Course Specification
(Bachelor)

Course Title:	Object-oriented Application Development
Course Code:	SE1101
Program:	BSc in Software Engineering BSc in Human Computer Interaction
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: **BSc in Software Engineering - (1st year/ 2nd level)** **BSc in Human Computer Interaction - (2nd year/ 3rd level)**					
4. Course General Description:					
This course provides students with a solid foundation in object-oriented programming (OOP) principles and practices. It emphasizes designing, developing, and implementing applications using object-oriented techniques. Students will learn about classes, objects, inheritance, polymorphism, and encapsulation, as well as introduction to design principle. Practical lab sessions and project will enable students to apply their knowledge in developing real-world applications.					
5. Pre-requirements for this course (if any)**:**					
BSc in Software Engineering: CS1701 - Computer Programming in Java **BSc in Human Computer Interaction:** CS1011 - Computer Programming in Python					
6. Co-requisites for this course (if any)**:**					
N/A					
7. Course Main Objective(s):					
By the end of the course, participants should be able to: • Understand and apply object-oriented principles in real-world applications. • Design and develop robust, reusable, and maintainable software. • Build a fully functional application using object-oriented principles.					

2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	٦٠	100%

No	Mode of Instruction	Contact Hours	Percentage
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**	
4.	**Tutorial**	
5.	**Others (specify)**	
Total		٦٠

B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods (BSc in Software Engineering)

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
1.0	**Knowledge and understanding**			
1.1	Understand the principles of object-oriented programming and how they improve software design.	K1	Lecture, Exercise	Quiz, Exams, Assignments
1.2	Understand concepts of inheritance, polymorphism, and encapsulation in application development.	K2	Lecture, Exercise	Quiz, Exams, Assignments
2.0	**Skills**			
2.1	Design and implement object-oriented applications using classes and objects.	S1	Lecture, Exercise	Quiz, Exams, Labs, Project
2.2	Develop and deploy object-oriented applications using an	S2	Lecture, Exercise	Quiz, Exams, Labs, Project

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
	integrated development environment (IDE).			
2.3	Write clean, maintainable, and reusable code.	S2	Lecture, Exercise	Quiz, Exams, Labs, Project
3.0	**Values, autonomy, and responsibility**			
3.1	Collaborate with peers on Application Development	V2	Lecture, Exercise	Quiz, Exams, Labs, Project

Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods (BSc in Human Computer Interaction)

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
1.0	**Knowledge and understanding**			
1.1	Understand the principles of object-oriented programming and how they improve software design.	K1	Lecture, Exercise	Quiz, Exams, Assignments
1.2	Understand concepts of inheritance, polymorphism, and encapsulation in application development.	K2	Lecture, Exercise	Quiz, Exams, Assignments
2.0	**Skills**			
2.1	Design and implement object-oriented applications using classes and objects.	S1	Lecture, Exercise	Quiz, Exams, Labs, Project
2.2	Develop and deploy object-oriented applications using an integrated development environment (IDE).	S4	Lecture, Exercise	Quiz, Exams, Labs, Project
2.3	Write clean, maintainable, and reusable code.	S4	Lecture, Exercise	Quiz, Exams, Labs, Project
3.0	**Values, autonomy, and responsibility**			
3.1	Collaborate with peers on Application Development	S5	Lecture, Exercise	Quiz, Exams, Labs, Project

C. Course Content

No	List of Topics	Contact Hours
1.	Introduction to Object-Oriented Programming Procedural vs. Object-Oriented Programming Core OOP principles: • Abstraction • Encapsulation • Inheritance • Polymorphism Real-world examples of OOP concepts	**4**
2.	Classes and Objects • Defining and creating classes and objects • Instance variables and methods • This keyword • Constructors and their usage • Method overloading	4
3.	Encapsulation and Access Modifiers • Importance of encapsulation • Access modifiers: public, private, protected, default • Getters and setters for data hiding • Writing modular and secure code	**8**
4.	Arrays, Lists, and Collections • Revisiting arrays in Java • Introduction to Array List and basic collection classes • Iterating over collections using loops	**8**

5.	Inheritance and Polymorphism
	• Introduction to inheritance
	• The extends keyword
	• Method overriding
	• The super keyword
	• Polymorphism basics:
	  • Static vs. dynamic binding
	  • Upcasting and down casting
		4

6.	Static Members and Keywords • Static variables and methods • The final keyword for constants and methods • Static blocks and their use cases	**4**
7.	Exception Handling • What are the exceptions?	**4**

	• Try-catch-finally blocks • Common exceptions (NullPointerException, ArrayIndexOutOfBoundsException) • Throwing and propagating exceptions • Writing custom exception classes	
8.	Java I/O Basics • Reading input using Scanner • Writing output to the console • File handling basics: • Reading from and writing to text files • Using FileReader and FileWriter	**8**
9.	Interfaces and Abstract Classes • Introduction to interfaces • Difference between interfaces and classes • Implementing multiple interfaces • Abstract classes and methods • Comparing abstract classes and interfaces	**4**
10.	Basic GUI Development with JavaFX • Introduction to JavaFX • Creating a simple graphical user interface • Handling user events (e.g., button clicks)	**4**
11.	Project Development • Small group or individual projects to consolidate learning: • Design a basic Java application using OOP principles. • Example projects: • Library Management System • Student Record Management • A simple console-based game (e.g., Tic-Tac-Toe)	**4**
12.	Advanced Topics: e.g Code Refactoring and SOLID or Debugging	**4**
Total		**60**

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Quizzes and Assignments	3-14	20
2.	Projects	15	10
4.	Lab Assessment	2-14	10
5.	Midterm	7	20

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
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	• Liang, Y. D. (2023). *Introduction to Java programming and data structures.* • Nelson, J. (2023). *Java illuminated.* • Lassoff, M. (2017). *Java programming for beginners: Learn the basics of Java programming*.
Supportive References	• Sharan, K., & Spath, P. (2022). *Learn JavaFX 17: Building user experience and interfaces with Java* (2nd ed.).
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 17TH MEETING FOR THE ACADEMIC YEAR 1446H
	22/04/2025