هيئة تقويم التعليم والتدريب
Education & Training Evaluation Commission
اعتماد
NCAAA
2024
TP-153

Course Specification
(Bachelor)

Course Title:	Object-Oriented Programming
Course Code:	212CCS-4
Program:	Bachelor of Science in Computer Science
Department:	Department of Computer Science
College:	College of Computer Science and Information Systems
Institution:	Najran University
Version:	2.0
Last Revision Date:	15 February 2023

Table of Contents

A. General information about the course:

1. Course Identification

1. Credit hours: (4)
4 (3, 2, 1) [Theory, Lab, Tutorial]

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

3. Level/year at which this course is offered: (Level 4/Year 2)

4. Course General Description:
The course introduces the fundamental concepts of imperative programming languages. Topics include data types, control structures, functions, arrays, files, exception handling, and the mechanics of running, testing, and debugging of processing programs. This course also covers the basic concepts for software design and reuse. One C, C++, Python, or Java programming language can be used as a representative imperative language of this course.

5. Pre-requirements for this course (if any):
211CCS-4

6. Co-requisites for this course (if any):
None

7. Course Main Objective(s):

2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	90	100%
2	E-learning		
3	Hybrid • Traditional classroom • E-learning		

No	Mode of Instruction	Contact Hours	Percentage
4	Distance learning		

3. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	**Lectures**	45
2.	**Laboratory/Studio**	30
3.	**Field**	
4.	**Tutorial**	15
5.	**Others (specify)**	
Total		90

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 the essential principles and concepts of object-oriented programming and their appropriateness in solving computational problems.	K_1, K_3	Cooperative Learning, Inquiry-based instruction, class discussion, problem-based learning	Quiz-1, Midterm, Assignment-1, Lab Activities, Final Theory Exam
2.0	**Skills**			
2.1	Apply object-oriented programming styles which impact on developing and maintaining GUI applications.	S_1, S_5	Cooperative Learning, Inquiry-based instruction, class discussion, formative assessment, problem-based learning	Assignment-4, Project-2, Lab Activities, Final Lab Exam, Final Theory Exam
2.2	Demonstrate the ability to explain, testing, correct and debugging processing object oriented programs	S_2, S_4	Cooperative Learning, Inquiry-based instruction, class discussion, formative assessment	Assignment-5, Project-3, Lab Activities, Final Lab Exam, Final Theory Exam
2.3	Design programs utilizing the principles of object-oriented to solve simple	S_2, S_5	Cooperative Learning, Inquiry-based instruction, class discussion, problem-based learning	Assignment-2, Project-1, Lab Activities, Final Lab Exam, Final Theory Exam

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
	computational problems			
2.4	Implement object-oriented principles to effectively and efficiently solve computational problems involving multiple objects	S_5	Cooperative Learning, Inquiry-based instruction, class discussion, formative assessment, problem-based learning	Quiz-2, Assignment-3 Midterm, Lab Activities Exam, Final Lab Exam, Final Theory Exam
3.0	**Values, autonomy, and responsibility**			
3.1	Write object-oriented programs with collaboration and teamwork in mind	C_1, C_2	Cooperative Learning, Inquiry-based instruction, Project-based learning, formative assessment	Assignment-5, Lab Activities, Project

C. Course Content

No	List of Topics	Contact Hours
1.	Revision to 113CSS-4	6
2.	Introduction to Object-Oriented Programming	6
3.	Methods	6
4.	Objects and Classes	6
5.	Object-Oriented Thinking – Class Abstraction and Encapsulation	6
6.	Inheritance and Polymorphism	12
7	Exception Handling and Text I/O	6
8	Abstract Classes and Interfaces	6
9	JavaFX Basics	6
10	Event-Driven Programming and Animation	6
11	Event-Driven Programming and Animation	6
12	JavaFX UI Controls and Multimedia	12
13	Revision	6
Total		**90**

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Quiz-1	5	5%
2.	Quiz-2	13	5%
3.	Assignment	Every week	10%
4.	Midterm or mini project (presentation)	10	20%
5.	Lab Activities	Every week	10%
6.	Final Lab Exam	15	10%
7.	Final Theory Exam	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	Introduction to Java Programming Comprehensive Version Tenth Edition, by Y. Daniel Liang, ISBN-13: 978-0133761313 ISBN-10: 0133761312
Supportive References	1. Herbert Schildt The Complete Reference, JAVA 2, 9th Edition, 2014, McGraw Hill Publishing Company Ltd. 2. Harvey M. Deitel and Paul J. Deitel, Java, How to Program: JavaTM, 9th Edition, 2011, Prentice Hall. 3. Thomas Wu, An Introduction to Object-Oriented Programming with JAVA, 5th Edition, 2009, McGraw-Hill. 4. Bruce Eckel, Thinking in Java, 2nd Edition, Prentice Hall
Electronic Materials	
Other Learning Materials	

2. Required Facilities and equipment

Items	Resources
facilities (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	Classrooms to accommodate 50 students per classroom with desks and chairs, and labs to accommodate 25 students per lab with advanced computers.
Technology equipment (projector, smart board, software)	Data show, stationaries, smart board, suitable IDE (Netbeans and Eclipse)
Other equipment (depending on the nature of the specialty)	ACs for labs and classrooms, black curtains

F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Collecting students' suggestions to facilitate more during the class.	Students	Verbal discussion
Student's questionnaire once during the semester about course learning outcomes.	Students	Indirect Survey
Achievement percentage of course learning outcomes, direct evaluation using CLO assessment sheet	Course Instructor	Direct evaluation using CLO achievement calculation
Teaching strategies	Quality unit	Indirect
Assessment methods	Quality unit	Indirect
Instructor performance	Quality unit	Indirect
Course content	Quality unit	Indirect

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

Assessment Methods (Direct, Indirect)

G. Specification Approval

COUNCIL /COMMITTEE	Computer Science Departmental Council
REFERENCE NO.	14440203-0185-00002
DATE	1st Sep, 2022