پوهنتون کاردان
KARDAN UNIVERSITY

Faculty of Computer Science

Department of Information Technology

Course Title:	Object Oriented Programming
Code:	**CS-107**
Coordinating Faculty/ Department:	Computer Science
Credits:	4 Credits
Pre-requisites :	Programming Language Concepts.
Lecturer Name and Contact:	Fazal Maula f.maula@kardan.edu.af

General Course Information

This course teaches object oriented programming to those who have learnt basic programming concepts and are ready to learn in-depth programming. It focuses on object-oriented programming using Java. The main concepts discussed are: Objects, Data Abstraction, Data Encapsulation, Polymorphism, and Inheritance. We teach the Java language constructors and objects that are used to implement these concepts. For example, Classes, methods, Object and constructors.

COURSE OBECTIVES

By the end of this course students will be able to:

1. Develop a working knowledge of the concepts of Object Oriented Programming (OOP);
2. Apply the concepts of OOP software design principles;
3. Apply java concepts, such as inheritance, data abstraction, and polymorphism;

4. Utilize the OOP concept to design medium-complexity Graphical User Interface Applications.
5. Skillfully apply essential programming concepts in the completion of lab and project assignments.

COURSE Learning Outcome

- Read and understand Java-based software code of medium-to-high complexity.
- Use standard and third party Java's API's when writing applications.
- Understand the basic principles of creating Java applications with graphical user interface (GUI).
- Understand the features of Java supporting object oriented programming
- Understand how to produce object-oriented software using Java
- Understand how to apply the major object-oriented concepts such as encapsulation, inheritance and polymorphism to implement object oriented programs in Java.
- Apply the abovementioned points to design, implement, appropriately document and test a Java application of medium complexity, consisting of multiple classes.

LEARNING REFERECNCES

1. Textbooks:

- "Java How to Program" by Paul Deitel (10th Edition or 11th)
- Vaskaran Sarcar - Interactive Object-Oriented Programming in Java (2nd Ed) (2020, Apress)
- Y. Daniel Liang - Intro to Java Programming, Comprehensive Version (2014, Prentice Hall)
- Y. Daniel Liang - Introduction to Java Programming, Brief Version (2012, Prentice Hall)
- Object Oriented Programming using Java by Simon Kendal

2. Websites

- JavaTpoint website (www.javatpoint.com)
- Study tonight website (www.studytonight.com)

TEACHING METHODS:

- **Lecture:**
  Oral presntation focuses on explaining topics, ideas and theoris
- **Experimental Techniques:**
  Case studiess
- **Computer Based practices:**
  SIMULATIONS

Class Room Etiquette:

Class room courtesy is expected of all students all the time. Following is general guideline for the class room conduct:

1. Refrain from chatting with fellow students.
2. Arrive on time. If you come late please be courteous to other students
3. Do not get up and walk out halfway through class.
4. Do not cut the instructor off at the end of class.
5. All cell phones must be turned off during class and examination.
6. Eating is not permitted.
7. Furthermore, please be cognizant of the learning process and purpose of you being in the class room and extend same respect to other students.

POLICIES & GUIDELINES

Guidelines for Format of Assignments: (General Recommendations)

- Students must submit their assignment in hard form within deadline.
- There will be a project which should be implemented by group of students in duration of the semester.
- There is no extension of time for the deadlines

Note:
All papers and course work submitted will be subject to plagiarism verification by software.

8.COURSE SCHEDULE / CONTENTS

Week No.	Topics for Discussion	Book Chapter #	Suggested Activity	Topic Outcomes
1	**Lecture 1: Introduction to Java Programming** 1) Computer Programming 2) Text Editor 3) Compiler 4) Interpreter 5) basics for Java language 6) Java History 7) Java Standard Edition 8) Java Micro Edition 9) Java Enterprise Edition 10) JavaFX	Chapter 1	Introduction of the students, Summarizing	Undrestand the basics of the JAVA Languages
2	**Lecture 2:First Java Program & Code Description**	Chapter 1	Practical class work, case studies, **Class**	Understand the basic approaches to the design of

	1) Netbeans 2) Code Description 3) Class Name convention 4) Good Programming Tips 5) Getting input from user 6) Arithmetic Operators 7) Class Activity		**Activity**	software applications.
3	**Lecture 3: Java Programming basics** 1) Java Variables 2) Identifiers 3) The general rules for naming variables 4) Java Data Types 5) Primitive Data Types 6) Numbers 7) Conditional Statements	Chapter 4	Case study, **Assignment** 01	Improve command on conditional structures of a program
4	**Lecture 4: If-else Statement** **&** **Switch Statement** 1) if statement 2) if-else statement 3) if-else-if ladder 4) nested if statement 5) Java Switch Statement 6) Points to Remember in switch 7) Class Task	Chapter 7	Practical work, Examples & different practical tasks.	To understand and implement arrays that has a crucial part in the software application deveopment.
5	**Lecture 5: Part I: Loops in Java:** **For loop** 1) Java Simple for Loop 2) Syntax 3) Flowchart 4) Programs 5) Class practice	Chapter 3	Case Study, Practical, **class Activity**	To undrestand the importance of class objects and accessibility to the class properties through the objects
6	**Lecture 5: Part II: Loops in Java:** **While loop** **&** **Do-While loop**	Chapter 6	Lecture, Summarizing, Practical Class Activity	Manipulating the java program tasks into different

	1) While loop 2) Syntax 3) The different parts of while loop 4) Flowchart of Java While Loop 5) Examples 6) Java Infinitive While Loop 7) do-while loop 8) Syntax 9) The different parts of do-while loop 10) Flowchart of do-while loop 11) Examples 12) Java Infinitive do-while Loop 13) Java Break Statement 14) Java Continue Statement 15) Questions and answers		**Assignment 02**	
7	**Lecture 6: Java Arrays** 1) Introduction to arrays 2) Access the Elements of an Array 3) Change an Array Element 4) Array Length 5) Traversing array using For loop 6) For-each Loop for Java Array 7) Syntax 8) Examples	Chapter 6	Lecture, Class activity, **use cases**	Undrestanding of different variable usages and access levels.
8	**Lecture 7: Exception Handling** 1) What is an Exception 2) Types of Exception 3) Checked exceptions 4) Unchecked Exceptions 5) Stack trace 6) Arithmetic Exception 7) Syntax 8) Handling an Arithmetic Exception 9) Input Mismatch Exception 10) Handling input Mismatch Exception 11) Handling Multiple exceptions 12) Catching Multiple Exceptions 13) Finally Block 14) The throw keyword	Chapter 9	Case Study, **Assignment 03**	Undrestand the reusibility and abstraction of the classess
9	**Lecture 8: Graphical User Interface (GUI)** 1) Graphical User Interface 2) AWT vs Swing 3) Java GUI 4) Swing Components	Chapter 9	Summarization, Case Study, **Practical Class Acticity**	To implement the practical cases of real world using inheritance

	5) Frame 6) Label 7) Button 8) Text Field 9) Password Field 10) Text Area			
10	**Lecture 9: Classes & Objects** 1) What is OOP? 2) OOP advantages over procedural programming 3) What are Classes and Objects? 4) Class 5) Object 6) Using Multiple Classes 7) Java Class Attributes 8) Accessing Attributes 9) Final Keyword 10) Multiple Objects 11) Java Class Methods 12) Static vs. Public 13) Access Methods With an Object 14) Java Constructors 15) Access Modifiers	Chapter 10	Practical Implementation, **Case Study**	Better implementation of interfaces to make efficient programs
11	**Lecture 10: Encapsulation** 1) Introduction 2) Setters 3) Getters 4) Example Practice	Chapter 8. OOP book	Lecture, Summarizing, Case Study, **Class Activity**	To undrestand the composition of the classes
12	**Lecture 11: Inheritance** 1) What is inheritance? 2) Inheritance Examples 3) Understand Types of Inheritance 4) What is IS-A relationship 5) How to use Extends Keyword 6) How to use Methods Overriding 7) What is Super Keyword	Chapter 8	Lecture, Summarizing, **Case Study**	To implement different interfaces to get abstract applications
13	**Lecture 12: Abstraction** 1) What is Abstraction 2) Abstract classes in Java 3) Abstract class vs Concrete class 4) How to create Abstract methods in Java 5) Rules for Java Abstract Class	Chapter 11	Lab work, Student Projects	Provide applications with no or less failure or bugs, provide the group work skill for the students

	6) Advantages of Abstraction			
14	**Lecture 13: Interfaces** 1) Java Interfaces introduction 2) How to use Interface 3) Interface Example 4) Why And When To Use Interfaces? 5) Example when to use Interface/Abstract 6) Abstract class vs Interface	WindowBuilder	Practical class work, case studies, pre-post testing	To start creating graphical apps, Use standard and third party Java's API's when writing applications.
15	**Lecture 14: Polymorphism** 1) What is Polymorphism? 2) How to implement it in Java? 3) What is method overriding? 4) What is method overloading? 5) Practical examples	Chapter 12	Case study, pre-post testing, **Project Assessment**	Understand the basic principles of creating Java applications with graphical user interface (GUI).
16	**Projects** 1) Students' projects 2) Presentation of the projects 3) Documentation 4) Selection of best projects 5) Final Term Paper pattern 6) Feedback of the course 7) Future Planning		Practical classwork, **Project Presentation**	Students will Empower their presentation skills

COURSE ASSESSMENT

SUMMARY

TYPE	PERCENTAGE	RATIONALE
Internal Assessment - Assignments: 5% - Class Activity 5% - Project: 10% - Attendance: 5%	25 %	Equal weightage is assignments and and at the same time emphasizing on the importance of class participation.
External Assessment - Midterm: 25% - Terminal: 50%	75%	Students develop an examination sense through midterm examination hence 25% weightage is appropriate. Midterm duly followed by terminal examination providing 50% weightage which is an opportunity to improve scores appropriately.

GRADING

MARK RANGE	GRADE POINT	GRADE	EXPECTED RESULT
90 and above	4.0	A	At least 25% expected to secure A grade
80 – 89	3.0	B	At least 35% expected to secure B grade
70 – 79	2.0	C	At least 30% expected to secure C grade
60 – 69	1.0	D	At least 7% expected to secure D grade
59 and below	0.0	F	At least 3% expected to secure F grade

Kardan University Policy on Plagiarism:

The students should submit a pure new work of theirs as assignments unless cited. The assignments are checked by the software for plagiarism. If more than 20% is plagiarized, the assignment will not be accepted.

Conduct and Important Policies:

Any student found guilty of a breach of ethics will be refered to Disiplinary Committee of the University.

a) Breach of ethics includes, but is not limited to plagiarism (the copying of other's ideas and passing them off as one's own); copying or other forms of cheating on examinations, papers, and reports; the sale, purchase, or distribution of term papers It is within an instructor's discretion to impose a lesser penalty, e.g., "zero" grade on a given assignment.

b) Course registration is charged by the management. Please approach the management for any queries about course enrolment. In no circumstances should you approach the lecturers who have no control on this.

c) Make-up exam for midterm and terminal exam is available only for those individuals, who are not able to attend their regural exams. Provision of supported documents are mandatory for grant of approval to participate in make-up exams. Those students who miss their regular exam without a genuin reason, will be entitled for 80% of total makeup marks.
Usually make-up exam starts a week after the regular exam finishes.

d) There is no make-up session for the oral presentations and quizzes. If you are absent from the oral presentation/ quiz without eligible reasons/documents, you will not earn any marks.

e) The project presentation should be made at the final session of the course. Each student has to prepare a presentation of 5-10 slides to showcase their project and they should run the project during presentations.

f) There is no supplementary exam for any failed course. Individuals, who fail the course, must retake the module.

Attendance:

Your regular and punctual attendance at lectures and seminars is expected in this course. University regulations indicate that if students attend less than 75% of scheduled classes they may be refused from final assessment.

This course policy has been approved in the Curriculum Development Committee meeting FCS/CDC-2025-01.

Head of the Committee ……

Dean, Faculty of Computer Science …….