Course Title:	**Object-oriented analysis and design**
Course Code:	CSE332
Program:	Master Degree In Computer 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 (1 -0.5-1.5)**

2. Course type

a. University ☐ College ☐ Department ■ Others ☐

b. Required ■ Elective ☐

3. Level/year at which this course is offered: 2.1/3

4. Pre-requisites for this course (if any)**:** basic of POO, Algorithms and data structures (CSE131), basic of UML

1. Mode of Instruction (mark all that apply)

No	Mode of Instruction	Contact Hours	Self-study	Total workload
1	**Traditional classroom**	…..	31	76
2	**Blended**	45		
3	**E-learning**	……		
4	**Distance learning**	……		
5	**Other (Specify)**	……		

2. Contact Hours (based on academic semester)

No	Activity	Contact Hours
1	**Lecture**	22.5
2	**Laboratory/Studio**	22.5
3	**Tutorial**	-
4	**Others**(specify)	-
	Total	45

B. Course Objectives and Learning Outcomes

Course Description

This course provides a comprehensive understanding of object-oriented analysis and design using UML (Unified Modeling Language) and the Unified Process Model. It focuses on mastering object-oriented concepts and applying UML modeling techniques to design software systems efficiently. The course emphasizes structural and behavioral modeling, enabling students to represent complex systems visually and simplify their development. Students will learn to create UML diagrams, choose appropriate models based on system requirements, and transition from conceptual design to implementation using object-oriented programming languages (C++, Java, etc.). Additionally, the course introduces reverse engineering techniques for analyzing and improving existing software systems.

Course Main Objective

- ✓ Understand the fundamental principles of the object-oriented approach.
- ✓ Master the Unified Modeling Language (UML) for software design.
- ✓ Identify and apply different UML models: static, dynamic, user-centered, architecture-centered.
- ✓ Develop and edit UML diagrams to represent system components and interactions.
- ✓ Manage a data dictionary for software documentation and traceability.
- ✓ Generate object-oriented code (C++, Java, etc.) from UML models.
- ✓ Apply reverse engineering techniques to extract models from existing source code.

1. Course Learning Outcomes

CLOs		Aligned PLOs
1	**Knowledge and Understanding**	
1.1	Understand the fundamentals of the object-oriented approach.	PLO.K1
1.2	Master the concepts and syntax of the Unified Modeling Language (UML).	PLO.K2
1.3	Differentiate between static and dynamic UML diagrams and determine their appropriate usage.	PLO.K3
2	**Skills**	
2.1	Create, edit, and refine UML models and diagrams.	PLO.S1
2.2	Manage a data dictionary for structured system documentation.	PLO.S2
2.3	Generate executable object-oriented code (C++, Java, etc.) from UML diagrams.	PLO.S3
2.4	Apply reverse engineering techniques to analyze and improve existing code.	PLO.S7

C. Course Content

No	List of Topics	Contact Hours
1	**Chapter 1:** Introduction to Object-Oriented Analysis and Design 1. Overview of software development paradigms 2. Definition and importance of Object-Oriented Analysis and Design (OOAD) 3. Core object-oriented principles: Encapsulation, Inheritance, Polymorphism, Abstraction 4. Advantages of object-oriented development	3
2	**Chapter 2:** Unified Modeling Language (UML) and Object-Oriented Concepts 1. Introduction to UML and its role in software development 2. Basic object-oriented concepts: Objects, Classes, Methods, Attributes 3. Relationships between objects: Association, Aggregation, Composition, Dependency 4. UML structural elements and their applications	4.5
3	**Chapter 3:** Structural (Static) Diagrams in UML 1. Use Case Diagrams – Identifying system actors and interactions 2. Class Diagrams – Representing system structure and relationships 3. Object Diagrams – Modeling instances of classes 4. Component Diagrams – Software module interactions **5.** Deployment Diagrams – Hardware/software system representation	2.5
5	**Chapter 4:** Behavioral (Dynamic) Diagrams in UML 1. Sequence Diagrams – Object interactions over time 2. Collaboration Diagrams – Message exchanges between objects 3. State Chart Diagrams – Object lifecycle states and transitions 4. Activity Diagrams – Process flow and system behavior modeling	6
7	**Chapter 5:** Software Development Process and Unified Process Model 1. Phases of the Unified Process Model: Inception, Elaboration, Construction, Transition 2. Best practices for software modeling using UML 3. Role of UML in Agile and iterative development 4. Case studies and real-world applications of OOAD	6
Total		22.5

C2. Practical Work Content

No	List of Topics	Contact Hours
1	Lab 1: Structural modeling: Use case diagrams	3
2	Lab 2: Object & Class diagrams	3
3	Lab 3: Dynamic modeling: Sequence diagrams	3
4	Lab 4: Collaborate diagrams	3
5	Lab 5: State charts diagrams	3
6	Lab 6: Activities diagrams	3
7	Lab 7: Unified process Model	4.5
	Total	22.5

D. 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	Understand the fundamentals of the object-oriented approach.	Lecturing	Assignments, Quizzes, Exams,
PLO.K2	Master the concepts and syntax of the Unified Modeling Language (UML).		
PLO.K3	Differentiate between static and dynamic UML diagrams and determine their appropriate usage.		
2.0	**Skills**		
PLO.S1	Create, edit, and refine UML models and diagrams.	Lecturing/Lab demonstration	Assignments, Quizzes, Exams,
PLO.S2	Manage a data dictionary for structured system documentation.		
PLO.S3	Generate executable object-oriented code (C++, Java, etc.) from UML diagrams.		
PLO.S7	Apply **reverse engineering techniques** to analyze and improve existing code.		

2. Assessment Tasks for Students

#	Assessmenttask*	Week Due	Percentage of Total Assessment Score
1	Practical Work (written or oral)	Weekly	25%
2	Quizzes, Homework assignments	Random	00%
3	First midTerm	9	25%

#	Assessmenttask*	Week Due	Percentage of Total Assessment Score
5	Final Exam	16	50%

E. Student Academic Counseling and Support

Arrangements for availability of faculty and teaching staff for individual student consultations and academic advice :
1- Office hours 2- Blackboard interface

F. Learning Resources and Facilities

1. Learning Resources

Required Text books	1. Muller, P. A. Modélisation Objet avec UML 2.5. 4th ed., Eyrolles, 2017. 2. Roques, Philippe. UML 2 en Action: De l'Analyse des Besoins à la Conception. 3rd ed., Eyrolles, 2018. 3. Warmer, Jos B., and Anneke G. Kleppe. The Object Constraint Language: Precise Modeling with UML. Addison-Wesley, 2003. 4. Muller, P. A., and Nathalie G. Modélisation Objet avec UML. 2nd ed., Eyrolles, Feb. 2000. 5. Roques, Philippe, and Frédéric Vallée. UML en Action. 2nd ed., Eyrolles, Nov. 2002. 6. Roques, Philippe. UML 2: Modéliser une Application Web. 4th ed., Eyrolles, Oct. 2008. 7. Rumbaugh, James, et al. The Unified Modeling Language Reference Manual. Addison-Wesley, 2005
Essential References Materials	-
Electronic Materials	1. **YouTube** – *Software Engineering & UML Tutorials* 2. **edX** – *Software Engineering Essentials* 3. **Coursera** – *Software Development Lifecycle & Agile Methodologies*
Other Learning Materials	NA

2. Facilities Required

Item	**Resources**
Accommodation	**Classroom board** **Computer lab with the necessary software** **Internet access**
Technology Resources	**Data projector**

G. Course Quality Evaluation

Evaluation Areas/Issues	Evaluators	Evaluation Methods
Effectiveness of teaching and assessment.	Students, Faculty, Program Leaders, Peer Reviewer	Direct/Indirect
Extent of achievement of course learning outcomes.	Faculty, Program Leaders, Peer Reviewer	Direct, Indirect
Quality of Learning resources	Faculty, Program Leaders, Peer Reviewer	Direct, Indirect
Teaching and learning quality and effectiveness.	Students, Faculty Program Leaders, Peer Reviewer	Direct, Indirect

H. Specification Approval Data

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