Course Specification

(Bachelor)

Course Title:	Advanced Software Engineering
Course Code:	SE4011
Program:	BSc in Computer Science
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: (4th year/ 7th or 8th level)

4. Course General Description:

Software engineers today are less likely to design data structures and algorithms from scratch and more likely to build systems from library and framework components. In this course, students engage with concepts related to the construction of software systems at scale, building on their understanding of the basic building blocks of data structures, algorithms, program structures, and computer structures.
The course covers technical topics in four areas:
(1) concepts of design for complex systems,
(2) object oriented programming,
(3) techniques for robustness, including testing and static and dynamic analysis for programs, and
(4) concurrent software. Students will gain concrete experience designing and building medium-sized systems. This course substantially improves its students' ability to apply general computer science knowledge to real-world problems using real-world tools and techniques.

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

SE1201 - Foundations of Software Engineering

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

7. Course Main Objective(s):

After completing this course, students will:

1. Be comfortable with object-oriented concepts and with programming in the Java language
2. Have experience designing medium-scale systems with patterns
3. Have experience testing and analyzing software
4. Understand principles of concurrency and be able to build concurrent software

2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	٦٠	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**	٣٠
2.	**Laboratory/Studio**	٣٠
3.	**Field**	0
4.	**Tutorial**	0
5.	**Others (specify)**	0
Total		٦٠

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 key concepts of object oriented programming	K1	Lecture, Exercise	Quiz, Exams, Assignments
1.2	Understand Scalability and Optimization	K1	Lecture, Exercise	Quiz, Exams, Assignments
1.3	Recognize Collaborative Development Workflows	K1	Lecture, Exercise	Quiz, Exams, Assignments
2.0	Skills			
2.1	Use modern development tools to design and	S3	Lecture, Exercise	Quiz, Exams, Assignments

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
	implement nontrivial programs			
2.2	Design code that is easy to understand, maintain, and reuse	S2	Lecture, Exercise	Quiz, Exams, Assignments
2.3	Use and design libraries, frameworks, and APIs	S2	Lecture, Exercise	Quiz, Exams, Assignments
3.0	Values, autonomy, and responsibility			
3.1	Work responsibly in teams, ensuring transparency and integrity.	V2	Lecture, Exercise	Quiz, Exams, Assignments

C. Course Content

No	List of Topics	Contact Hours
1.	OO basics, Dynamic dispatch, Encapsulation; example code	4
2.	Unit testing, continuous integration - Specification and unit testing - Test case design - IDEs, Build system, Continuous Integration, Librarie	4
3.	UML and Patterns - Design for Reuse: Inheritance and Composition - Responsibility assignment	4
4.	Refactoring and Anti-patterns	4
5.	Introduction to GUIs	4
6.	Reactive Programming	4
7.	Local Parallelism	4
8.	Libraries and Frameworks	4
9.	API Design	4
10.	Organizing Systems at Scale: Modules	4
11.	DevOps	4
12.	Static vs Dynamic Typing, Static Analysis	4
13.	AI-Assisted Code Generation and Refactoring • AI-Powered Code Generation • AI-Assisted Code Refactoring • Bug Detection and Auto-Fix Using AI	8
Total		**60**

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Quizzes and Assignments	2-15	20
2.	Projects	2-14	10
4.	Lab Assessment	2-14	10
5.	Midterm	7	20
6.	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	• Larman, C. (2004). *Applying UML and patterns: An introduction to object-oriented analysis and design and iterative development* (3rd ed.). Prentice Hall. ISBN 978-0131489066. • Bloch, J. (2017). *Effective Java* (3rd ed.). Addison-Wesley. ISBN 978-0134685991.
Supportive References	• Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley Professional. • Sestoft, P. (2016). *Java precisely* (3rd ed.). MIT Press. • Eck, D. J. (2019). *Introduction to programming using Java* (8th ed.). Retrieved from http://math.hws.edu/javanotes/ • Litvin, M., & Litvin, G. (2011). *Blue Pelican Java*. Retrieved from https://www.bluepelicanjava.com/ • Crockford, D. (2008). *JavaScript: The good parts*. O'Reilly Media. • Cherny, B. (2019). *Programming TypeScript: Making your JavaScript applications scale*. O'Reilly Media. • Thomas, D., & Hunt, A. (2019). *The pragmatic programmer: Your journey to mastery, (2nd ed.)*. Addison-Wesley Professional.
Electronic Materials	• Carnegie Mellon University. (2024). *Software engineering course materials: 17-214 Principles of software*

	construction. Retrieved from https://cmu-17-214.github.io/s2024/index.html
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 17TH MEETING FOR THE ACADEMIC YEAR 1446H**
DATE	**22/04/2025**

COUNCIL /COMMITTEE	**Computer Science and Artificial Intelligence Department Council**
REFERENCE NO.	**THE 16TH MEETING FOR THE ACADEMIC YEAR 1446H**
DATE	**22/04/2025**