



Gyanmanjari
Innovative University

Course Syllabus
Gyanmanjari Institute of Technology
Semester-I

Subject: Object Oriented Programing – I (BETCE1130)

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. Object-oriented programming emphasis on the fundamentals of the structured design with classes, including development, testing, implementation and documentation also includes object-oriented programming techniques, classes and objects. Java is a simple, portable, distributive, robust, secure, dynamic, architecture neutral, object-oriented programming language. Java programming language is designed to enable the development of a small, reliable, portable, distributed, real-time operating platform, high- performance applications for the widest range of computing platforms possible as well as cross- platform interaction. By making applications available across heterogeneous environments, businesses can provide more services, boost end-user productivity, communication and collaboration to enterprise and consumer applications. The Java programming language originated as part of a research project to develop advanced software for a wide variety of network devices and embedded systems. The Java programming language is used as the teaching vehicle for this course.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks					Total Marks
CI	T	P		C	Theory Marks		Practical Marks		
			ESE		MSE	V	P	ALA	
4	0	2	5	60	30	10	20	30	150

Legends: CI-Class Room 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	Poster Presentation. Poster Topic will be provided by the subject faculty. Student need to prepared poster and upload file on Moodle. (Group of two students)	05
2	Object Oriented Quizzes. Unit wise MCQ test conducted on Moodle. Per unit maximum 5 questions will be allocated to the students.	05
3	Problem solving using Programming. Problem statement will be provided to the student by the subject faculty. Student will provide solution and upload on Moodle. (Group of two students)	10
4	Reverse Engineering. Output of the program will be given to the student by the subject faculty. Student will provide solution and upload on Moodle. (Group of two students)	10
Total		30

Course Content:

Sr. No	Course content	Hrs.	(%) Weightage
1	Introduction to Java Programming Language: <ul style="list-style-type: none"> • 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 Decrement. 	15	25 %



	<ul style="list-style-type: none"> Decision & Control Statements: Selection Statement (if, if...else, switch), Loops (while, do-while, for), Jump Statements: break, continue, return. 		
2	<p>Object Oriented Programming Concepts:</p> <ul style="list-style-type: none"> 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. 	15	25 %
3	<p>Inheritance, Packages & Interfaces:</p> <ul style="list-style-type: none"> 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. 	15	25 %
4	<p>Exception Handling & Multithreading:</p> <ul style="list-style-type: none"> 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 	12	25 %



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 (%)	20%	35%	35%	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 it predefines methods.

List of Practical

Sr. No	Descriptions	Unit No	Hrs.
1	Install JDK, write a simple "Hello World" or similar java program, compilation, debugging, executing using java compiler and interpreter.	I	1
2	Write a program in Java to find maximum of three numbers using conditional operator.	I	1
3	Write a program in Java to reverse the digits of a number using while loop.	I	1
4	Write a program in Java to add two 3*3 matrices.	I	1
5	Write a program in Java to generate first n prime numbers.	I	1
6	Write a program in Java which has a class Student having	II	1



	two instance variables enrollment No and name. Create three objects of Student class in main method and display student's name.		
7	Write a program in Java which has a class Rectangle having two instance variables height and weight. Initialize the class using constructor.	II	1
8	Write a program in Java demonstrate the use of "this" keyword.	II	1
9	Write a program in Java to demonstrate the use of "static" keyword.	II	1
10	Write a program in Java to demonstrate the use of "final" keyword.	II	1
11	Write a program in Java which has a class Shape having 2 overloaded methods area (float radius) and area (float length, float width). Display the area of circle and rectangle using overloaded methods.	II	1
12	Write a java program to demonstrate use of "String" class methods: chatAt(), contains(), format(), length(), split()	II	1
13	Write a program in Java to demonstrate single inheritance	III	1
14	Write a program in Java to demonstrate multilevel inheritance	III	1
15	Write a program in Java to demonstrate hierarchical inheritance.	III	1
16	Write a program in Java to demonstrate method overriding.	III	1
17	Write a program in Java which has a class Car having two instance variables topSpeed and name. Override toString() method in Car class. Create 5 instances of Car class and print the instances.	III	2
18	Write a program in Java to implement multiple inheritance using interfaces.	III	1
19	Write a program in Java which has an abstract class Shape having three subclasses: Triangle, Rectangle, and Circle. Define method area () in the abstract class Shape and override area() method to calculate the area.	III	2
20	Write a program in Java to demonstrate use of final class.	III	1
21	Write a program in Java to demonstrate use of package.	III	1
22	Write a program in Java to develop user defined exception for 'Divide by Zero' error.	IV	1
23	Write a program in Java to develop Banking Application in which user deposits the amount Rs 25000 and then start withdrawing of Rs 20000, Rs 4000 and it throws exception "Not Sufficient Fund" when user withdraws Rs. 2000 thereafter.	IV	2
24	Write a program that executes two threads. One thread displays "Thread1" every 1000 milliseconds, and the other displays "Thread2" every 2000 milliseconds. Create the	IV	2



	threads by extending the Thread class		
25	Write a program that executes two threads. One thread will print the even numbers and another thread will print odd numbers from 1 to 200.	IV	2
		Total	30

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 , Rajkumar Buyya,S Thamarai Selvi, Xingchen Chu,McGrawHill
3. Programming in Java, Sachin Malhotra, Saurabh Choudhary, Oxford
4. Programming with JAVA, E Balagurusamy, McGrawHill
5. CORE JAVA volume -I Cay Horstmann, Pearson

