



Course Specification

(Bachelor)

Course Title: **Software Modelling and Analysis**

Course Code: **SE2301**

Program: **BSc in Software Engineering**

Department: **Software Engineering**

College: **College of Computing**

Institution: **Umm Al Qura University**

Version: **1.0**

Last Revision Date: **22/04/2025**



Table of Contents

A. General information about the course:	3
B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods	4
C. Course Content	5
D. Students Assessment Activities	6
E. Learning Resources and Facilities	6
F. Assessment of Course Quality	7
G. Specification Approval	7



A. General information about the course:

1. Course Identification

1. Credit hours: (3)

2. Course type

A. University College Department Track Others
 B. Required Elective

3. Level/year at which this course is offered: (2nd year/ 4th level)

4. Course General Description:

This course provides a comprehensive understanding of the principles and practices of software modeling and analysis. Students will learn how to design, represent, and analyze software systems using various modeling techniques and tools. The course emphasizes the application of Unified Modeling Language (UML), system modeling best practices, and formal analysis techniques to ensure software quality and reliability. Students will also explore methods for verifying and validating software models.

5. Pre-requirements for this course (if any):

SE2203 - Requirement Engineering

6. Co-requisites for this course (if any):

7. Course Main Objective(s):

Upon successful completion of this course, you will be able to:

- 1- Introduce the role of modeling in software development.
- 2- Teach students to create and interpret various software models, including structural, behavioral, and dynamic models.
- 3- Provide tools and techniques for analyzing models to ensure system correctness and performance.

2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	45	100%
2	E-learning	0	0
3	Hybrid	0	0





No	Mode of Instruction	Contact Hours	Percentage
	<ul style="list-style-type: none"> Traditional classroom E-learning 		
4	Distance learning	0	0

3. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	Lectures	45
2.	Laboratory/Studio	0
3.	Field	0
4.	Tutorial	0
5.	Others (specify)	0
Total		45

B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
1.0	Knowledge and understanding			
1.1	Understand the purpose and importance of software modeling and analysis.	K2	Lecture, Exercise	Quiz, Exams, Assignments
1.2	Learn the syntax and semantics of UML diagrams and their applications.	K1	Lecture, Exercise	Quiz, Exams, Assignments
1.3	Explore formal analysis methods for verifying and validating software models.	K1	Lecture, Exercise	Quiz, Exams, Assignments
2.0	Skills			
2.1	Design and interpret structural models (e.g., class diagrams, component diagrams).	S1	Lecture, Exercise	Quiz, Exams, Assignments



Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
2.2	Develop behavioral models (e.g., sequence diagrams, state machine diagrams).	S1	Lecture, Exercise	Quiz, Exams, Assignments
2.3	Apply static and dynamic analysis techniques to ensure software quality.	S3	Lecture, Exercise	Quiz, Exams, Assignments
2.4	Use modeling tools such as Enterprise Architect, MagicDraw, or Visual Paradigm.	S3	Lecture, Exercise	Quiz, Exams, Assignments
3.0	Values, autonomy, and responsibility			
3.1	Promote the creation of high-quality, maintainable, and reliable software systems.	V1	Lecture, Exercise	Quiz, Exams, Assignments
3.2	Commit to rigorous analysis and ethical practices in software modeling.	V1	Lecture, Exercise	Quiz, Exams, Assignments

C. Course Content

No	List of Topics	Contact Hours
1.	Introduction to Software Modeling and Analysis: Role of models in software engineering	3
2.	Benefits and challenges of software modeling	3
3.	Introduction to UML and Modeling Standards: Syntax and semantics of UML diagrams	3
4.	Overview of modeling standards (e.g., SysML, BPMN)	3
5.	Structural Modeling: Class diagrams, object diagrams, and package diagrams	3
6.	Component diagrams and deployment diagrams	3
7.	Behavioral Modeling: Sequence diagrams and communication diagrams	3
8.	Activity diagrams and state machine diagrams	3





9.	Dynamic Modeling: Interaction overview diagrams and timing diagrams	3
10.	Model Analysis and Validation: Static analysis methods	3
11.	Dynamic analysis and simulation of software models	3
12.	Formal Methods in Software Modeling: Formal languages (e.g., Z notation, Alloy)	3
13.	Verifying and validating software models	3
14.	Modeling Tools and Frameworks: Overview of popular modeling tools	3
15.	Hands-on practice with tools such as Enterprise Architect, MagicDraw, and Visual Paradigm	3
Total		45

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Quizzes	2-14	15
2.	Projects	2-14	15
3.	Assignments	2-14	10
4.	Mid Term	7	20
5.	Final Exam	16-17	40

*Assessment Activities (i.e., Written test, oral test, oral presentation, group project, essay, etc.).

E. Learning Resources and Facilities

1. References and Learning Resources

Essential References	<ul style="list-style-type: none"> Gomaa, H. (2011). <i>Software modeling and design: UML, use cases, patterns, and software architectures</i>. Cambridge University Press. ISBN 978-0521764148. Fowler, M. (2018). <i>UML distilled: A brief guide to the standard object modeling language</i>. Addison-Wesley Professional. ISBN 978-0321193681. Blaha, M., & Rumbaugh, J. (2005). <i>Object-oriented modeling and design with UML (2nd ed.)</i>. Pearson. ISBN 978-0131968592.
Supportive References	



Electronic Materials

Other Learning Materials

2. Required Facilities and equipment

Items	Resources
facilities (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	Traditional Classroom
Technology equipment (projector, smart board, software)	Multimedia Projector
Other equipment (depending on the nature of the specialty)	N/A

F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of teaching	Students	Direct, Indirect
Effectiveness of Students assessment	Faculty, Peer reviewer	Direct, Indirect
Quality of learning resources	Faculty, Course coordinator	Direct, Indirect
The extent to which CLOs have been achieved	Course coordinator, Program management committee	Direct
Other		

Assessors (Students, Faculty, Program Leaders, Peer Reviewers, Others (specify))

Assessment Methods (Direct, Indirect)

G. Specification Approval

COUNCIL /COMMITTEE	SOFTWARE ENGINEERING DEPARTMENT COUNCIL
REFERENCE NO.	THE 17 TH MEETING FOR THE ACADEMIC YEAR 1446H
DATE	22/04/2025

