

Subject Code: 1ET1100305	Subject Title: Object Oriented Programming Using Java
Prerequisite	C Language

Course Objective:

The objective is to impart programming skills used in object oriented language java. The students are expected to learn it enough so that they can develop programs in Java and the web solutions like creating applets etc.

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
Lecture	Tutorial	Practical	Credit	Theory		Practical		Total
				University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
3	1	2	5	70	30	30	20	150

Subject Contents			
Sr. No	Topic	Total Hours	Weight (%)
1	FUNDAMENTALS OF OBJECT ORIENTED PROGRAMMING Introduction, Basic Concepts of OOP, Java features, Simple Java program structure, Java tokens, Implementing a Java Program, Java Virtual Machine Constants, Variables, Data Types, Declaration of Variables, Giving Value to Variables, Scope of variables, Symbolic Constants, Type casting, Getting Value of Variables, Standard Default values, Operators & Expressions. BRANCHING & LOOPING Decision making with if statement, if-else statement, Nesting of if-else statements, the else if ladder, the switch statement, the conditional operator, While loop, do-while loop, for loop, Jumps in loops. ARRAYS AND STRINGS: Arrays, One-dimensional arrays, Creating an array, Two – dimensional arrays, Strings, Wrapper classes.	6	15
2	CLASSES, OBJECTS & METHODS Defining a class, Adding variables, Adding methods, Creating objects, Accessing class members, Constructors, Method overloading, Static members, Nesting of methods	6	15
3	INHERITANCE AND INTERFACES Extending a class, Types of Inheritance, Overriding methods, Final variables and methods, Final classes, Abstract methods and classes INTERFACES: Introduction, Defining interfaces, Extending interfaces, Implementing interfaces, Assessing interface variables	6	15
4	MULTITHREADED PROGRAMMING AND EXCEPTION HANDLING Introduction, Creating Threads, Extending the Threads, Stopping and Blocking a Thread, Lifecycle of a Thread, Using Thread Methods, Thread Exceptions, Thread Priority, Synchronization, Implementing the 'Runnable' Interface. ERRORS AND EXCEPTIONS: Types of errors: Compile-time errors, Run-time errors, Exceptions, Exception handling, Multiple Catch Statements, Using final statement.	7	20
5	PACKAGES AND I/O PROGRAMMING Introduction, Java API Packages, Using System Packages, Naming conventions, Creating Packages, Accessing a Package, using a Package. INPUT/OUTPUT FILES IN JAVA: Introduction, Concept of Streams, Stream classes, Byte Stream Classes, Input Stream Classes, Output Stream Classes, Character Stream classes: Reader stream classes, Writer Stream classes, Using Streams, Reading and writing files.	8	15
7	Event and GUI programming Event handling in java, Event types, Mouse and key events, GUI Basics, Panels, Frames, Layout Managers: Flow Layout, Border Layout, Grid Layout, GUI components like Buttons, Check Boxes, Radio Buttons, Labels, Text Fields, Text Areas, Combo Boxes, Lists, Scroll Bars, Sliders, Windows, Menus, Dialog Box, Applet and its life cycle.	8	20

Course Outcome:

At the end of this course the student will:

Understand the concept and underlying principles of Object-Oriented Programming.

1. Understand how object-oriented concepts are incorporated into the Java programming language.
2. Develop problem-solving and programming skills using OOP concept.
3. Develop the ability to solve real-world problems through multi-threaded programming using Java.
4. Become familiar with creating packages and working with files in Java.

Course Learning Outcomes (CLO): On completion of this course, the students will be able to:

CLO	Description	Bloom's Taxonomy Level
CLO 1	Understand the concept and underlying principles of Object-Oriented Programming.	Understanding
CLO 2	Understand how object-oriented concepts are incorporated into the Java programming language.	Understanding, Applying
CLO 3	Develop problem-solving and programming skills using OOP concept.	Applying
CLO 4	Develop the ability to solve real-world problems through multi-threaded programming using Java.	Analyzing, Evaluating
CLO 5	Become familiar with creating packages and working with files in Java.	Analyzing, Evaluating

Mapping of CLOs with POs & PSOs

Course Learning Outcomes	Program Outcomes (POs)												Program Specific Outcomes(PSOs)			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CLO1	M	M	H	H	M	M	L	L	M	H	M	M	M	H	L	L
CLO2	M	H	H	M	M	H	L	L	M	M	L	L	H	H	M	M
CLO3	H	M	H	M	H	M	M	L	M	L	M	M	M	M	H	M
CLO4	M	H	M	H	M	H	L	L	M	M	M	L	H	H	M	H
CLO5	M	H	H	H	M	M	L	L	M	M	M	L	M	H	M	M

List of Book References:

1. The Complete Reference, Java 2 (Fourth Edition), Herbert Schild, TMH.
2. Introduction to JAVA Programming a primer, Balaguruswamy.
3. Introduction to Java Programming (Comprehensive Version), Daniel Liang, Seventh Edition, Pearson.
4. Programming in Java, Sachin Malhotra & Saurabh Chaudhary, Oxford University.
5. Head First Java, Katy Sierra & Bert Bates, SPD (O'Reilly).

E-Resources / Web Links

1. <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. <http://docs.oracle.com/javase/specs/jls/se7/html/index.html>
3. <http://docs.oracle.com/javase/tutorial/java/index.html>
4. <http://www.javatpoint.com/>
5. <http://www.tutorialspoint.com/java/>
6. <http://www.learnjavaonline.org/>

List of Experiments:

Note: The experiment list provided beneath is for reference only. The course teacher may change/formulate it as per his/her methodology and requirement.

1. Install JDK, write a simple "Hello World" java program, compilation, debugging, executing using java compiler and interpreter.
2. Write a program in Java to generate first n prime numbers.
3. Write a program in Java to find a maximum of three numbers using a conditional operator.
4. Write a program in Java to find the second maximum of n numbers without using arrays.
5. Write a program that calculates percentage marks of the student if marks of 6 subjects are given.
6. Write a program in Java to reverse the digits of a number using a while loop.
7. Write a program to count the number of words that start with capital letters.
8. Write a program to find if a given number or string is palindrome or not.
9. Write a program in Java to multiply two matrices.
10. Write a Java application to create a class "time" having hour, minute and second as data members and get time(), show time() and add time() as methods. Also define the necessary constructors. Create two objects of "time" and add them and display the result.
11. Write a program in Java to demonstrate the use of 'final' keyword in the field declaration. How it is accessed using the objects.
12. Write a program in Java to demonstrate single inheritance, multilevel inheritance and hierarchical inheritance.
13. Write an application that illustrates how to access a hidden variable. Class A declares a static variable x. The class B extends A and declares an instance variable x, display () method in class B displays both of these variables.
14. Write a program that illustrates interface inheritance. Interface P12 inherits from both P1 and P2. Each interface declares one constant and one method. The class Q implements P12. Instantiate Q and invoke each of its methods. Each method displays one of the constants.
15. Write an application that illustrates method overriding in the same package and different packages. Also demonstrate accessibility rules in inside and outside packages.
16. Describe abstract class called Shape which has three subclasses say Triangle, Rectangle, and Circle. Define one method area () in the abstract class and override this area () in these three subclasses to calculate for specific objects i.e. area() of Triangle subclass should calculate area of triangle etc. Same for Rectangle and Circle.
17. Write a program in Java to demonstrate implementation of multiple inheritance using interfaces.
18. Write a small application in Java to develop Banking Application in which the user deposits the amount Rs 1000.00 and then starts withdrawing of Rs 400.00, Rs 300.00 and it throws exception "Not Sufficient Fund" when user withdraws Rs. 500 thereafter.
19. Write a program that executes two threads. One thread displays "Thread1" every 2,000 milliseconds, and the other displays "Thread2" every 4,000 milliseconds. Create the threads by extending the Thread class.
20. Write a program that executes two threads. One thread will print the even numbers and another thread will print odd numbers from 1 to 50.
21. Write a program in Java to create, write, modify, read operations on a Text file.
22. Write a Java Application to enter a two number in two text fields and display the sum in the third text field when you press the Add button.