هيئة تقويم التعليم والتدريب
Education & Training Evaluation Commission

Course Specification
(Bachelor)

Course Title: programming 2

Course Code: CS1251

Program: Computer Science

Department: Computer Science & Engineering

College: Computer Science and Information Technology

Institution: Al Baha University

Version: v1.0

/Last Revision Date: 7/10/2023

Table of Contents

A. General information about the course:

1. Course Identification

1. Credit hours: (4 hours)

2. Course type

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

3. Level/year at which this course is offered: (1/2)

4. Course general Description:

Programming 2 course is designed to introduce the concepts and principles of object-oriented programming (OOP). Students will learn how to design, implement, and maintain software systems using OOP techniques such as class design, object instantiation and method implementation. The course covers the fundamental principles of OOP, such as encapsulation, inheritance, polymorphism, and abstraction. These techniques and principles will be introduced with various practical examples. Upon successfully completing the course, students will be able to apply OOP principles to design and implement robust, scalable, and maintainable software systems.

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

Programming 1 (CS1005)

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

Programming 1 (CS1005)

7. Course Main Objective(s):

- **Recall the OOP principles, and their benefits and uses.**
- **Write programs that define classes and use objects of those classes.**
- **Design classes following the principles of class abstraction, encapsulation and information hiding.**
- **Write programs that utilize inheritance and polymorphism.**
- **Apply exception-handling techniques.**
- **Write programs that read, write and modify text files.**

- **Write programs that utilize inheritance and polymorphism.**
- **Demonstrate an understanding of abstract classes and interfaces.**
- **Use generic classes and interfaces.**
- **Code programs that use part of the language's API (e.g. display simple graphical components, use common methods for manipulating built-in data structures or get a taste of event-driven programming).**
- **Work both independently and collaboratively.**

2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	33	60%
2	E-learning		
3	Hybrid • Traditional classroom • E-learning		
4	Distance learning	22	40%

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		

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	Recall the OOP principles, and their benefits and uses.	K1	Lectures	• Midterm • Final Exam

Code	Course Learning Outcomes	Code of CLOs aligned with program	Teaching Strategies	Assessment Methods
2.0	**Skills**			
2.1	Write programs that define classes and use objects of those classes.	S1	• Tutorials • Lectures • Case study project	• Midterm • Project (rubric) • Final Exam
2.2	Design classes following the principles of class abstraction, encapsulation and information hiding.	S1	• Tutorials • Lectures • Case study project	• Midterm • Project (rubric) • Final Exam
2.3	Write programs that utilize inheritance and polymorphism.	S1	• Tutorials • Lectures • Case study Assignment	• Assignment (rubric) • Midterm exam Final Exam
2.4	Apply exception-handling techniques.	S1	• Tutorials • Lectures • Problem-based learning Assignment	• Assignment (rubric) Final Exam
2.5	Write programs that read, write and modify text files.	S1	• Tutorials • Lectures • Case study project	• Project (rubric) Final Exam
2.6	Demonstrate an understanding of abstract classes and interfaces.	S1	• Tutorials • Lectures • Case study Assignment	• Assignment (rubric) Final Exam
2.7	Use generic classes and interfaces.	S3	• Tutorials • Lectures • Problem-based learning project	• Project (rubric) Final Exam
2.8	Code programs that use part of the language's API	S3	• Tutorials • Lectures • Case study Assignment	• Assignment (rubric) Final Exam
3.0	**Values, autonomy, and responsibility**			
3.1	Work both independently and collaboratively.	V1	• Project • Assignments	• Rubric

Code	Course Learning Outcomes	Code of CLOs aligned with program	Teaching Strategies	Assessment Methods
…				

C. Course Content

No	List of Topics	Contact Hours
1.	Introduction	1
2.	Objects and Classes	9
3	Object-oriented thinking (class abstraction and encapsulation, and class relationships)	5
4	Inheritance and Polymorphism	10
5	Exception handling	4
6	Text I/O	6
7	Abstract classes and Interfaces	5
8	Generics	5
9	Built-in classes (e.g. GUI, data structures, event-driven programming)	10
Total		**55**

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Midterm exam	5	20%
2.	Assignments/Discussions	Periodically	20%
3.	Project/Presentation	10-11	20%
…	Final exam	13	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	"Introduction to Java Programming, Comprehensive" by D. Liang, Pearson, 2015.
Supportive References	• "Introduction to Java Programming and Data Structures, Comprehensive Version" by D. Liang, Pearson, 2019. • Introduction to Programming Using Java" by D. J. Eck, 2019. • "Java How to Program" by P. Deitel, H. Deitel, Pearson, 2017.
Electronic Materials	• Access to the Saudi Digital Library (SDL).

	• Using the learning management system of the university – Rafid System (https://lms.bu.edu.sa/).
	 • For API:
	     o https://docs.oracle.com/javase/8/docs/api/
	 • For tutorials:
	     o Java:
	       ▪ https://docs.oracle.com/javase/tutorial/
	       ▪ https://www.tutorialspoint.com/java/
	       ▪ https://www.javatpoint.com/java-tutorial
	       ▪ https://www.codecademy.com/learn/learn-java
	 https://www.udemy.com/java-tutorial/

Other Learning Materials	• Sololearn (mobile app) or similar

2. Required Facilities and equipment

Items	Resources
facilities (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	All the lectures should be in a well-prepared lab that can accommodate 25 students at most.
Technology equipment (projector, smart board, software)	• A digital image projection system with a connection to a computer. • High-speed Internet connection. • An instructor computer station. An application to manage labs and learning sessions (e.g. NetSupport School).
Other equipment (depending on the nature of the specialty)	**If we have disability students will ask for different equipment's.**

F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of teaching		
Effectiveness of Students assessment	• Students • Exam Evaluation Committee Course Coordinator	• Survey (indirect) • Exam Review (direct) Review of course file (direct)
Quality of learning resources	• Faculty Students	Survey (indirect)
The extent to which CLOs have been achieved	• Faculty Program Leaders or Course Coordinator	• Exams (direct) Exit Exams (direct)
Other		

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

Assessment Methods (Direct, Indirect)

G. Specification Approval

COUNCIL /COMMITTEE	DEPARTMENT COUNCIL MEETING
REFERENCE NO.	8TH COUNCIL MEETING
DATE	