



Course Specifications

Course Title:	Object Oriented Programming
Course Code:	2120CSS-4
Program:	BSc in Computer Science
Department:	Computer Science
College:	Computer Science and Information Systems
Institution:	Najran University

Table of Contents

A. Course Identification.....	3
6. Mode of Instruction (mark all that apply)	3
B. Course Objectives and Learning Outcomes.....	3
1. Course Description	3
2. Course Main Objective.....	3
3. Course Learning Outcomes	4
C. Course Content.....	4
D. Teaching and Assessment	4
1. Alignment of Course Learning Outcomes with Teaching Strategies and Assessment Methods.....	4
2. Assessment Tasks for Students	5
E. Student Academic Counseling and Support	6
F. Learning Resources and Facilities.....	6
1.Learning Resources	6
2. Facilities Required.....	6
G. Course Quality Evaluation	7
H. Specification Approval Data	7

A. Course Identification

1. Credit hours:	4(3, 2, 1)
2. Course type	
a.	University <input type="checkbox"/> College <input checked="" type="checkbox"/> Department <input type="checkbox"/> Others <input type="checkbox"/>
b.	Required <input checked="" type="checkbox"/> Elective <input type="checkbox"/>
3. Level/year at which this course is offered:	Year 2 / Level 5
4. Pre-requisites for this course (if any):	111CSS-4
5. Co-requisites for this course (if any):	None

6. Mode of Instruction (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	60	100%
2	Blended		
3	E-learning		
4	Distance learning		
5	Other		

7. Contact Hours (based on academic semester)

No	Activity	Contact Hours
1	Lecture	30
2	Laboratory/Studio	20
3	Tutorial	10
4	Others (specify)	
	Total	60

B. Course Objectives and Learning Outcomes

1. Course 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 languages can be used a representative imperative language of this course.
2. Course Main Objective

3. Course Learning Outcomes

CLOs		Aligned PLOs
1	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
1.2		
1.3		
1...		
2	Skills :	
2.1	Apply object oriented programming styles which impact on developing and maintaining GUI applications.	S ₁ , S ₅
2.2	Demonstrate the ability in explaining, testing, correcting and debugging processing object oriented programs	S ₂ , S ₄
2.3	Design programs utilizing the principles of object oriented to solve simple computational problems	S ₂ , S ₅
2...	Implement object oriented principles to effectively and efficiently solve computational problems involving multiple objects	S ₅
3	Values:	
3.1	Write object oriented programs with collaboration and team work in mind	C ₁ , C ₂
3.2		
3.3		
3...		

C. Course Content

No	List of Topics	Contact Hours
1	Revision to 113CSS-4	4
2	Introduction to Object Oriented Programming	4
3	Objects and Classes	8
4	Object Oriented Thinking – Class Abstraction and Encapsulation	4
5	Inheritance and Polymorphism	8
6	Exception Handling and Text I/O	4
7	Abstract Classes and Interfaces	8
8	JavaFX Basics	4
9	Event Driven Programming and Animation	4
10	JavaFX UI Controls and Multimedia	8
11	Revision	4
Total		60

D. Teaching and Assessment

1. Alignment of Course Learning Outcomes with Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
1.0	Knowledge and Understanding		

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
1.1	Explain the essential principles and concepts of object-oriented programming and their appropriateness in solving computational problems	Cooperative Learning, Inquiry-based instruction, class-discussion, problem-based learning	Quiz-1, Midterm, Assignment-1, Lab Activities, Final Theory Exam
1.2			
...			
2.0	Skills		
2.1	CLO#4: Apply object oriented programming styles which impact on developing and maintaining GUI applications.	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	CLO#5: Demonstrate the ability in explaining, testing, correcting and debugging processing object oriented programs	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 computational problems	Cooperative Learning, Inquiry-based instruction, class-discussion, problem-based learning	Assignment-2, Project-1, Lab Activities, Final Lab Exam, Final Theory Exam
2.4	Implement object oriented principles to effectively and efficiently solve computational problems involving multiple objects	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		
3.1	Write object oriented programs with collaboration and team work in mind	Cooperative Learning, Inquiry-based instruction, Project-based learning, formative assessment	Assignment-6, Lab Activities, Project-1, Project-2, Project-3
...			

2. Assessment Tasks for Students

#	Assessment task*	Week Due	Percentage of Total Assessment Score
1	Quiz-1	2	5%
2	Quiz-2	7	5%
3	Assignment or mini project (presentation)	Every week	10%
4	Midterm	5	20%
6	Lab Activities	Every week	10%

#	Assessment task*	Week Due	Percentage of Total Assessment Score
7	Final Lab Exam	11	10%
8	Final Theory Exam	12 or 13	40%

*Assessment task (i.e., written test, oral test, oral presentation, group project, essay, etc.)

E. Student Academic Counseling and Support

Arrangements for availability of faculty and teaching staff for individual student consultations and academic advice:

Specifying office hours, forming discussion groups using social media (e.g. Facebook and Twitter) and messenger application (e.g. WhatsApp and Telegram)

F. Learning Resources and Facilities

1. Learning Resources

Required Textbooks	Introduction to Java Programming Comprehensive Version Tenth Edition, by Y. Daniel Liang, ISBN-13: 978-0133761313 ISBN-10: 0133761312
Essential References Materials	<ol style="list-style-type: none"> Herbert Schildt The Complete Reference, JAVA 2, 9th Edition, 2014, McGraw Hill Publishing Company Ltd. Harvey M. Deitel and Paul J. Deitel, Java, How to Program: Java™, 9th Edition, 2011, Prentice Hall. Thomas Wu, An Introduction to Object-Oriented Programming with JAVA, 5th Edition, 2009, McGraw-Hill. Bruce Eckel, Thinking in Java, 2nd Edition, Prentice Hall
Electronic Materials	
Other Learning Materials	

2. Facilities Required

Item	Resources
Accommodation (Classrooms, laboratories, demonstration rooms/labs, etc.)	Classrooms to accommodate 50 students per classroom with desks and chairs, labs to accommodate 25 students per lab with advanced computers.
Technology Resources (AV, data show, Smart Board, software, etc.)	Data show, stationaries, smart board, suitable IDE (Netbeans and Eclipse)
Other Resources (Specify, e.g. if specific laboratory equipment is required, list requirements or attach a list)	ACs for labs and classrooms, black curtains

G. Course Quality Evaluation

Evaluation Areas/Issues	Evaluators	Evaluation Methods
Effectiveness of teaching	Students, faculty and peer review	Indirect (questionnaires and interviews)
Assessment	Faculty and student	Direct and indirect (exams, quizzes, lab works and questionnaires)
Achievement of course learning outcome	Faculty	Direct and indirect (exams, quizzes, lab works and questionnaires)
Skills of analysis and discussion	Faculty	Project presentation
Quality of learning resources	Faculty, student, head of department	Written exam
Professional skills	Faculty	Practical exam
Ability to work in groups	Faculty and students	Assignments

Evaluation areas (e.g., Effectiveness of teaching and assessment, Extent of achievement of course learning outcomes, Quality of learning resources, etc.)

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

Assessment Methods (Direct, Indirect)

H. Specification Approval Data

Council / Committee	Computer Science Departmental Council
Reference No.	14440203-0185-00002
Date	1st Sep, 2022