Course Specification

(Bachelor)

Course Title:	Complex System Engineering
Course Code:	SE3610
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: (3rd year/ 5th or 6th level) or (4th year/8th level)					
4. Course General Description:					
This course introduces students to the principles and practices of engineering complex systems. Students will explore the design, analysis, and management of large-scale software systems that exhibit high levels of complexity due to their size, interdependencies, or dynamic behaviors. Topics include system modeling, architectural patterns, scalability, resilience, and system-of-systems engineering. The course emphasizes practical techniques and tools to manage complexity in software systems effectively.					
5. Pre-requirements for this course (if any)**:**					
SE3103 - Software Testing and Quality Assurance					
6. Co-requisites for this course (if any)**:**					
N/A					
7. Course Main Objective(s):					
Upon successful completion of this course, you will be able to: 1. Understand the characteristics of complex systems and their unique engineering challenges. 2. Learn techniques for designing scalable and resilient software systems. 3. Apply system modeling and simulation to analyze system behavior. 4. Explore the application of software engineering principles in system-of-systems contexts.					

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
	• 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	Define the principles and challenges of complex systems engineering.	K1	Lecture, Exercise	Quiz, Exams
1.2	Understand system architecture, scalability, and resilience in software systems.	K1	Lecture, Exercise	Quiz, Exams
1.3	Learn the methodologies for system modeling and simulation.	K2	Lecture, Exercise	Quiz, Exams
2.0	**Skills**			
2.1	Design scalable and maintainable architectures for complex systems.	S1	Lecture, Exercise	Quiz, Exams
2.2	Perform system analysis using modeling tools and simulation frameworks.	S2	Lecture, Exercise	Quiz, Exams

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
2.3	Apply fault-tolerant and resilient engineering practices.	S2	Lecture, Exercise	Quiz, Exams
2.4	Use tools for monitoring, testing, and maintaining complex software systems.	S3	Lecture, Exercise	Quiz, Exams
3.0	**Values, autonomy, and responsibility**			
3.1	Promote sustainable engineering practices in complex systems development.	V1	Lecture, Exercise	Quiz, Exams
3.2	Commit to ethical and responsible engineering decisions.	V1	Lecture, Exercise	Quiz, Exams

C. Course Content

No	List of Topics	Contact Hours
1.	Introduction to Complex Systems: Definition, characteristics, and examples in software engineering	3
2.	System Architecture and Design: Architectural patterns, scalability, and performance optimization	6
3.	System Modeling and Simulation: Techniques (e.g., UML, SysML) and tools for analysis	6
4.	Resilience and Fault Tolerance: Designing fault-tolerant and resilient systems	3
5.	Dynamic Behaviors in Complex Systems: Emergent behaviors, interactions, and feedback loops	6
6.	System-of-Systems Engineering: Challenges and interdependencies in SoS contexts	6
7.	Scalability and Load Balancing: Techniques for scalability and load balancing in distributed systems	6
8.	Tools for Complex Systems Engineering: Modeling, monitoring, and observability tools	3

9.	Testing and Maintenance of Complex Systems: Strategies and CI/CD pipelines	3
10.	Case Studies and Applications: Real-world examples and lessons learned	3
Total		45

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Assignments and Quizzes	2-14	15
2.	Projects	2-14	15
3.	Practicals	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	Flumerfelt S, Schwartz KG, Mavris D, Briceno S. *Complex Systems Engineering: Theory and Practice*.; 2019.
Supportive References	Oliver DW, Kelliher TP, Keegan JG. *Engineering Complex Systems with Models and Objects*. McGraw-Hill Companies; 1997. Bass L, Clements P, Kazman R. *Software Architecture in practice: Software Architect Practice_c3*. Addison-Wesley; 2012.
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

Items	Resources
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 17TH MEETING FOR THE ACADEMIC YEAR 1446H**
DATE	**22/04/2025**