هيئة تقويم التعليم والتدريب
Education & Training Evaluation Commission

2023
TP-153

Course Specification

(Bachelor)

Course Title:	Software Design and Development 1
Course Code:	SE1251
Program:	Software Engineering
Department:	Software Engineering
College:	Computing and Information
Institution:	Al-Baha University
Version:	V1.0
Last Revision Date:	24-4-2024

Table of Contents

A. General information about the course:

1. Course Identification

1. Credit hours: (4)

2. Course type

A.	☐University	☐College	☒ Department	☐Track	☐Others
B.	☒ Required		☐Elective		

3. Level/year at which this course is offered: (6th /2nd year)

4. Course general Description:

This course provides a broad and in depth understanding of architectural and design concepts, principles, strategies, styles, notations, and methods. The course, in addition, provides applied knowledge of software design principles such as encapsulation, polymorphism, reusability, flexibility, portability and robustness. Furthermore, it highlights the design and programming paradigms such as domain-specific, pattern-oriented, component-oriented, aspect-oriented, object-oriented paradigms. However, the focus will be on object-orientation paradigm.

Finally, software construction from various software design representations will be introduced.

5. Pre-requirements for this course (if any): SE1002 Requirements Engineering 1

6. Pre-requirements for this course (if any)**: None**

7. Course Main Objective(s):

The main aim of this course is to teach the students the concept of software deigns. This includes design principles and methods. In addition, it provides students with needed knowledge of design and programming paradigms. Furthermore, it helps the students to acquire knowledge and skills in software construction from software design representations.

2. Teaching mode (mark all that apply)

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

3. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	**Lectures**	33
2.	**Laboratory/Studio**	22
3.	**Field**	
4.	**Tutorial**	
5.	**Others (specify)**	
Total		55

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

Code	Course Learning Outcomes	Code of CLOs aligned with program	Teaching Strategies	Assessment Methods
1.0	**Knowledge and understanding**			
1.1	Recognize the concepts software Design and construction	**K1**	Lecture, exercise	Quiz, final exam ,assignments
1.2	Define knowledge of key design principles, paradigms, and other design considerations.	**K2**	Lecture, exercise	Quiz, final exam , ,assignments
1.3	describe the design methods and OO design methods	**K3**	Lecture, exercise	Quiz, final exams, ,assignments
2.0	Skills			
2.1	Design a software using	**S1**	Lecture, Group discussion,	Final exams, assignments,

Code	Course Learning Outcomes	Code of CLOs aligned with program	Teaching Strategies	Assessment Methods
	design principles		tutorials	project
2.2	Apply design methods on software requirements for considerable software size	**S2**	Lecture Group discussion, tutorials	Final exams, assignments, project
2.3	Construct a code stubs from a various software design representations	**S3**	Lecture, Group discussion, tutorials	Final exams, assignments, project
3.0	programming paradigm according to a			
3.1	Demonstrate responsibility, ethics, and effective teamwork	**V1**	Project	Assignment, project

C. Course Content

No	List of Topics	Contact Hours
1.	Introduction to software lifecycle and its models and phases such as waterfall, spiral, prototyping, Agile models.	6
2.	Software design principles: Flexibility, Reusability, Encapsulation, Modularity, Robustness, Portability, Abstraction.	6
3	Design methods: Functional decomposition, Data Flow, and Design Based on Data Structure	6
4	Domain-specific, pattern-oriented, component-oriented, aspect-oriented Object-orientation design (Booch, Fusion, and RUP)	3
5	UML Structure Diagrams: Class Diagram, Component Diagram, Deployment Diagram, Object Diagram, Package Diagram, Composite Diagram	6
6	UML Behavioral Diagrams: Use Case Diagram, Activity Diagram, State Machine Diagram, Sequence Diagram, Communication Diagram, Interaction Diagram	**6**
Total		**33**

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Project	10	20%
2.	Lab exam	11	20%

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
3.	Midterm Exam	5	20%
4	Final Exam	12	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	**Gomaa, Hassan. Software modeling and design: UML, use cases, patterns, and software architectures. Cambridge University Press, 2011.** **Van Vliet, Hans, Hans Van Vliet, and J. C. Van Vliet. Software engineering: principles and practice. Vol. 13. Hoboken, NJ: John Wiley & Sons, 2008**
Supportive References	**N/A**
Electronic Materials	**N/A**
Other Learning Materials	**N/A**

2. Required Facilities and equipment

Items	Resources
facilities (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	Classroom and Blackboard
Technology equipment (projector, smart board, software)	Data show and software
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 • Faculty • Peer Reviewers • Program Leader Course Coordinator	•Surveys (indirect). •Direct feedback from students. •Course evaluation by Peer Reviewers (indirect). •Class visit by Program Leader (indirect) Comprehensive Course report (where we can find information about teaching difficulties and action plan, …)
Effectiveness of	• Students • Faculty	• Surveys (indirect).

Assessment Areas/Issues	Assessor	Assessment Methods
Students assessment	• Peer Reviewers • Program Leader • Exam Evaluation Committee Course Coordinator	• Direct feedback from students. • Course evaluation by Peer Reviewers (indirect). • Class visit by Program Leader (indirect) Exam evaluation by the Exam Evaluation Committee (indirect)
Quality of learning resources	• Students • Faculty • Peer Reviewers Course Coordinator	• Surveys (indirect) • Course evaluation by Peer Reviewers (indirect). Comprehensive Course report (where we can find information about difficulties and challenges about learning resources as well as consequences and action plan, …)
The extent to which CLOs have been achieved	• Faculty • Program Leader Course Coordinator	• Student Results (direct) Comprehensive Course report (where we can find the CLO assessment results)
Other	None	None

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

Assessment Methods (Direct, Indirect)

G. Specification Approval

COUNCIL /COMMITTEE	**Curriculum Committee**
REFERENCE NO.	
DATE	**28 April 2024**