Course Specification

(Bachelor)

Course Title:	Software Design and Architecture
Course Code:	SE3102
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:

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 level)					
4. Course General Description:					
The primary objective of the Software Architecture course is to provide students with a comprehensive and practical understanding of software architecture—its creation, documentation, and application in real-world scenarios. The course combines fundamental theoretical concepts from the extensive body of knowledge on software architecture with essential best practices and key insights for successfully applying these principles in industrial projects of any scale. Students will explore various example designs, engaging in discussions to evaluate the trade-offs of critical design decisions. They will be expected to demonstrate mastery of theoretical concepts, make informed design choices, and effectively communicate their architectural designs to others.					
5. Pre-requirements for this course (if any)**:**					
SE1201 - Foundations of Software Engineering					
6. Co-requisites for this course (if any)**:**					
N/A					
7. Course Main Objective(s):					
Upon successful completion of this unit, students should be able to: 1. Equip students with foundational knowledge and advanced principles of software design and architecture, including modeling techniques and design patterns. 2. Develop students' ability to analyze, evaluate, and create software architectures that meet specified quality attributes such as scalability, maintainability, and reliability. 3. Foster problem-solving and critical-thinking skills to address practical challenges in software design and architecture across diverse application domains.					

4. Instill ethical and professional responsibility in students while encouraging effective teamwork and collaboration in designing and evaluating high-quality software systems.

2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	60	100%
2	E-learning	0	0
3	Hybrid • Traditional classroom • E-learning	0	0
4	Distance learning	0	0

3. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	**Lectures**	30
2.	**Laboratory/Studio**	30
3.	**Field**	0
4.	**Tutorial**	0
5.	**Others (specify)**	0
Total		60

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	Explain core concepts, principles, and methodologies of software design and architecture.	K1	Lectures, Interactive Discussions	Quizzes, Assignments, Exams
1.2	Assess software architectures based on quality attributes such as performance, security, and usability.	K2	Lectures, Interactive Discussions	Quizzes, Assignments, Exams
2.0	**Skills**			

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
2.1	Apply modeling techniques, such as static and dynamic modeling, to create detailed software designs.	S1	Exercises, Group projects	Assignments, Project reports and presentations
2.2	Analyze and evaluate different software architectural styles and design patterns for specific needs.	S2	Exercises, Group projects	Assignment, Project reports and presentations
2.3	Design scalable, maintainable, and reliable software architectures for various software applications.	S2	Exercises, Group projects	Assignments, Project reports and presentations
3.0	**Values, autonomy, and responsibility**			
3.1	Demonstrate an understanding of professional responsibilities and ethical considerations in software design and architecture decisions.	V1	Role-playing ethical scenarios, Reflection case study	Ethical case study evaluation
3.2	Collaborate effectively within a team to design and evaluate software architectures.	V2	Group Projects, Tutorials	Group Project

C. Course Content

No	List of Topics	Contact Hours
1.	**Introduction to Software Design and Architecture** • Importance and principles • Role in software development lifecycle	**4**
2.	**Software Modeling and Design Methods** • Static Modeling • Object and Class Structuring • Dynamic Interaction Modeling • Finite State Machines • State-Dependent Dynamic Interaction Modeling	**12**
3.	**Design Patterns** • Creational, Structural, and Behavioral Patterns	**4**

	• Application of Design Patterns	
4.	**Architectural Styles and Views** • Layered, Client/Server, and Component-Based Architectures • Service-Oriented Architectures (SOA) • Real-Time and Concurrent Architectures	**8**
5.	**Designing Software Architectures** • Subsystem Architectural Design • Object-Oriented Software Architecture Design • Client/Server and Distributed Systems • Service-Oriented and Component-Based Architectures	**12**
6.	**Software Quality Attributes and Architecture Assessment** • Key Quality Attributes (Performance, Scalability, Security, etc.) • Methods for Architecture Assessment	**8**
7.	**Software Architecture Solutions** • Software Interfaces • Methods for Architecture Assessment	**10**
8.	**Emerging Trends in Software Design and Architecture** • Microservices Architecture • Cloud-Native Architectures • AI-Driven Architectural Design	**2**
Total		**60**

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	• Bass, L., Clements, P., & Kazman, R. (2021). *Software architecture in practice* (4th ed.). Addison-Wesley. • Gomaa, H. (2011). *Software modeling and design: UML, use cases, patterns, and software architectures*. Cambridge University Press. • Van Vliet, H. (2008). *Software engineering: Principles and practice* (3rd ed.). John Wiley & Sons.
Supportive References	• Cervantes, H., & Kazman, R. (2024). *Designing software architectures: A practical approach* (2nd ed.). Addison-Wesley Professional. • Richards, M., & Ford, N. (2020). *Fundamentals of software architecture: An engineering approach*. O'Reilly Media. • Roberts, M. (2018, May). Serverless architectures. Retrieved from https://www.Martinfowler.com • Srikrishnan, Y. (2017). The death of the architect. *DZone*. Retrieved from https://dzone.com
Electronic Materials	• IEEE Computer Society. (2024). *Software design course*. Retrieved from https://www.computer.org/product/education/software-design-course
Other Learning Materials	

2. Required Facilities and equipment

Items	Resources
facilities (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	Classroom
Technology equipment (projector, smart board, software)	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 17TH MEETING FOR THE ACADEMIC YEAR 1446H**
DATE	**22/04/2025**