

Course Title:	Software Engineering
Course Code:	CSE331
Program:	Computer science Engineering
Department:	Computer Engineering
Course coordinator:	Dr. Rim Afdhal
Institution:	Private Higher School of Engineers of Gafsa (ESIP)

A. Course Identification

1. Credit hours: 3 (2-1-0)			
2. Course type			
a. College Department Others			
b. Fundamental Transversal Optional			
3. Level/year at which this course is offered: 2.1/3			
4. Pre-requisites for this course (if any): CSE131, Databases, Programming			

1. Mode of Instruction (mark all that apply)

No	Mode of Instruction	Contact Hours	Self- study	Total workload
1	Traditional classroom			
2	Blended	33		
3	E-learning		35	80
4	Distance learning			
5	Other (Project)	12		

2. Contact Hours (based on academic semester)

No	Activity	Contact Hours
1	Lecture Privee de Gais	23
2	Laboratory/Studio	-
3	Tutorial	10
4	Others (Project)	12
	Total	45



B. Course Objectives and Learning Outcomes

Course Description

This course explores key software engineering principles, development methodologies, and best practices. It covers software life cycles, requirements engineering, system modeling, software design, and quality assurance. Students will learn about Agile methods, DevOps, CI/CD, software testing, and security. Practical sessions include hands-on experience with UML, Git, Docker, and automated testing tools to develop and maintain reliable software systems.

Course Main Objective

- ✓ Define software engineering and understand its key challenges.
- ✓ Identify software quality attributes and fundamental principles.
- Explain the software life cycle and compare traditional and modern development methodologies.
- ✓ Address design challenges and ethical considerations in software engineering.
- ✓ Analyze and solve software engineering problems to develop optimal solutions.

CLO	CLOs	
1	Knowledge and Understanding	
1.1	Define software engineering and understand its key challenges.	
1.2	Identify software quality attributes and fundamental principles	PLO K1
1.3	Explain the software life cycle and compare traditional and modern development processes.	I DOMAI
2	Skills	
2.2	Address design challenges and ethical considerations in software engineering.	PLO.S2
2.6	Apply object-oriented programming concepts in the design of embedded software to enhance modularity, reusability, and maintainability.	PLO.S6
2.7	Analyze and solve software engineering problems to develop optimal solutions.	PLO.S7

1. Course Learning Outcomes

C. Course Content

No	List of Topics	Contact Hours
1	 Chapter 1: Introduction to Software Engineering 1. Definition and key concepts of Software Engineering 2. History and evolution (Software crisis, development paradigms) 3. Software development processes 4. Core principles: modularity, abstraction, reusability 5. Application domains 	3
2	Chapter 2: Software Life Cycle and Development Methodologies1. Definition and models of the software life cycle2. Traditional approaches: Waterfall, V-Model, Spiral	4



No	List of Topics	Contact Hours
	3. Modern approaches: Agile (Scrum, Kanban, Extreme	
	Programming)	
	4. Rational Unified Process (RUP) and DevOps	
	5. Software lifecycle management (ALM - Application	
	Lifecycle Management)	
	 Chapter 3: Requirements Engineering and Analysis (4 hours) 1. Definition and classification of requirements (functional, non-functional, constraints) 2. Techniques for gathering requirements (interviews, surveys, surveys) 	
3	observation, prototyping)	4
5	3. Requirement modeling (UML diagrams: use cases, scenarios,	
	class diagrams)	
	4. Validation and verification of requirements	
	5. Change management and traceability	
	Chapter 4: Software Design and Architecture	
	1. Fundamental principles of software design	
	2. Design approaches: procedural, object-oriented (UML,	
	Design Patterns)	
	3. Software architecture: microservices, SOA (Service-Oriented	
4	Architecture), monolithic vs. distributed	5
	 Advanced design patterns (MVC, Singleton, Factory, Observer, Adapter) 	
	5. Software architecture modeling with UML: component	
	diagrams, sequence diagrams	
	6. Security and robustness in software design	
	Chapter 5: Software Development and Engineering Tools (4 hours)	
	1. Advanced programming languages (Java, Python, C++,	11
	Kotlin)	
Ec	 Development environments and tools (IDEs, Git, Docker, CI/CD) 	nieur
5	3. Dependency management and versioning (Maven, Gradle, GitHub, GitLab)	4
	4. Code quality and code reviews (Code smells, refactoring.	
	SonarQube)	
	5. Model-Driven Architecture (MDA - Model-Driven	
	Engineering)	
6	Chapter 6: Validation, Testing, and Software Quality Assurance	A
0	1. Strategies and principles of software testing	4



No	List of Topics	Contact Hours
	2. Types of tests: unit, integration, functional, non-functional	
	(performance, security, compatibility)	
	3. Modern approaches: Test-Driven Development (TDD), Behavior-	
	Driven Development (BDD)	
	4. Test automation: Selenium, JUnit, PyTest, Cypress	
	CI/CD and DevOps: Continuous integration and continuous delivery	
Total		23

D. Tutorial Content

No	List of Topics	Contact Hours
1	Expression and analysis of Needs	1
2	Case Study	2
3	Software development process and process model	2
4	Conception	2
5	Black and White box testing	3
	Total	10

Project Content

No	List of Topics	Contact Hours
1	Preliminary study	1.5
2	Needs analysis and specification	1.5
3	Description of the backlog product	1.5
3	Development of sprint1	1.5
4	Development of sprint2	2
5	Development of sprint3	2
Total		10

Ecole Supérieure d'Ingénieurs

E. Teaching and Assessment

1. Alignment of Course Learning Outcomes with Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
1.0	Knowledge and Understanding		
PLO.K1	Define software engineering and understand its key challenges.	- Lecturing	
	Identify software quality attributes and		



Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
	fundamental principles. Explain the software life cycle and compare traditional and modern		- Assignments, Quizzes , Exams,
2.0	Skills		
PLO.S2	Address design challenges and ethical considerations in software engineering. Apply object-oriented programming concepts in the design of embedded software to enhance modularity, reusability, and maintainability. Analyze and solve software	- Lecturing	- Assignments, Quizzes , Exams,
PLO.S7	engineering problems to develop optimal solutions.		

2. Assessment Tasks for Students

#	Assessment task*	W	eek Due	Percentage of Total Assessment Score
1	Practical Work (written or oral)		Weekly	10%
2	Quizzes, Homework assignments		Random	10%
3	First mid Term		8	25%
4	Final Exam		16	65%

Student Academic Counselling and Support

Arrangements for availability of faculty and teaching staff for individual student consultations and academic advice:

- Office hours
- Blackboard interface

F. Learning Resources and Facilities

1. Learning Resources

Required Textbooks1. Sommerville, I. – Software Engineering (10th Edition),
Pearson.



	 Pressman, R. S., & Maxim, B. R. – Software Engineering: A Practitioner's Approach (9th Edition), McGraw-Hill Gamma, E., Helm, R., Johnson, R., & Vlissides, J. – Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley.
Essential References Materials	NA
Electronic Materials	 edX: Software Engineering Essentials – www.edx.org Coursera: Software Development Lifecycle & Agile Methodologies – www.coursera.org YouTube: Software Engineering Lectures & Tutorials
Other Learning Materials	- NA

2. Facilities Required

Item	Resources	
Accommodation		
(Classrooms, laboratories, demonstration	classroom board software	
rooms/labs, etc.)		
Technology Resources	data shawe	
(AV, data show, Smart Board, software, etc.)	uata snow;	

G. Course Quality Evaluation

Evaluation Areas/Issues	Evaluators	Evaluation Methods
Effectiveness of teaching and	Students, course coordinator, Alumni,	Direct/Indirect
assessment.	Employers	Direct/indirect
Extent of achievement of course	Faculty, Program Leaders, quality	Direct
learning outcomes.	department	
Quality of Learning resources	Faculty, Program Leaders,	Direct, Indirect
Teaching and learning quality	Students Faculty Program Londors	Direct, Indirect
and effectiveness.	Students, Faculty Flogram Leaders,	

H. Specification Approval Data

Council / Committee	Computer Engineering Council
Date	11/09/2023