Faculty of Engineering and Technology

Second Year (4th Sem) Bachelor of Technology

In Effect from Academic Year 2017-18

Subject Code: 1ET1030406	Subject Title: Java Programming
Pre-requisite	C,C++

Course Objective:

Java is a general-purpose computer programming language that is a class-based, object-oriented. It is intended to let application developers "write once, run anywhere" meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
				Theory		Practical		
Lecture	Tutorial	Practical	Credit	University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	Total
4	-	2	5	70	30	30	20	150

Subject Contents			
Sr. No	Topic	Total Hours	Weight (%)
1	**Basics of Java:** Features of Java, Byte Code and Java Virtual Machine, JDK, Data types, Operator, Variables, Control Statements – If , else, nested if, if-else ladders, Switch, while, do-while, for, for-each, break, continue.	3	10
2	**Array and String:** Single and Multidimensional Array, String class, String Buffer class, Operations on string, Command line argument, Use of Wrapper Class, Inbuilt classes like Character, File, this reference	3	5
3	**Classes, Objects and Methods:** Class, Object, Object reference, Access control, modifiers, Constructor, Constructor Overloading, Method Overloading, Finalizer, Recursion, new operator, this and static keyword, finalize() method, Nested class, Inner class, Abstract class.	4	10
4	**Inheritance & Interfaces and Polymorphism:** Use of Inheritance, Inheriting Data members and Methods, constructor in inheritance, Multilevel Inheritance – method overriding Handle multilevel constructors – super keyword, Stop Inheritance - Final keywords, Creation and Implementation of an interface, Interface reference, instance of operator, Interface inheritance, Dynamic method dispatch/ Dynamic Binding ,Understanding of Java Object Class, Comparison between Abstract Class and interface.	5	15
5	**Package:** Use of Package, CLASSPATH, Import statement, Static import, Access control across Packages.	3	10
6	**Collection Classes :** List, AbstractList, Dictionary Class, HashTable, ArrayList, LinkedList, Enumeration, Vector, Properties, Introduction to Java.util package	3	10
7	**I/O programming :** Introduction to Stream, Byte Stream, Character stream, Readers and Writers,	4	10

	File Class, File InputStream, File Output Stream, InputStreamReader, OutputStreamWriter, FileReader, FileWriter, Buffered Reader, Text and Binary I/O, Binary I/O classes.		
8	**Exception Handling:** Exception and Error, Use of try, catch, throw, throws and finally, Built in Exception, User Defined exception class, Throwable Class.	2	10
9	**Multithreading in java :** Thread life cycle and methods, Thread class and Runnable interface, Thread synchronization, Exception handling with try-catch-finally, Collections in java,	3	10
10	**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, Introduction to swing.	4	10

Course Outcome:

After learning the course the students should be able to:

→Understand object oriented programming concepts and implement in java.
→Comprehend building blocks of OOPs language, inheritance, package and interfaces.
→Identify exception handling methods.
→Implement multithreading in object oriented programs.

List of References:

1. The Complete Reference, Java 2 (Fourth Edition), Herbert Schild, TMH.
2. Introduction to Java Programming (Comprehensive Version), Daniel Liang, Seventh Edition, Pearson.
3. Programming in Java, Sachin Malhotra & Saurabh Chaudhary, Oxford University Press.
4. Murach's Beginning Java 2, Doug Lowe, Joel Murach and Andrea Steelman, SPD.
5. Core Java Volume-I Fundamentals, Eight Edition, Horstmann & Cornell, Pearson Education.
6. Java Programming, D. S. Malik, Cengage Learning.
7. Big Java, 3rd Edition, Horstmann, Wiley-India.
8. Head First Java, Katy Sierra & Bert Bates, SPD (O'Reilly).
9. Java Fundamentals A comprehensive introduction By Herbert Schildt, Dale Skrien, McGrawHill Education.
10. Programming with Java A Primer – E.Balaguruswamy,Mc Grawhill
11.Core Java Volume-I Fundamentals Horstmann & Cornell, - Pearson Education. - Eight Edition

E-Resource:

i. Java Development Kit: http://www.oracle.com/technetwork/java/javase/downloads/index.html
ii. http://docs.oracle.com/javase/specs/jls/se7/html/index.html
iii. http://docs.oracle.com/javase/tutorial/java/index.html
iv. http://www.javatpoint.com/
v. http://www.tutorialspoint.com/java/
vi. http://www.learnjavaonline.org/

vii. http://www.c4learn.com/javaprogramming/
viii. http://www.learn-java-tutorial.com/

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.

Sr.No	Practical Exercise
1.	Write a program to convert rupees to dollar.
2.	Write a program that calculates percentage marks of the student if marks of 6 subjects are given.
3.	Write a program to enter two numbers and perform mathematical operations on them

4.	Write an interactive program to print a diamond shape. For example, if user enters the number 3, the diamond will be as follows:
	  *
	 * *
	* * *
	 * *
	  *

5.	Write a program to find length of string and print second half of the string.
6.	Write a program to accept a line and check how many consonants and vowels are there in line.
7.	Write a program to count the number of words that start with capital letters.
8.	Write a program to find that given number or string is palindrome or not.
9.	Write a Java application to create a class “time “ having hour, minute and second as data members and gettime(), showtime() and addtime() as method. Also define the necessary constructors. Create two objects of “time” and add them and display the result.
10.	Create a class Room having two data members, length & width. Now write another class RoomArea that creates an instance of this class and displays area of a room.
11.	Create an abstract class "shape" having two data members, constructors and abstract method disparea(). Derive a class "rectangle" and "traingle" from "shape" class which include appropriate constructors and implement disparea() methods. Create objects of "rectangle" and "triangle" classes and call the disparea() methods.
12.	Write a Java program to demonstrate (a) use of implementing interfaces(b) use of extending interfaces.
13.	Create a class Figure having one method area(). Create one more class Rectangle which will override area() method of rectangle class. Demonstrate with example.
14.	Create a class called Student. Write a student manager program to manipulate the student information from files by using FileInputStream and FileOutputStream
15.	Write a program using Applet (a)to display a message in the Applet (b) for configuring Applets by passing parameters.
16.	Write a Java Application to enter a two number in two text fields and display the sum in the third text field when you pressed the Add button.
17.	Write a program in java applet for filling various graphical objects with color.
18.	Write a java **applet** that tracks the **position of the mouse** when it is dragged or moved. At the current mouse position, it **displays message (x, y)** showing current position of the mouse. The message should disappear as soon as the user releases the mouse.
19.	Write a java applet to create a **frame** with exit capabilities. **Handle events for mouse**

	pressed, released, clicked and dragged by **displaying appropriate message** describing the event at the coordinates where the event has taken place.