speak fluently and confidently.
Content:	**Chapter 1: Introduction to technical writing** 1. Understanding the importance of effective communication in computer science and engineering. 2. Overview of scientific writing conventions and standards. 3. Setting up the course expectations and assessment criteria. **Chapter 2: Language skills development** 1. Grammar and syntax for technical writing. 2. Vocabulary building and technical terminology. 3. Writing exercises for clarity and coherence. **Chapter 3: Research and review papers** 1. Analyzing and evaluating research papers in computer science and engineering. 2. Understanding the structure of scientific papers. 3. Citation and referencing techniques. **Chapter 4: Writing techniques** 1. Organizing information logically in technical documents. 2. Creating effective abstracts and summaries. 3. Presenting data and results using graphs and tables. 4. Editing and proofreading techniques. 5. Document formatting, including figures, tables, and references. 6. Ethical considerations in scientific writing. **Practical Work:** 1. Student presentations on research papers 2. Course reflection and feedback 3. Research paper structure and organization 4. Conducting literature review 5. Preparing and delivering technical presentations. 6. Peer review of final research papers. **Self-study:** Students will: - Develop a structured plan that outlines what they will study and when. - Periodically assess your progress, identify areas where they need improvement, and revise their study plan accordingly. - Choose textbooks and materials designed for advanced technical English. - Work on writing technical reports, research papers, and essays related to their engineering field. Review and edit their work for clarity and coherence. - Improve their oral communication skills by engaging in discussions and presentations related to their engineering field. Record and review their presentations for self-assessment.

Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work, activities, T. B. L and P.B.L.
Assessment method(s):	A midterm written test (D.S) is planned to ensure the evaluation of the **learning outcomes** achievement level. A practical project / portfolio presentation to evaluate: - Class Participation and Engagement - Individual and Group Presentations - Written Assignments and Reports
Assessment criteria(s) :	**Knowledge and understanding:** Students should be able to introduce themselves both formally and informally to others, understand different types of written texts and summarize them and understand graphics, visuals, tables, and diagrams and analyze them orally or transfer them into a written text. **Skills :** - Speaking fluently and confidently. - Acquiring new vocabularies related to the main topics. - Writing accurately and coherently.
Assessment measure(s):	Assigning of a final average: Placement test, Indirect assessment, Exercises, Peer to peer assessment, Recording a conversation, Weekly assignments, and Quizzes.
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final examination assessing writing skills and comprehension: 60%.
Target skill(s):	1. Develop language skills necessary for effective scientific communication. 2. Understand and apply scientific writing conventions, including structure, clarity, and conciseness. 3. Analyze and critically evaluate technical literature in the field of computer science and engineering. 4. Produce well-structured and coherent technical documents, including research reports, papers, and project proposals. 5. Use appropriate referencing and citation techniques in accordance with academic and professional standards.
Teaching Manual:	● PowerPoint presentation ● Lecturing ● Tutorials ● Courses project Lecturing
References:	[1]:"Scientific Writing and Communication: Papers, Proposals, and Presentations" by Angelika H. Hofmann. [2]: "The Elements of Style" by William Strunk Jr. and E.B. White [3]: "Technical English: Writing, Reading and Speaking" by Nell Gluckman and Murat Yasar [4]: Supplementary materials and research papers from the field of computer science and engineering.

Department of Computer Science
C-5.5.1-Elective Course 1:AZ900

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Elective Unit 2**		**Course**	**Elective Course 1: AZ900**		**Code**	**C-5.5.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Common**
Type of education	**Lecture**	**21h**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**5**
	Workshop	**--**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**35h00**					
Total workload		**56h00 / Semester**			**Coefficient**		**2**
					ECTS credit		**2**
Responsible Teacher		**Mr Nafaa Haffar**	**Involved Teacher(s)**	**Mr Nafaa Haffar**			
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**	**Objective**	**Obj 1 and Obj 3**			
			PLO	**P3: Innovation, Research and Emerging Technologies P9: Lifelong Learning and Adaptability**			

Prerequisites:	• Familiarity with fundamental IT concepts such as networking, storage, and computing is essential for grasping cloud computing principles effectively. • Proficiency in operating systems concepts, including installation, configuration, and management, provides a solid foundation for learning cloud technologies.
Corequisites:	having some familiarity with the Azure platform can facilitate understanding of the course content and enhance learning outcomes.
Learning outcomes	Upon completion of this teaching module, the student will be able to: 1. Gain a comprehensive understanding of cloud computing principles, including the benefits, types of cloud services, and deployment models, to effectively leverage cloud technologies. 2. Develop the ability to describe essential Azure services, such as computing, storage, and networking, and their roles in building and deploying cloud solutions, enabling informed decision-making in Azure deployments. 3. Acquire knowledge of Azure pricing models, cost management tools, and support options to effectively manage costs and optimize spending on Azure services, ensuring efficient resource utilization.

	4. Explore key principles of cloud security, privacy, compliance, and trust, and understand how Azure implements security measures to protect data and ensure compliance with regulatory requirements, fostering trust in cloud deployments.
Content:	**Chapter 1: Azure Governance and Compliance** 1. IaaS 2. PaaS 3. SaaS 4. Public Cloud 5. Private Cloud 6. Hybrid Cloud **Chapter 2: Azure Services** 1. compute 2. storage 3. networking 4. Databases **Chapter 3: Azure Pricing and SLA** 1. Azure pricing models 2. service-level agreements (SLAs) 3. cost management **Chapter 4: Azure Governance and Compliance** 1. Azure governance methodologies 2. management groups 3. Subscriptions 4. compliance standards **Chapter 5: Azure Identity and Access Management (IAM)** 1. Azure identity services 2. Azure Active Directory (Azure AD) 3. role-based access control (RBAC)
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1 and 2.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 3**

Assessment criteria(s) :	During the midterm exam, students should be able to define and explain Azure Cloud, Azure Governance and Compliance and Azure Services By the end of the final exam, students should be able to review Microsoft's official AZ-900 exam skills.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Microsoft's official AZ-900 exam skills 2. Develop the ability to describe essential Azure services such as computing, storage, and networking, and their role in building and deploying cloud solutions, enabling informed decision-making in Azure deployments. 3. Acquire knowledge of Azure pricing models, cost management tools, and support options to effectively manage costs and optimize spending on Azure services, ensuring efficient resource utilization. 4. Understanding Cloud Security and Compliance: Explore key principles of cloud security, privacy, compliance, and trust, and understand how Azure implements security measures to protect data and ensure compliance with regulatory requirements, fostering trust in cloud deployments.
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] https://learn.microsoft.com/en-us/credentials/certifications/exams/az-900/ [2] Zacker, C. (2020). Exam Ref AZ-900 Microsoft Azure Fundamentals (2nd ed.). Microsoft Press. [3] Pluralsight. (n.d.). Azure courses. https://www.pluralsight.com/browse/cloud-computing/azure [4] Sharma, S. (2019). AZ-900 Azure Exam Preparation Guide: Pass Azure Fundamentals (1st ed.).

Department of Computer Science

C-5.5.1-Elective Course 1 : Preparation for AI Certification (HCIA-AI)

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Elective Unit 2</td><td>Course</td><td colspan="2">Elective Course 1 Preparation for AI Certification (HCIA-AI)</td><td>Code</td><td>C-5.5.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">5</td></tr>
<tr><td>Workshop</td><td>--</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>35h00</td><td colspan="2"></td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Wael Sellami</td><td>Involved Teacher(s)</td><td colspan="3">Mr Hatem Jarboua</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1 and Obj 3</td></tr>
<tr><td>PLO</td><td colspan="3">P3: Innovation, Research and Emerging Technologies
P9: Lifelong Learning and Adaptability</td></tr>
<tr><td colspan="2"><u>Prerequisites:</u></td><td colspan="6">• Familiarity with programming concepts such as variables, loops, and functions is recommended to understand the implementation of AI algorithms.
• Basic knowledge of statistics, including concepts such as mean, median, and standard deviation, is beneficial for comprehending machine learning algorithms.</td></tr>
<tr><td colspan="2"><u>Corequisites:</u></td><td colspan="6">Practical experience with data analysis and programming in Python is advantageous for grasping AI concepts effectively.</td></tr>
<tr><td colspan="2"><u>Learning outcomes</u></td><td colspan="6">Upon completion of this teaching module, the student will be able to:
1. Gain a comprehensive understanding of fundamental concepts in artificial intelligence, including machine learning, deep learning, and neural networks.
2. Develop the ability to implement various machine learning algorithms, such as linear regression, logistic regression, and decision trees, for solving real-world problems.
3. Learn about deep learning techniques, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), and understand their applications in computer vision, natural language processing, and speech recognition.</td></tr>
</table>

	4. Acquire hands-on experience in applying AI techniques to practical scenarios, such as image classification, sentiment analysis, and predictive analytics, through project-based learning.
Content:	**Content:** **Chapter 1**: Introduction to Artificial Intelligence ● Overview of artificial intelligence, its history, and applications. ● Explanation of basic AI concepts, including machine learning, deep learning, and neural networks. **Chapter 2**: Machine Learning Algorithms ● Detailed exploration of various machine learning algorithms, such as linear regression, logistic regression, decision trees, and k-nearest neighbors. ● Understanding of the theoretical foundations and practical applications of each algorithm. **Chapter 3:** Deep Learning Techniques ● Introduction to deep learning, including neural networks, activation functions, and optimization algorithms. ● Deep dive into convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for image classification and sequence modeling tasks. **Chapter 4**: AI Applications and Future Trends ● Examination of AI applications in computer vision, natural language processing, and robotics. ● Discussion of emerging trends in AI research and potential future developments. **Tutorials:** **Tutorial 1**: Introduction to Python for AI ● Basics of Python programming language for AI applications, including data manipulation, visualization, and libraries such as NumPy and Pandas. **Tutorial 2**: Implementing Machine Learning Algorithms ● Step-by-step tutorials on implementing machine learning algorithms in Python using libraries like scikit-learn, including model training, evaluation, and hyperparameter tuning. **Tutorial 3**: Deep Learning with TensorFlow ● Hands-on tutorials on building and training deep learning models using TensorFlow, covering topics such as building CNNs for image classification and RNNs for sequence modeling. **Tutorial 4**: AI Project Development ● Guided project tutorials where students apply AI techniques to real-world datasets, solving tasks such as image classification, sentiment analysis, and predictive modeling. **Practical Work:** **Practical Work 1**: Data Preparation and Preprocessing ● Hands-on exercises on data cleaning, preprocessing, and feature engineering for AI applications, using libraries like Pandas and scikit-learn.

	Practical Work 2: Model Training and Evaluation ● Practical sessions on model training and evaluation, including techniques for cross-validation, model selection, and performance metrics interpretation. **Practical Work 3**: Deep Learning Model Development ● Project-based practical work on developing deep learning models for computer vision and natural language processing tasks, including data augmentation and transfer learning techniques. **Practical Work 4**: AI Project Implementation ● Guided project work where students implement end-to-end AI solutions, from data preprocessing to model deployment, focusing on real-world applications and industry-relevant projects.
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1 and 2.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 3**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain Azure Cloud, Azure Governance and Compliance and Azure Services By the end of the final exam, students should be able to review Microsoft's official AZ-900 exam skills.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	**1.** Learn to analyze datasets using statistical and exploratory data analysis techniques to identify patterns and insights crucial for AI model development. **2.** Develop the ability to implement various machine learning algorithms such as linear regression, decision trees, and support vector machines to solve classification and regression problems. **3.** Acquire skills in building and training neural networks, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for tasks like image classification and sequence prediction.

	4. Learn to evaluate the performance of AI models using appropriate metrics and techniques such as cross-validation, confusion matrices, and ROC curves to ensure optimal model performance and generalization.
Teaching Manual:	• Powerpoint presentation • Tutorial booklet • Workbooks
References:	[1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. [2] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer. [3] Raschka, S., & Mirjalili, V. (2019). Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow (3rd ed.). Packt Publishing. [4] Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd ed.). O'Reilly Media.

Department of Computer Science

C-5.5.2-Elective Course 2: blockchain

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Elective Unit 2</td><td>Course</td><td colspan="2">Elective Course 2: blockchain</td><td>Code</td><td>C-5.5.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">5</td></tr>
<tr><td>Workshop</td><td>--</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>--</td><td rowspan="2">Coefficient</td><td rowspan="2">2</td></tr>
<tr><td>Self Study</td><td>35h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Soufien GDAIM</td><td>Involved Teacher(s)</td><td colspan="2">Mr Soufien GDAIM</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1 and Obj 3</td></tr>
<tr><td>PLO</td><td colspan="2">P3: Innovation, Research and Emerging Technologies
P9: Lifelong Learning and Adaptability</td></tr>
<tr><td>Prerequisites:</td><td colspan="7">● Fundamental understanding of data structures and algorithms.
● Basic knowledge of information systems security.
● Familiarity with distributed systems and networking.</td></tr>
<tr><td>Corequisites:</td><td colspan="7">None</td></tr>
<tr><td>Learning outcomes</td><td colspan="7">Upon completion of this teaching module, the student will be able to:

1. Understand the fundamental concepts of blockchain technology, including decentralized ledger, cryptographic hashing, and consensus mechanisms.
2. Analyze the benefits and challenges of blockchain technology in various applications.
3. Design and implement basic blockchain applications using smart contracts and decentralized applications (DApps).
4. Evaluate security considerations and best practices in blockchain development.
5. Critically assess real-world blockchain use cases and their impact on different industries.</td></tr>
<tr><td>Content:</td><td colspan="7">Chapter 1: Introduction to Blockchain
1. What is Blockchain?
2. History and Evolution of Blockchain Technology
3. Key Terminologies: Blocks, Decentralization, Consensus Mechanisms
4. Use Cases and Importance of Blockchain</td></tr>
</table>

Chapter 2: Blockchain Architecture

1. Blockchain Components: Nodes, Transactions, Blocks
2. Public vs. Private Blockchains
3. Blockchain Protocols: Bitcoin, Ethereum, Hyperledger
4. How Mining Works

Chapter 3: Cryptography in Blockchain

1. Role of Cryptography in Blockchain
2. Public and Private Keys
3. Digital Signatures
4. Hash Functions

Chapter 4: Consensus Mechanisms

1. Proof of Work (PoW)
2. Proof of Stake (PoS)
3. Delegated Proof of Stake (DPoS)
4. Practical Byzantine Fault Tolerance (PBFT)

Chapter 5: Smart Contracts

1. Understanding Smart Contracts
2. Introduction to Solidity (Ethereum)
3. Writing and Deploying a Simple Smart Contract
4. Use Cases for Smart Contracts

Chapter 6: Blockchain Challenges and Limitations

1. Scalability and Performance Issues
2. Security Concerns
3. Regulatory and Legal Challenges
4. Environmental Impact

Chapter 7: Real-world Applications

1. Blockchain in Finance and Banking
2. Supply Chain Management
3. Healthcare and Pharmaceuticals
4. Voting and Governance Systems
5. Cross-Industry Use Cases

Tutorials

Tutorial 1: Setting up a Private Blockchain Network

- Guide students through the process of setting up a private blockchain network using platforms like Ethereum or Hyperledger Fabric.
- Provide step-by-step instructions for deploying nodes, creating transactions, and forming blocks within the network.

	Tutorial 2: Smart Contract Development with Solidity ● Introduce students to Solidity, the programming language for writing smart contracts on the Ethereum blockchain. ● Provide hands-on exercises for writing and deploying simple smart contracts, such as token contracts or basic voting contracts. **Tutorial 3**: Cryptography in Blockchain ● Hands-on practice with cryptographic techniques used in blockchain, such as generating public and private keys, creating digital signatures, and implementing hash functions. ● Demonstrate how cryptographic operations are used in blockchain transactions and block validation. **Tutorial 4**: Consensus Mechanisms Simulation ● Simulate various consensus mechanisms such as Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT) using simple scenarios. ● Guide students in implementing consensus algorithms using pseudocode or simple scripts.
Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, and 2** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 3, 4, and 5**
Assessment criteria(s) :	During the midterm exam, students should be able to understand blockchain concepts, analyze and compare consensus mechanisms. Students should be able to demonstrate proficiency in blockchain cryptography and display knowledge of smart contracts. By the end of the final exam, students should be to provide real-world examples of blockchain applications and to demonstrate proficiency in designing and implementing smart contracts. Students should be able also to apply critical thinking to evaluate blockchain challenges, and integrate knowledge to propose solutions for real-world problems.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20%

	• Final written exam: 60%.
Target skill(s):	1. Blockchain technology understanding and application 2. Cryptography and security knowledge 3. Smart contract development skills 4. Critical thinking and problem-solving
Teaching Manual:	• Powerpoint presentation • Tutorial booklet • Workbooks
References:	[1] D. Drescher, Blockchain Basics: A Practical Approach, 2018 [2] P. Vigna, The Truth Machine: The Blockchain and the Future of Everything, 2018 [3] S. Vyasl, Blockchain Technology Exploring Opportunities, Challenges, and Applications, 2022

Department of Computer Science

C-5.5.2-Elective Course 2: Cyber Security

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Elective Unit 2</td><td>Course</td><td>Elective Course 2: Cyber Security</td><td colspan="2">Code</td><td>C-5.5.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Common</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="4">Semester</td><td rowspan="4">5</td></tr>
<tr><td>Workshop</td><td>--</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>35h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">56h00 / Semester</td><td>Coefficient</td><td>2</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>2</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Ilyes Kooli</td><td>Involved Teacher(s)</td><td colspan="2">Mr Ilyes Kooli</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1 and Obj 3</td></tr>
<tr><td>PLO</td><td colspan="2">P3: Innovation, Research and Emerging Technologies
P9: Lifelong Learning and Adaptability</td></tr>
<tr><td>Prerequisites:</td><td colspan="7">Basic knowledge of information systems security.
Basic knowledge of networking and operating systems.
Knowledge of fundamental networking and operating systems.
Familiarity with basic programming and scripting concepts.</td></tr>
<tr><td>Corequisites:</td><td colspan="7">System and Network Administration</td></tr>
<tr><td>Learning outcomes</td><td colspan="7">Upon completion of this teaching module, the student will be able to:

1. Analyze and assess security threats and vulnerabilities.
2. Design and implement effective security measures for networks and systems.
3. Understand the legal and ethical aspects of cybersecurity.
4. Demonstrate proficiency in security tools and techniques.
5. Communicate effectively about cybersecurity issues.</td></tr>
<tr><td>Content:</td><td colspan="7">Chapter 1: Introduction to Cybersecurity
1. What is cybersecurity?
2. The importance of cybersecurity
3. The different types of cybersecurity threats
4. The different types of cybersecurity solutions

Chapter 2: Cybersecurity Risks
1. Risk assessment
2. Risk management</td></tr>
</table>

3. Common cybersecurity risks

Chapter 3: Cybersecurity Solutions

1. Access control
2. Authentication
3. Authorization
4. Encryption
5. Network security

Chapter 4: Cybersecurity Evaluation

1. Security testing
2. Vulnerability assessment
3. Risk assessment

Chapter 5: Cybersecurity Best Practices

1. Password management
2. Patch management
3. Incident response

Chapter 6: Emerging Cybersecurity Trends

1. Artificial intelligence and machine learning
2. Cloud computing
3. Blockchain
4. Internet of Things (IoT)

Tutorials

Tutorial 1: Cybersecurity Threat Simulation

- Set up a simulated cybersecurity environment using virtual machines or cloud platforms.
- Guide students through practical exercises where they simulate and mitigate these threats using security tools and techniques.

Tutorial 2: Access Control Implementation

- Demonstrate the setup of user accounts, role-based access controls (RBAC), and access control lists (ACLs) using common security platforms or operating systems.
- Allow students to practice implementing access control policies and permissions to restrict unauthorized access to resources.

Tutorial 3: Encryption and Network Security Configuration

- Guide them in configuring firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS) to enhance network security.
- Provide practical exercises where students encrypt data transmissions and configure network security measures to protect against unauthorized access and data breaches.

Tutorial 4: Security Testing and Vulnerability Assessment

- Introduce students to security testing methodologies such as penetration testing and vulnerability scanning.
- Set up a lab environment with vulnerable systems or applications for students to perform security tests.

Teaching and learning method(s):	Face to face (face-to-face) 21 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2 and 3** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes ., 4, 5 and 6**
Assessment criteria(s) :	Learning outcomes for midterm exams, students should be able to define and explain the fundamental concepts of cybersecurity, analyze and assess cybersecurity risks, design and implement basic cybersecurity solutions. By the end of the final exam, students should be able to evaluate the effectiveness of cybersecurity solutions, to apply cybersecurity principles and best practices to real-world scenarios. They should also be able to discuss emerging cybersecurity trends
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	**1.** Critical thinking and problem-solving in cybersecurity. **2.** Technical skills in configuring and using security tools. **3.** Ethical decision-making and understanding legal frameworks.
Teaching Manual:	**1.** PowerPoint presentation **2.** Tutorial booklet **3.** Workbooks
References:	[1] A. Calder and S. East, Incident Response: A Methodology for Computer Security Incidents, 2022 [2] J. Erickson, Hacking: The Art of Exploitation, 2022 [3] M. Sikorski and A. Honig, Practical Malware Analysis, 2022 [4] C. Pfleeger and and S. Lawrence, Cybersecurity Essentials, 2023.

Department of Computer Science

NES-C-5.6.1-Next Generation Network (NGN)

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Networks and Embedded System 3</td><td>Course</td><td colspan="2">Next Generation Network (NGN)</td><td>Code</td><td>NES-C-5.6.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Networks and Embedded System</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">5</td></tr>
<tr><td>Workshop</td><td>21h 00</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>--</td><td rowspan="2">Coefficient</td><td rowspan="2">3</td></tr>
<tr><td>Self Study</td><td>42h00</td><td colspan="2">Total workload</td><td colspan="2">84h00 / Semester</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mme Elhem Blel</td><td>Involved Teacher(s)</td><td colspan="2">Mr ElHem Blel</td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1 and Ob j2</td></tr>
<tr><td>PLO</td><td colspan="3">P2: Implementation Techniques
P3: Innovation, Research and Emerging Technologies
P8: Communication and Collaboration</td></tr>
<tr><td colspan="2"><u>Prerequisites:</u></td><td colspan="6">❑ Bachelor's degree in a related field (e.g., computer science, telecommunications).
❑ Basic knowledge of computer networks.</td></tr>
<tr><td colspan="2"><u>Corequisites:</u></td><td colspan="6">❑ Advanced Network Protocols
❑ Network Security and Privacy
❑ Data Center Technologies
❑ Wireless and Mobile Networks</td></tr>
<tr><td colspan="2"><u>Learning outcomes</u></td><td colspan="6">Upon completion of this teaching module, the student will be able to:
1. Analyze and evaluate the architecture and components of Next Generation Networks (NGN).
2. Design, implement, and manage NGN solutions.
3. Understand and apply advanced network protocols and technologies.
4. Evaluate the security and privacy implications of NGN.
5. Develop solutions for real-world NGN challenges.</td></tr>
<tr><td colspan="2"><u>Content:</u></td><td colspan="6">Chapter 1: Introduction to NGN

1. Overview of telecommunication evolution
2. Basic notions of NGN
3. Key drivers and motivations for NGN
4. NGN vs. legacy networks</td></tr>
</table>

	Chapter 2: NGN Architecture 1. Core NGN architecture components 2. IP Multimedia Subsystem (IMS) 3. Convergence of voice, data, and video 4. Quality of Service (QoS) in NGN 5. Security considerations in NGN **Chapter 3: NGN Protocols and Technologies** 1. SIP (Session Initiation Protocol) and its role in NGN 2. MPLS (Multiprotocol Label Switching) in NGN 3. Voice over IP (VoIP) in NGN 4. NGN Transport Networks 5. Software-Defined Networking (SDN) in NGN **Chapter 4: NGN Services and Applications** 1. NGN-based services: VoIP, IPTV, Unified Communications 2. Over-the-Top (OTT) services in NGN 3. Internet of Things (IoT) in NGN 4. Content delivery in NGN **Chapter 5: Emerging Trends in NGN** 1. 5G and NGN convergence 2. Edge computing in NGN 3. Future NGN technologies: Quantum networking, AI, and more 4. Sustainability and energy efficiency in NGN **Chapter 6: Case Studies and Projects** 1. Analyze and discuss real-world NGN implementations and case studies 2. Group projects related to NGN design, deployment, or optimization —---------- **Project :** Students are required to choose a research or project subject related to NGN. For exemple ❑ "Implementing Quality of Service (QoS) in NGN" ❑ "The Impact of 5G on Next Generation Networks" ❑ "NGN Security Best Practices and Case Studies" ❑ "Future Trends in NGN: IoT and Edge Computing Integration" ❑ "Real-time Communication in NGN: VoIP and Video Conferencing" —---------- **Self-study:** Title: "Next-Generation Network Virtualization and Resource Orchestration" Description: For this self-study project, students will explore the cutting-edge field of network virtualization and resource orchestration within Next Generation Networks (NGN). The project will involve in-depth research, analysis, and practical experimentation, aiming to gain a comprehensive understanding of how network virtualization and resource orchestration are transforming modern network infrastructures.
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.

Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2. and 3.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 4 and 5**
Assessment criteria(s) :	Here are some assessment criteria that could be applied: ❑ Depth of Knowledge: Assess students' depth of understanding of NGN concepts and technologies, including advanced topics such as network virtualization, SDN, NFV, and emerging trends. Evaluation should go beyond surface-level knowledge and require in-depth comprehension. ❑ Critical Thinking: Evaluate students' ability to analyze and critically assess NGN-related issues, challenges, and solutions. Encourage them to think critically and consider multiple perspectives when addressing complex network problems. ❑ Research Skills: Assess the quality and rigor of students' research. They should demonstrate the ability to find and use relevant academic sources, conduct experiments or investigations, and draw well-founded conclusions. ❑ Problem-Solving: Evaluate students' problem-solving skills in the context of NGN. Assess their capacity to identify network issues, propose solutions, and evaluate the effectiveness of those solutions. ❑ Practical Skills: For courses with a practical component, assess students' ability to apply NGN concepts by configuring and managing network infrastructures, utilizing virtualization platforms, and troubleshooting network issues.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Network design and analysis 2. Prototyping and implementation 3. Security assessment and risk management 4. Critical thinking and problem-solving 5. Communication and teamwork
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	[1] "Next Generation Networks: Concepts, Challenges, and Solutions" by M. Januszewski [2]"NGN Architectures, Protocols, and Services" by H. Mohamed [3] "Network Security Essentials" by W. Stallings

Department of Computer Science
NES-C-5.6.2-Embedded Systems

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Networks and Embedded System 3**		**Course**	**Embedded Systems**		**Code**	**NES-C-5.6.1**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Networks and Embedded System**
Type of education	**Lecture**	**21h 00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**5**
	Workshop	**21h 00**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**42h00**					
Total workload	**84h00 / Semester**				**Coefficient**		**3**
					ECTS credit		**3**
Responsible Teacher	**Mr. Soufien GDAIM**		**Involved Teacher(s)**	**Mr. Soufien GDAIM**			
Review committee	**-Prof. PHD. MOUNIR ZRIGUI** **-M.A. PHD. M.A. OMRANI** **-M.A. PHD. WAEL SELLAMI** **- Director of study A. SELMI**		**Objective**	**Obj 1 and Ob j2**			
			PLO	**P2: Implementation Techniques** **P3: Innovation, Research and Emerging Technologies** **P8: Communication and Collaboration**			

Prerequisites:	Basic electronic, knowledge algorithm and C programming.
Corequisites:	Microprocessors and Microcontrollers, IoT Protocols and Communication
Learning outcomes	At the end of this teaching module, the student will be able to: 1. Comprehend the evolution, importance, and implications of advanced embedded system architectures. 2. Distinguish between different architectures, their features, and their application areas. 3. Proficiency in High-Level Embedded Design: 4. Utilize modern design techniques, including UML and hardware description languages, to model, visualize, and describe embedded system functionalities. 5. Demonstrate capability in applying best practices in embedded C/C++ programming. 6. Debug embedded software issues effectively and understand the nuances of firmware management. 7. Implement, manage, and troubleshoot advanced communication protocols in embedded contexts. 8. Identify and address security vulnerabilities in embedded systems, applying encryption, secure coding, and hardware-based security measures. 9. Design embedded systems with optimized power consumption using various power management techniques.
Contents :	**Course content :** **Chapiter 1 : Advanced Embedded System Architectures**

1. Evolution of Embedded Systems
2. Multi-core architectures and their implications
3. Hardware/Software co-design

Chapter 2: High-Level Design and Modelling

1. Model-based design for embedded systems
2. UML (Unified Modeling Language) for real-time systems
3. Hardware description languages like VHDL and Verilog

Chapter 3: Embedded Software Development and Debugging

1. Embedded C/C++ programming best practices
2. Debugging techniques for embedded systems
3. Firmware versioning and update mechanisms

Chapter 4: Advanced Communication Protocols and embedded Security

1. In-depth exploration of protocols like I2C, SPI, CAN, and Ethernet in embedded contexts
2. Wireless protocols for embedded systems: Zigbee, LoRa, Bluetooth, and WiFi
3. Security challenges in embedded designs
4. Encryption and secure coding practices
5. Hardware-based security features: TPMs (Trusted Platform Modules), Secure Boot, etc.

Chapter 5: Low-Power Design Techniques

1. Power management techniques in embedded systems
2. Sleep modes,
3. Clock gating
4. Dynamic voltage scaling

Chapter 6: Future Trends in Embedded Electronics and case studies

1. Quantum computing implications for embedded systems
2. Embedded systems in IoT (Internet of Things) and the challenges involved
3. The role of embedded systems in autonomous vehicles and robotics
4. Practical exercises on state-of-the-art platforms and tools
5. Industry-based case studies highlighting challenges and solutions in real-world scenarios

Content of the tutorials

T1 : Create a UML model for a real-time embedded application (e.g., a thermostat).
T2 : Write an Embedded C program to interface with a temperature sensor.
T3 : Interface an embedded system with an IoT platform and push sensor data to the cloud.

Content of the Workshops

Workshop 1: Introduction to Embedded System Development Environment
Objective: Familiarize participants with embedded development tools.
-Setting up the development environment.
-Introduction to common IDEs like Keil, Eclipse for Embedded, etc.
-Flashing a basic program to a microcontroller.

Workshop 2: Embedded Communication Protocols
Objective: Explore communication between embedded devices.
-Set up Bluetooth communication between two microcontrollers esp32.
-Set up Zigbee communication

	Workshop 3: Embedded Security Basics Objective: Understand the fundamentals of embedded system security. -Implementing basic encryption and decryption on transmitted data. -Secure boot implementation on a microcontroller. **Workshop 5:** IoT Integration with Embedded Systems Objective: Connecting embedded systems to the cloud. -Integrating with platforms. -Sending sensor data to the cloud and basic visualization. **Self Study :** - Different self study themes integrate into this subject such as: - Analog interfacing: Sensors, actuators - Basics of timers and counters - Understanding interrupt-driven programming - Uses of timers: PWM, time-keeping, task scheduling - Low power design considerations - Techniques for power-saving: sleep modes, clock scaling - Measuring and monitoring power consumption - Safety-critical embedded systems
Teaching and learning method:	- Face to face - A course space is created on the easylearn.ovh platform to ensure distance learning. This space contains a set of educational resources made available to learners.
Teaching Technique(s):	- Face-to-face lecture and tutorial sessions. - Practical sessions and problem solving via computer - Group discussion and problem solving
Assessment method(s) :	● A mid-term written test (DS) and a final written test (Exam) are planned. ● • The mid-term written test will be devoted to evaluating the level of achievement of learning outcomes 1, 2, 3 and 4. ● • The final written test is devoted to evaluating the level of achievement of learning outcomes5, 6, 7, 8 and 9. ● A practical exam to assess the student's ability to use an embedded C algorithm for embedded board. ● A grade will be distributed for self-study work. This work will be controlled by the teacher.
assessment criteria	During the midterm test, students should be able to master the UML and hardware description languages At the end of the final exam, students must be able to develop an embedded C/C++ program. During a practical exam, students should be able to run an C/C++ embedded program without error by commenting on the different instructions.
Assessment measure	Allocation of a final average Mid-term grade, Workshop grade and exam grade
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The final average is calculated according to the following formula: ● Mid-term written test: 20% ● Practical work score: (handling and practical exam score + self study work score)/2: 20% ● Final written exam: 60%.
Targeted skills :	1. Knowledge of multi-core architectures and its significance in modern design. 2. Ability to model real-time applications using UML. 3. Proficiency in Embedded C/C++ programming. 4. Ability to debug embedded systems using best practices.

	5. Deep understanding of communication protocols used in embedded systems. **6.** Skills in implementing security measures for embedded designs. **7.** Ability to implement power-saving mechanisms in embedded designs. **8.** Knowledge of the challenges and solutions in real-world embedded applications. **9.** Skills in integrating embedded systems with IoT platforms and cloud services. **10.** Familiarity with popular embedded development environments and tools.
Teaching Manual:	● Course support ● Tutorial support ● Workshop booklet
Bibliographic References:	[1] Marwedel, P. (2021). Embedded system design: embedded systems foundations of cyber-physical systems, and the internet of things (p. 433). Springer Nature. [2] Böhm, W., Broy, M., Klein, C., Pohl, K., Rumpe, B., & Schröck, S. (2021). Model-based engineering of collaborative embedded systems: Extensions of the spes methodology (p. 404). Springer Nature. [3] Ünsalan, C., Gürhan, H. D., & Yücel, M. E. (2022). Embedded System Design with ARM Cortex-M Microcontrollers. Springer International Publishing. [4] Moshayedi, A. J., Kolahdooz, A., & Liao, L. (2022). Unity in Embedded System Design and Robotics: A Step-by-step Guide. CRC Press. [5] Morshed, B. I., & Morshed. (2021). Embedded Systems-A Hardware-Software Co-Design Approach. Springer International Publishing. [6] Gliwa, P. (2021). Embedded Software Timing: Methodology, Analysis and Practical Tips with a Focus on Automotive. Springer Nature.

Department of Computer Science
SE-C-5.6.1-ERP/CRM

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Software Engineering 3</td><td>Course</td><td colspan="2">ERP/CRM</td><td>Code</td><td>SE-C-5.6.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="4">Semester</td><td rowspan="4">5</td></tr>
<tr><td>Workshop</td><td>21h 00</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>42h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Hatem Sandid</td><td>Involved Teacher(s)</td><td colspan="3">Hatem Sandid</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1 and Ob j2</td></tr>
<tr><td>PLO</td><td colspan="3">P2: Implementation Techniques
P3: Innovation, Research and Emerging Technologies
P8: Communication and Collaboration</td></tr>
<tr><td>Prerequisites:</td><td colspan="7">Database and Software Engineering</td></tr>
<tr><td>Corequisites:</td><td colspan="7">Decision support systems</td></tr>
<tr><td>Learning outcomes</td><td colspan="7">The objective of this course is to provide the student with knowledge related to the implementation of integrated management software packages, ERP.

The course deals with the roles and issues of integrated systems in a project management context and will allow the student to evaluate the technological needs of a company during the implementation and configuration of integrated systems.

The student should understand what an ERP and/or CRM is, and what issues they address for the company.</td></tr>
<tr><td>Content:</td><td colspan="7">Chapter 1 : Introduction to ERP

Definition and concept of ERP

Evolution and history of ERP systems

Importance and benefits of ERP for organizations

Chapter 2 : ERP Modules</td></tr>
</table>

	Overview of core modules such as Finance, Human Resources, Supply Chain Management, Manufacturing, Sales, and Customer Relationship Management (CRM) Functionality and features of each module Integration between modules and their role in organizational processes **Chapter 3 : ERP Selection and Implementation** Needs assessment and requirements gathering Vendor selection criteria ERP implementation methodologies (e.g., Waterfall, Agile) Change management and organizational readiness Risk management and mitigation strategies **Chapter 4 : ERP Configuration and Customization** Configuration options available within ERP systems Customization vs. configuration Best practices for configuring ERP modules to meet organizational needs **Chapter 5 : ERP Integration and Data Migration** Integration with other enterprise systems (e.g., CRM, SCM) Data migration strategies and challenges Data cleansing and validation techniques **Content of the Practical Work :** **PW 1 :** Installation and configuration of an open source ERP : Odoo and/or Dolibarr **PW 2**: Database initialization (Third party, Products...) and ERP handling **PW 3:** Workflows and business processes **PW 4:** Realization of a commercial workflow (Quotes, orders, invoices...). **PW 5:** Realization of a manufacturing workflow (PO, Bill of Materials...). **PW6**: Discovery of the CRM functionalities **PW 7:** Initialization of accounts, contacts, prospects...
Teaching and learning method(s):	Face to face (face-to-face) 45 hours. A course space is created on the easylearn.ovh **platform** to ensure distance learning during the COVID period. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work

Assessment method(s):	A continuous assessment and a final PW exam are planned.
Assessment criteria(s) :	At the end of the final PW exam, students should be able to create one or more workflows using the chosen ERP.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. Define and choose the right ERP for the company. 2. Installation and configuration of an ERP 3. ERP administration according to the company's workflows and IS
Teaching Manual:	● Powerpoint presentation ● Tutorial booklet ● Workbooks
References:	Mainly training materials and official ERP documentation on the web. (www.odoo.com)

Department of Computer Science
SE-C-5.6.2-DevOps

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Software Engineering 3</td><td>Course</td><td colspan="2">DevOps</td><td>Code</td><td>SE-C-5.6.2</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Software Engineering</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="2">Semester</td><td rowspan="2">5</td></tr>
<tr><td>Workshop</td><td>21h 00</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>42h00</td><td colspan="2"></td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Mr Wael Sellami</td><td>Involved Teacher(s)</td><td colspan="3">Mr Hatem Jarboua</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">Obj 1 and Ob j2</td></tr>
<tr><td>PLO</td><td colspan="3">P2: Implementation Techniques
P3: Innovation, Research and Emerging Technologies
P8: Communication and Collaboration</td></tr>
<tr><td colspan="2">Prerequisites:</td><td colspan="6">1. Understanding of Software Development:
○ Knowledge of programming languages and software development principles.
○ Familiarity with version control systems (e.g., Git).
2. Basic System Administration:
○ Proficiency in using the command line in Linux/Unix environments.
○ Understanding of system and network administration concepts.</td></tr>
<tr><td colspan="2">Corequisites:</td><td colspan="6">Understanding DEV and System and network basics</td></tr>
<tr><td colspan="2">Learning outcomes</td><td colspan="6">Upon completion of this teaching module, the student will be able to:
1. Understanding of devops Principles.
2. Use version control systems like Git to manage source code and collaborate effectively.
3. Set up CI pipelines to automate code integration and testing.
4. Implement CI best practices to ensure code quality.
5. Deploy applications and infrastructure changes automatically using CD pipelines.
6. Understand containerization concepts and technologies like Docker
7. Deploy and manage containerized applications in Kubernetes clusters.
8. Write infrastructure as code using tools like Terraform or Ansible.
9. Set up monitoring and alerting systems to track the health and performance of applications and infrastructure.</td></tr>
</table>

	10. Respond to incidents and troubleshoot issues effectively.
Content:	**Chapter 1 : Introduction to DevOps** 1. Understand the basics of DevOps, its principles, and benefits. 2. Explore the history and evolution of DevOps. 3. Read case studies and real-world examples of DevOps success. **Chapter 2 : Source Code Management** 1. Learn about version control systems (e.g., Git). 2. Explore branching, merging, and best practices in source code management. 3. Set up your Git repository, practice basic commands. **Chapter3 : Continuous Integration (CI)** 1. Study CI concepts and its importance in DevOps. 2. Hands-on experience with popular CI tools like Jenkins, . 3. Create a simple CI/CD pipeline for a sample application. **Chapter 4 : Containerization and Orchestration** 1. Learn about Docker containers and their benefits. 2. Explore container orchestration tools such as Kubernetes. 3. Deploy a containerized application on a Kubernetes cluster. **Week 9-10: Configuration Management** 1. Understand the importance of infrastructure as code (IaC). 2. Learn tools like Ansible management. 3. Write playbooks or manifests for managing server configurations. **Week 11-12: Monitoring and Logging** 1. Study the importance of monitoring and logging in DevOps. 2. Explore tools like Prometheus, Grafana, ELK Stack for monitoring and logging. 3. Set up monitoring and logging for your sample application.

Teaching and learning method(s):	Face to face (face-to-face) 45 hours.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	• A midterm written test (D.S), a practice test and a final written test (Exam) are planned.
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain and apply key DevOps principles like automation, collaboration, and continuous improvement, Proficiency in using version control systems like Git, Setting up CI pipelines that automate code integration and testing have Competence in deploying and managing containers in orchestration platforms like Kubernetes and Ability to respond to and resolve incidents effectively
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: • Mid-term written test (D.S): 20% • Practical work: 20% • Final written exam: 60%.
Target skill(s):	1. To be able to do all the Pipeline Process CI/CD(continuous Integration & continuous deployment)
Teaching Manual:	• Powerpoint presentation • Tutorial booklet • Workbooks • Cloud access • Officials documentations of hashicorp tools
References:	[1] https://www.docker.com/ [2] https://kubernetes.io/ [3] https://docs.ansible.com/ [4] http://git.com/

Department of Computer Science
BDBI-C-5.6.1-Data Mining 2

Last Update Date: May 7, 2024

Syllabus Sheet

<table>
<tr><td>Unit</td><td colspan="2">Big Data and Business Intelligence 3</td><td>Course</td><td colspan="2">Data Mining 2</td><td>Code</td><td>BDBI-C-5.6.1</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science Department</td><td>National Diploma</td><td colspan="2">Informatics M.Sc.</td><td>Area of specialization</td><td>Big Data and Business Intelligence</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture</td><td>21h 00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>--</td><td rowspan="4">Semester</td><td rowspan="4">5</td></tr>
<tr><td>Workshop</td><td>21h 00</td><td rowspan="3">Mixed</td><td rowspan="3">XX</td></tr>
<tr><td>Project</td><td>--</td></tr>
<tr><td>Self Study</td><td>42h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">84h00 / Semester</td><td>Coefficient</td><td>3</td></tr>
<tr><td colspan="5"></td><td>ECTS credit</td><td>3</td></tr>
<tr><td colspan="2">Responsible Teacher</td><td colspan="2">Prof Zrigui Mounir</td><td>Involved Teacher(s)</td><td colspan="2">Prof Zrigui Mounir</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Prof. PHD. MOUNIR ZRIGUI
-M.A. PHD. M.A. OMRANI
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">Obj 1 and Ob j2</td></tr>
<tr><td>PLO</td><td colspan="2">P2: Implementation Techniques
P3: Innovation, Research and Emerging Technologies
P8: Communication and Collaboration</td></tr>
<tr><td>Prerequisites:</td><td colspan="7">Data mining fundamentals</td></tr>
<tr><td>Corequisites:</td><td colspan="7"></td></tr>
<tr><td>Learning outcomes</td><td colspan="7">Upon completion of this teaching module, the student will be able to:

1. Review and reinforce their understanding of fundamental data mining concepts.
2. Apply advanced data mining algorithms and techniques to solve complex data analysis problems.
3. Evaluate and interpret the results of data mining models.
4. Implement data mining projects in real-world scenarios.
5. Discuss the ethical and legal aspects of data mining.</td></tr>
<tr><td>Content:</td><td colspan="7">Chapter 1: Review of Basic Data Mining Concepts
1. Key data mining concepts and terminology
2. Data preprocessing and cleaning
3. Data mining process and techniques
4. Introduction to ethical considerations</td></tr>
</table>

	Chapter 2: Advanced Data Mining Techniques ● Association rule mining ● Text mining and natural language processing ● Anomaly detection ● Clustering algorithms **Chapter 3: Advanced Machine Learning for Data Mining** 1. Deep learning for data mining 2. Ensembles and stacking 3. Time series analysis 4. Reinforcement learning in data mining **Chapter 4: Big Data and Data Mining** 1. Introduction to big data technologies 2. Distributed data mining 3. Scalable data mining algorithms 4. Case studies in big data mining **Chapter 5: Real-World Applications** 1. Data mining in healthcare 2. Data mining in finance 3. Data mining in marketing 4. Data mining in social media and recommendation systems
Teaching and learning method(s):	Face to face for 45 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned. ● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1,2 and 3.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 4 and 5**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain fundamental data mining concepts and apply advanced data mining techniques. By the end of the final exam, students should be able to implement data mining projects in real-world scenarios and discuss the ethical and legal aspects of data mining.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	**1.** Advanced data analysis and interpretation **2.** Practical implementation of data mining techniques **3.** Ethical and legal awareness in data mining

	4. Real-world problem-solving using data mining
Teaching Manual:	• Powerpoint presentation • Tutorial booklet • Workbooks
References:	[1] Data Mining et Statistique décisionnelle, Stéphane Tufféry, Editions Technip 2017 [2] Data mining Découverte de connaissances dans les données, Daniel T. Larose, Chantal D. Larose, Vuibert Edition 2018 [3] Exploration de Données et Méthodes Statistiques Data Analysis & Data Mining avec le Logiciel R, Lise Bellanger, Richard Tomassone, Edition ELLIPSES 2014

Department of Computer Science
BDBI-C-5.6.2-NoSQL Databases

Last Update Date: May 7, 2024

Syllabus Sheet

Unit	**Big Data and Business Intelligence 3**		**Course**	**NoSQL Databases**		**Code**	**BDBI-C-5.6.2**
Department	**Computer Science Department**		**National Diploma**	**Informatics M.Sc.**		**Area of specialization**	**Big Data and Business Intelligence**
Type of education	**Lecture**	**21h 00**	**Examination regime**	**Continuous Assessment**	**--**	**Semester**	**5**
	Workshop	**21h 00**					
	Project	**--**		**Mixed**	**XX**		
	Self Study	**42h00**					
Total workload		**84h00 / Semester**			**Coefficient**		**3**
					ECTS credit		**3**
Responsible Teacher		**Mr Mohsen Maraoui**		**Involved Teacher(s)**	**Mr Mohsen Maraoui**		
Review committee		**-Prof. PHD. MOUNIR ZRIGUI -M.A. PHD. M.A. OMRANI -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI**		**Objective**	**Obj 1 and Ob j2**		
				PLO	**P2: Implementation Techniques P3: Innovation, Research and Emerging Technologies P8: Communication and Collaboration**		

Prerequisites:	Basic understanding of databases and relational database systems. Knowledge of database design and normalization. Familiarity with SQL syntax and querying techniques.
Corequisites:	Distributed Systems.
Learning outcomes	Upon completion of this teaching module, the student will be able to: – Define and explain the foundational concepts of NoSQL databases. – Distinguish between various NoSQL database types, including document, key-value, column-family, and graph databases. – Design, model, and implement database solutions using NoSQL technologies. – Analyze the advantages, use-cases, and limitations of NoSQL databases over traditional RDBMS. – Apply consistency, availability, and partition tolerance (CAP theorem) principles in the context of NoSQL databases. – Evaluate security and performance considerations specific to NoSQL databases. – Implement scaling and replication strategies for NoSQL databases. – Understand the integration of NoSQL databases in modern application architectures.

Content:	**Chapter 1:** Introduction to NoSQL 1. What is NoSQL? 2. History and evolution of NoSQL 3. Why NoSQL? **Chapter 2: Types of NoSQL Databases** 1. Document databases (e.g., MongoDB) 2. Key-Value stores (e.g., Redis) 3. Column-family stores (e.g., Cassandra) 4. Graph databases (e.g., Neo4j) **Chapter 3: NoSQL Database Design** 1. Data modeling for NoSQL 2. Schema-less design considerations 3. Use case Design task **Chapter 4: CAP Theorem** 1. Consistency 2. Availability 3. Partition tolerance **Chapter 5: Performance and Scaling** 1. Scaling strategies 2. Replication and sharding 3. Advanced case **Chapter 6: Security in NoSQL** 1. Access control 2. Encryption techniques **Chapter 7: Real-world Applications and Integration** 1. Use cases for NoSQL databases 2. Integration in modern web applications
	List of Workshops: 1. Setting up MongoDB and basic operations. 2. Designing a data model for a NoSQL database. 3. Redis in action: caching techniques. 4. Graph database modeling using Neo4j. 5. Scaling and replication in Cassandra. 6. Implementing security measures in a NoSQL environment. 7. Building a mini application integrating a NoSQL backend.

	Self-Study Project: Research and develop a prototype of a real-time chat application that uses a combination of NoSQL databases to handle different data structures (e.g., user profiles, chat histories, online statuses). The application should have the following features: User Profiles: Allow users to register, log in, and manage their profiles. Real-time Messaging: Users should be able to send and receive messages in real time. Chat Histories: Store and retrieve past conversations. Online Statuses: Show when users are online, offline, or away. Technical Requirements: User Profiles: Use a document-based NoSQL database like MongoDB to manage user profiles. This will allow for flexibility in storing varied user data. Real-time Messaging: Consider using a key-value store like Redis for caching current online users and managing real-time conversations. Chat Histories: A column-family store like Cassandra would be suitable for storing chat histories given its scalability for write-intensive workloads. Online Statuses: A graph database like Neo4j can be used to establish relationships between users and manage online statuses, showing connections and active chat participants. Expected Deliverables: Codebase: Complete source code of the application, including database setup scripts. Documentation: A comprehensive documentation detailing: The architecture of the application. Rationale for choosing specific NoSQL databases for each feature. Setup and deployment instructions. Presentation: A presentation that summarizes the project, demonstrates the application, and highlights the benefits of using NoSQL databases for specific components. References for Guidance: "MongoDB: The Definitive Guide" by Shannon Bradshaw, Eoin Brazil, and Kristina Chodorow. "Redis in Action" by Josiah L. Carlson. "Mastering Apache Cassandra" by Nishant Neeraj. "Neo4j in Action" by Aleksa Vukotic and Nicki Watt. Evaluation Criteria: The project will be evaluated based on: Functionality and completeness of the application. The efficiency of database operations (reading, writing, updating, deleting). Quality and clarity of the documentation and presentation. The rationale behind the selection of specific NoSQL databases for different components.
Teaching and learning method(s):	Face to face 84 hours. A course space is created on the easylearn.ovh platform to ensure distance learning and self-study. This space includes a set of educational resources available to learners.
Teaching technique(s):	Classroom lectures with tutorials and practical work
Assessment method(s):	A midterm written test (D.S) and a final written test (Exam) are planned.

	● The mid-term written test will be devoted to the evaluation of the level of achievement of **learning outcomes 1, 2, 3, and 4.** ● The final written test is devoted to the evaluation of the level of achievement of **learning outcomes 5, 6, 7 and 8.**
Assessment criteria(s) :	During the midterm exam, students should be able to define and explain the foundational concepts, different types of NoSQL databases, and basic design strategies. By the end of the final exam, students should be able to discuss CAP theorem, implement security and performance considerations, and integrate NoSQL databases in real-world applications.
Assessment measure(s):	Assigning of a final average
Criteria for assigning the final grade	The final average ranges from 0 (minimum) to 20 (maximum) The minimum validation average is 10/20 The final average is calculated as follows: ● Mid-term written test (D.S): 20% ● Practical work: 20% ● Final written exam: 60%.
Target skill(s):	1. NoSQL database design and modeling. 2. Efficient querying in different NoSQL environments. 3. Performance tuning and optimization in NoSQL. 4. Understanding of CAP theorem's implications. 5. NoSQL database scaling and replication techniques. 6. Integration of NoSQL in application architectures.
Teaching Manual:	● PowerPoint presentation ● Tutorial booklet ● Workbooks
References:	[1] "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence" by Martin Fowler and Pramod Sadalage. [2]"Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement" by Luc Perkins, Eric Redmond, and Jim R. Wilson. [3] "NoSQL for Mere Mortals" by Dan Sullivan.

Computer Science Department

UE-6.1 Graduation research project

Last Update date: June 01, 2024

Syllabus SHEET

<table>
<tr><td>Unit</td><td colspan="3">Graduation research project</td><td>Course</td><td colspan="2">Graduation research project</td><td>Code</td><td>UE-6.1</td></tr>
<tr><td>Department</td><td colspan="3">Computer science</td><td>National Diploma</td><td colspan="2">Computer science Engineering</td><td>Area of specialization</td><td>Common core</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture (Courses/TD)</td><td>00h00</td><td rowspan="4">Examination regime</td><td>Continuous Assessment</td><td>---</td><td rowspan="2">Semester</td><td rowspan="2">3</td></tr>
<tr><td>Workshop</td><td>00h00</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>600h0</td></tr>
<tr><td>Self- study</td><td>00h00</td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">600h00 / Semester</td><td colspan="2">Coefficient</td><td>22</td></tr>
<tr><td colspan="5"></td><td colspan="2">ECTS credit</td><td>22</td></tr>
<tr><td colspan="2">Responsible teacher</td><td colspan="2">------</td><td>Involved Teacher(s)</td><td colspan="3">----</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Profess. PHD. Mounir Zrigui
-M.A. PHD. OMRANI AMINE
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="3">2. Professional Development and Industry-Focused Leadership</td></tr>
<tr><td>PLO</td><td colspan="3">P1: Technical Proficiency and Application</td></tr>
<tr><td colspan="2"><u>Prerequisites:</u></td><td colspan="7">The engineering student is required to demonstrate perfect knowledge of the learning outcomes as described in their program.</td></tr>
<tr><td colspan="2"><u>Learning outcomes</u></td><td colspan="7">At the end of this PFE, the engineering student is expected to:
1. Have obtained more maturity in relation to their field of Computer Engineering.</td></tr>
</table>

	2. Be able to define an overall working methodology to solve a complex problem. 3. Be able to describe the difficulties encountered in solving a real problem. 4. Have more experience in terms of integrating into a new team and being able to work with sharing and transparency. 5. Be able to communicate clearly about the activities of your project, the solutions and relate them to the cumulative learning outcomes. 6. Integrate aspects of scientific and technological knowledge accumulated during your career into each stage of your work. 7. Demonstrate exemplary disciplinary capacity.
Content:	The PFE must be described explicitly by mutual agreement between the academic supervisor and the company supervisor.
Teaching and learning method:	Project-based learning.
Assessment method	At the end of their internship, the trainee is required to submit a manuscript describing the topic, the achievements, along with the most objective analysis possible. After validation by their department, the trainee must go through the following steps: **I- Stage 1 : Evaluation by the Supervisor** **I-1- Objective :** Evaluation of the initial planning, understanding of the topic, work methodology, and project monitoring by the student. **I-2- Evaluation Criteria:** ● Clarity of the project proposal. ● Relevance of the topic to the field of electrical engineering. ● Quality of the literature review. ● Research and development methodology. ● Adherence to deadlines and the initial schedule. ● Communication and interaction with the supervisor. **II- Stage 2 : Evaluation by the Reviewer during the Technical Presentation** **II-1 Objective :** Assessment of the student's technical skills, ability to present and defend the project before a review committee.

	II-2 Evaluation Criteria: ● Quality of the oral presentation. ● Mastery of technical concepts related to the project. ● Ability to respond to questions from the review committee. ● Originality and contribution of the project to the field of Computer science engineering. ● Quality of the written report. **III- Stage 3 : Evaluation during the Final Presentation** **III-1- Objective :** Comprehensive evaluation of the project and the student's performance in presenting the work accomplished. **III-2- Evaluation Criteria:** ● Clarity and relevance of the project presentation. ● Quality of the demonstration (if applicable). ● Contribution to advancements in the field of electrical engineering. ● Responses to questions from the jury during the final presentation. **IV- Stage 4: Final Grade and Validation** **IV-1- Objective:** Compilation of the previous evaluations to assign a final grade to the final year project. **IV-2- Evaluation Criteria:** ● Evaluation by the supervisor. ● Evaluation by the reviewer during the technical presentation. ● Evaluation during the final presentation. ● The final grade for the final year project is calculated, taking into account the three previous evaluations. An average or weighted score can be assigned based on the institution's criteria. The project's validation is determined by achieving a predefined minimum grade. This four-stage evaluation approach ensures a comprehensive, objective, and fair assessment of the final year project in Computer science engineering. It highlights technical skills, communication, planning, and the contribution to the field of Computer science engineering.
Assessment criteria	The evaluation of the **Graduate project** will focus on the following points: • The quality of the report and the oral presentation • The quality of the answers to the questions.

	• The choice of the methodology adopted to resolve the problems, Rigor of the approach: (choice, tools, method, summaries, etc.) • The relevance of the proposed solutions. • Project value: (Compliance with specifications, execution, technical file, • The degree of integration of the solutions proposed into the company's activities.
Evaluation measure	Award of a final average and a mention.

Computer Science Department
UE-6.2 Operational Internship 1

Last Update date: June 28, 2024

Syllabus SHEET

Unit	**Operational Internship 1 Unit 6**		**Subject**	**Operational Internship 1**		**Code**	**UE-6.2**
Department	**Computer Science**		**National Diploma**	**Computer Science Department**		**Area of specialization**	**Common Core**
Type of education	**Lecture (Courses/TD)**	**00h00**	**Examination regime**	**Continuous Assessment**		**Semester**	**6**
	Workshop	**00h00**					
	Project	**120h0**		**Mixed**	**XX**		
	Self- study	**00h00**					
Total workload	**120h (1 or 2 month in the summer)**				**Coefficient**		**4**
					ECTS credit		**4**
Responsible teacher	**—**		**Involved Teacher(s)**	**—**			
Review committee	-Profess. PHD. Mounir Zrigui -M.A. PHD. OMRANI AMINE -M.A. PHD. WAEL SELLAMI - Director of study A. SELMI		**Objective**	2. Professional Development and Industry-Focused Leadership			
			PLO	P1: Technical Proficiency and Application			

Prerequisites:	Scientific Backgrounds, Programming, Design Tools, writing skills
learning outcomes	Industrial internships are an important part of the Engineering Programs . These internships are carried out with industrial partners to provide the optimum learning experience for students as well as industrial contacts for the University
Thrilled :	-Student intern will be part of a team -Site tours, and exposure to executives and mentors. -Doing some engineering tasks (creating reports, analysis of a specific issue) -Other tasks are assigned depending on backgrounds, skills
Teaching and learning method :	120 hours (30 hours per week for 4 weeks) - 1 month in summer holiday

requirements and forms of examination	Students will demonstrate their progress by the following: 1. Recording notes about their daily tasks & activities 2. Presenting a work-in-progress talk 3. Submitting a manuscript by the specified deadline
Teaching Manual :	Project Ball
Bibliographic References :	Dissertation Guidebook - Other references given by supervisors

Computer Science Department
UE-6.3 Technical Internship 2

Last Update date: June 28, 2024

Syllabus SHEET

<table>
<tr><td>Unit</td><td colspan="2">Technical Internship 2
Unit 6</td><td>Subject</td><td colspan="2">Technical Internship 2</td><td>Coded</td><td>UE-6.3</td></tr>
<tr><td>Department</td><td colspan="2">Computer Science</td><td>Field</td><td colspan="2">Computer Science Department</td><td>Option</td><td>Common Core</td></tr>
<tr><td rowspan="4">Type of education</td><td>Lecture (Courses/TD)</td><td>00h00</td><td rowspan="4">Examination system</td><td>Continuous Assessment</td><td></td><td rowspan="2">Semester</td><td rowspan="2">6</td></tr>
<tr><td>Workshop</td><td>00h00</td><td rowspan="2">Mixed</td><td rowspan="2">XX</td></tr>
<tr><td>Project</td><td>120h0</td></tr>
<tr><td>Self- study</td><td>00h00</td><td></td><td></td></tr>
<tr><td colspan="2">Total workload</td><td colspan="3">120h (1 or 2 month in the summer)</td><td>Coefficient</td><td>4</td></tr>
<tr><td colspan="2"></td><td colspan="3"></td><td>ECTS credit</td><td>4</td></tr>
<tr><td colspan="2">Responsible teacher</td><td colspan="2">—</td><td>Involved Teacher(s)</td><td colspan="2">—</td></tr>
<tr><td colspan="2" rowspan="2">Review committee</td><td colspan="2" rowspan="2">-Profess. PHD. Mounir Zrigui
-M.A. PHD. OMRANI AMINE
-M.A. PHD. WAEL SELLAMI
- Director of study A. SELMI</td><td>Objective</td><td colspan="2">2. Professional Development and Industry-Focused Leadership</td></tr>
<tr><td>PLO</td><td colspan="2">P1: Technical Proficiency and Application</td></tr>
<tr><td colspan="2">Prerequisites:</td><td colspan="6">Scientific Backgrounds, Programming, Design Tools, writing skills</td></tr>
<tr><td colspan="2">learning outcomes</td><td colspan="6">Industrial internships are an important part of the Engineering Programs . These internships are
carried out with industrial partners to provide the optimum learning experience for students as well
as industrial contacts for the University</td></tr>
<tr><td colspan="2">Thrilled :</td><td colspan="6">-Student intern will be part of a team
-Site tours, and exposure to executives and mentors.
-Doing some engineering tasks (creating reports, analysis of a specific issue)
-Other tasks are assigned depending on backgrounds, skills</td></tr>
<tr><td colspan="2">Teaching and learning method :</td><td colspan="6">120 hours (30 hours per week for 4 weeks) - 1 month in summer holiday</td></tr>
</table>

requirements and forms of examination	Students will demonstrate their progress by the following: 1. Recording notes about their daily tasks & activities 2. Presenting a work-in-progress talk 3. Submitting a manuscript by the specified deadline
Teaching Manual :	Project Ball
Bibliographic References :	Dissertation Guidebook - Other references given by supervisors