दिव्यम् ददामि ते चक्षुः GMIU Gyanmanjari Innovative University

Course Syllabus
Gyanmanjari College of Computer Science
Semester-1(MSCIT)

Subject:Object Oriented Programing using Java-MSCIT11503

Type of course: Major Core

Prerequisite: Basic Knowledge of C and C++ Language

Rationale:

This course is designed to teach object-oriented programming concepts, techniques, and applications using the Java programming language. It emphasizes the fundamentals of structured and object-oriented design with classes, including program development, testing, implementation, and documentation. The course also covers core object-oriented programming techniques involving classes and objects.

Java is a simple, portable, distributed, robust, secure, dynamic, architecture-neutral, and object-oriented programming language. It is designed to support the development of small, reliable, portable, and high-performance applications across a wide range of computing platforms, including distributed and real-time systems. Java enables cross-platform interaction, making applications accessible in heterogeneous environments.

By enabling applications to run across diverse platforms, Java helps businesses provide enhanced services, improve end-user productivity, and support communication and collaboration in both enterprise and consumer applications. The Java programming language originated as part of a research project aimed at developing advanced software for network devices and embedded systems. In this course, Java is used as the primary teaching language to introduce and apply object-oriented programming principles.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks					Total Marks
CI	T	P	C	Theory Marks		Practical Marks		CA	
				ESE	MSE	V	P	ALA	
3	0	4	5	60	30	10	20	30	150

Legends: CI-ClassRoom Instructions; T – Tutorial; P - Practical; C – Credit; ESE - End Semester Examination; MSE- Mid Semester Examination; V – Viva; CA - Continuous Assessment; ALA-Active Learning Activities.

Continuous Assessment:

(For each activity maximum-minimum range is 5 to 10 marks)

Sr. No	Active Learning Activities	Marks
1	Presentation on Particular Topic (Each Student 1 PPT Present)	10
2	Test For Each Unit (1 to 5 Unit)	20
3	Group Discussions and Debates	10
4	Object Oriented Quizzes	10
	Total	50

CourseContent:

Sr. No	Course content	Hrs	% Weightage
1	**Introductionn to JavaProgrammingLanguage:** • Introduction to Java and Brief history, java features, java Applications • Java components: Java Virtual Machine (JVM), Java Runtime Environment (JRE), JDK (Java Development Kit). Importance of byte code and Garbage Collection. • Java environment setup; Structure of java program; Compilation and execution of java program, Comment Syntax. • Primitive Data Types : byte, short, int, long, float, double, char, Boolean • Identifiers, Declarations of constants & variables, Type Conversion and Type Casting, Scope of variables. • Arrays of Primitive Data Types, Types of Arrays : one-dimensional and two- dimensional array. • Different Operators: Arithmetic, Bitwise, Rational, Logical, Assignment, Conditional, Ternary, Increment and	**09**	**20**

GYANMANJARI INNOVATIVE UNIVERSITY
GMIU
BHAVNAGAR

	Decrement. • Decision & Control Statements: SelectionStatement (if, if...else, switch), Loops (while, do-while, for), Jump Statements : break, continue, return		
2	Object Oriented Programming Concepts: • Procedure-Oriented vs. Object Oriented Programming concept. • Basics of OOP: Class, Object, Encapsulation, Polymorphism Abstraction, Inheritance • Defining classes, fields and methods, creating objects. • Accessing rules : public, private, protected, default • this keyword, static keyword, final keyword • Constructors: Default constructors, Parameterized constructors, Copy constructors, Passing object as a parameter • method overloading, constructor overloading • Wrapper Classes, String Class and its methods: chatAt(), contains(), format(), length(), split() • User Input: Scanner class and Command Line Arguments.	09	20
3	Inheritance, Packages & Interfaces: • Basics of Inheritance, Types of inheritance: single, multiple, multilevel, hierarchical and hybrid inheritance. • method overriding, Object class and overriding its methods : equals(), toString(), finalize(), hashCode(). • Defining interface,implementing interface, multiple inheritance using interface. • Abstract class and final class. • Creating package, importing package, access rules for packages.	09	20
4	Exception Handling & Multithreading: • Types of errors, exceptions, try...catch statement, multiple catch blocks, throw and throws keywords, finally clause, uses of exceptions, user defined exceptions • Concept of Multithreading, Creating thread, extending Thread class, implementing Runnable interface, life cycle of a thread, Thread priority, Thread exception handing in threads	09	20
5	**File Handling and Collections Framework:** Stream classes, class hierarchy, Useful I/O classes:	07	20

	FileInputStream, FileOutputStream. Creation of text file, reading and writing text files. Collections Framework overview, Collection classes: ArrayList, LinkedList, HashSet. The For-Each loop Map class :HashMap		

Suggested Specification table with Marks (Theory):60

Distribution of Theory Marks (Revised Bloom's Taxonomy)						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage	30	40	20	10	0	0

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcome:

After learning the course, the students should be able to:	
CO1	Understand the fundamentals java language, including features, components and libraries of java.
CO2	Demonstrate how to define and use classes, interfaces, create objects and methods, how to override and overload methods, compile and execute programs.
CO3	Create packages & interfaces, demonstrate package programs.
CO4	Write a program using exception handling, multithreading with its predefine methods.
CO5	Write a program using Files, binary I/O, collection Frameworks for agiven problem.

List of Practical

Sr.No	Practical Outcomes(PrOs)	Unit No.	Approx.Hrs. Req

1.	* Install JDK, write a simple "Hello World" or similar java program,compilation,debugging,executingusingjavacompilerandinterpreter.	I	2
2.	Write a Java program to find the maximum of three numbers using the conditional operator.	I	1
3.	Write a Java program to reverse the digits of a number using a while loop.	I	1
4.	Write a Java program to add two 3×3 matrices.	I	2
5.	Write a Java program to generate the first *n* prime numbers.	I	2
6.	Write a Java program that defines a class `Student` with two instance variables: `enrollmentNo` and `name`. Create three objects of the `Student` class in the `main()` method and display the students' names..	II	1
7.	*Write a Java program that defines a class `Rectangle` with two instance variables: `height` and `width`. Initialize the class using a constructor.	II	1
8.	Write a Java program to demonstrate the use of the `this` keyword.	II	2
9.	Write a Java program to demonstrate the use of the `static` keyword.	II	2
10.	Write a Java program to demonstrate the use of the `final` keyword.	II	2
11.	*Write a Java program that defines a class `Shape` with two overloaded methods: • `area(float radius)` • `area(float length, float width)` Display the area of a circle and a rectangle using method overloading.	II	2
12	Write a Java program to demonstrate the use of `String` class methods: `charAt()`, `contains()`, `format()`, `length()`, and `split()`.	II	2
		II	1
14.	Write a Java program to demonstrate single inheritance.	III	1
15.	Write a Java program to demonstrate multilevel inheritance.	III	2
16.	Write a Java program to demonstrate hierarchical inheritance.	III	2

17.	Write a Java program to demonstrate method overriding.	III	2
18.	*Write a Java program that defines a class `Car` with two instance variables: `topSpeed` and `name`. Override the `toString()` method in the `Car` class. Create five instances of the `Car` class and print the object details.	III	2
19.	Write a Java program to implement multiple inheritance using interfaces.	III	2
20.	*Write a Java program that defines an abstract class `Shape` with three subclasses: `Triangle`, `Rectangle`, and `Circle`. Declare an abstract method `area()` in the `Shape` class and override it in each subclass to calculate the area.	III	4
21.	Write a Java program to demonstrate the use of a `final` class.	III	2
22.	Write a Java program to demonstrate the use of packages.	III	2
23.	Write a Java program to create a user-defined exception for a "Divide by Zero" error.	IV	2
24.	*Write a Java program to develop a simple banking application where: • The user deposits Rs. 25,000 • Withdraws Rs. 20,000 and Rs. 4,000 • Throws a user-defined exception "Not Sufficient Fund" when the user attempts to withdraw Rs. 2,000 thereafter.	IV	2
25.	*Write a Java program that executes two threads: • One thread displays "Thread1" every 1000 milliseconds • Another thread displays "Thread2" every 2000 milliseconds Create the threads by extending the Thread class.	IV	2
26.	Write a Java program that executes two threads: • One thread prints even numbers • Another thread prints odd numbers The numbers should be displayed from 1 to 200.	IV	2
27.	*Write a Java program to perform read and write operations on a text file.	V	2
28.	Write a Java program to demonstrate the use of List: • Create an ArrayList and add weekdays (in string format) • Create a LinkedList and add months (in string format) Display both lists.	V	2
29.	Write a Java program to create a `HashSet`, add colors (in string format), and iterate through all elements using a for-each loop.	V	2

30.	*Write a Java program to create a HashMap and store data of five students (enrollment number and name). Retrieve and display the student's name using the enrollment number.	V	2

Instructional Method:

The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.

From the content 10% topics are suggested for flipped mode instruction.

Students will use supplementary resources such as online videos, NPTEL/SWAYAM videos, e-courses, Virtual Laboratory

The internal evaluation will be done on the basis of Active Learning Assignment

Practical/Viva examination will be conducted at the end of semester for evaluation of performance of students in laboratory.

Reference Books:

1) Intro to Java Programming, 10th edition, Y.Daniel Liang, Pearson

2) Object oriented programming with Java ,RajkumarBuyya,SThamaraiSelvi, Xingchen Chu,

McGrawHill

3) Programming in Java, Sachin Malhotra, SaurabhChoudhary, Oxford

4) Programming with JAVA , E Balagurusamy, McGrawHill

5) CORE JAVA volume -I Cay Horstmann, Pearson