**JAVA PROGRAMMING**

**COURSE OUTCOMES**

At the end of the course, the student will develop ability to

1. List all OOP features to design object oriented applications, and execute straight forward programs using a high level language.
2. Discuss the principles and practice of object oriented analysis and design in the construction of robust, maintainable programs which satisfy their requirements.
3. Analyze implementation, compilation, testing and run java programs comprising more than one class, to address a particular software problem.
4. Classify effective user interface applications through AWT controls and swings.
5. Examine use of members of classes in the Java API.
6. Summarize the framework and architecture for MVC’s

**UNIT I**

**Object Oriented Thinking**

OOP Principles, Java buzzwords, data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion, concepts of classes, objects, constructors, methods, access specifiers, garbage collection.

**UNIT II**

**Inheritance**

Super class, Sub class, Types of inheritance’s using final with inheritance, polymorphism- method overriding, Dynamic Method dispatch, abstract classes, Interfaces, variables in interface and extending interfaces. Overloading methods, parameter passing, recursion. Packages Defining, Creating and Accessing a Package, importing packages.

**UNIT III**

**Exception Handling**

Need for Exceptional Handling, try, catch, throw, throws and finally, built in exceptions, creating own exception. Streams- File Input Stream, File Output Stream, Data Input Stream, Data Output Stream, Scanner, File Reader, File Write. Byte Array, Char Array.

**UNIT IV**

**Multi Threading**

Differences between multithreading and multitasking, thread life cycle, creating threads, synchronizing threads, daemon threads, thread groups. Event Handling: Events, Event sources, Event classes, Event Listeners, handling mouse events, keyboard events, Adapter classes, inner classes.

**UNIT V**

**AWT Controls**

Labels, button, text components, check box, check box groups, choices, lists, menu bar layout manager types – boarder, grid, flow, card and grib bag. limitations of AWT, MVC architecture, components, containers

**UNIT VI**

**Swings**

Introduction exploring swing- J Applet, J Frame and J Component, Icons and Labels, text fields, buttons – The J Button class, Check boxes, Radio buttons, Combo boxes, Tabbed Panes, Scroll Panes, Trees, and Tables. JDBC Concepts With Simple Programs.

**TEXT BOOKS**

1. Java 7 Programming - Black Book, By Kogent Learning Solutions Inc., Freamtech Publications

2. Herbert schildt “Java the complete reference”, 7th Edition, TMH,ISBN:0072263857

**REFERENCE BOOKS**

1. Y. Daniel Liang “Introduction to Java programming” 6th Edition, pearson education, ISBN:10:0132221586
2. R.A. Johnson-An introduction to Java programming and object oriented application development, Thomson, ISBN:-10:0619217464
3. Head First Java 2nd Edition by Kathy Sierra, Oreilly Publication
4. T.Budd “Understanding OOP with Java” updated Edition, Pearson education, ISBN:10:0201612739

**JAVA PROGRAMMING LAB**

**(Professional Elective - II)**

**COURSE OUTCOMES**

At the end of the course, the students will develop ability to

1. Design, compile, test and execute straightforward programs using a high level language.
2. Discuss the principles and practice of object oriented analysis and design in the construction of robust, maintainable programs which satisfy their requirements.
3. Analyze implementation, compilation, testing and run java programs comprising more than one class, to address a particular software problem.
4. Illustrate synchronization using multithreading.
5. Classify effective user interface applications through AWT controls and swings.
6. Program and examine use of members of classes in the Java API.

**Week 1**

Write a Java program that prints all real solutions to the quadratic equation ax2 + bx + c = 0. Read in a, b, c and use the quadratic formula. If the descriminent (b2 - 4ac) is negative, display a message stating that there are no real solutions.

* 1. The Fibonacci sequence is defined by the following rule:

The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a Java program that uses both recursive and non recursive functions to print the nth value in the Fibonacci sequence.

**Week 2**

* 1. Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.
  2. Write a Java program to multiply two given matrices.
  3. Write a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use StringTokenizer class of java.util)

**Week 3**

1. Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome.
2. Write a Java program for sorting a given list of names in ascending order.
3. Write a Java program to make frequency count of words in a given text

**Week 4**

1. Write a Java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
2. Write a Java program that reads a file and displays the file on the screen, with a line number before each line.
3. Write a Java program that displays the number of characters, lines and words in a text file.

**Week 5**

* 1. Write a Java program that:
     1. Implements stack ADT.
     2. Converts infix expression into Postfix form
     3. Evaluates the postfix expression

**Week 6**

Develop an applet that displays a simple message.

Develop an applet that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” is clicked.

**Week 7**

Write a Java program that works as a simple calculator. Use a grid layout to arrange Buttons for the digits and for the +, -,\*, % operations. Add a text field to display the result.

**Week 8**

1. Write a Java program for handling mouse events.

**Week 9**

1. Write a Java program that creates three threads. First thread displays “Good Morning” Every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds.
2. Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.

**Week 10**

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

**Week 11**

1. Write a java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time No light is on when the program starts.
2. Write a Java program that allows the user to draw lines, rectangles and ovals.

**Week12**

1. A demonstration of the Progress Monitor toolbar. A timer is used to induce progress.
2. This example also shows how to use the UI Manager properties associated with progress monitors.
3. Sample Swing application that manages several internal frames. This is the main class for working with the Site Frame and Page Frame classes.