VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
OBJECT ORIENTED PROGRAMMING THROUGH JAVA

B.Tech. II Year I Sem.   L   T   P   C

                                                                                  3   0   0   3

Course Objectives

➢ To Understand the basic object-oriented programming concepts and apply them in problemsolving.

➢ To Illustrate inheritance concepts for reusing the program.

➢ To Demonstrate multitasking by using multiple threads and event handling

➢ To Develop data-centric applications using JDBC.

➢ To Understand the basics of java console and GUI based programming

UNIT - I

Object oriented thinking and Java Basics- Need for oop paradigm, summary of oop concepts, coping with complexity, abstraction mechanisms. A way of viewing world – Agents, responsibility, messages, methods, History of Java, Java buzzwords, data types, variables, scope and lifetime of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, method binding, inheritance, overriding and exceptions, parameter passing, recursion, nested and inner classes, exploring string class.

UNIT - II

Inheritance, Packages and Interfaces – Hierarchical abstractions, Base class object, subclass, subtype, substitutability, forms of inheritance specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super uses,using final with inheritance, polymorphism- method overriding, abstract classes, the Object class. Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring java.io.

UNIT - III

Exception handling and Multithreading-- Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception subclasses. String handling, Exploring java.util. Differences between multithreading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication, thread groups, daemon threads. Enumerations, autoboxing, annotations, generics.

UNIT - IV

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes. The AWT class hierarchy, user interface components- labels, button, canvas, scrollbars, text components, check box, checkbox groups, choices, lists panels – scrollpane, dialogs, menubar, graphics, layout manager – layout manager types – border, grid, flow, card and grid bag.

UNIT - V

Applets – Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets. Swing – Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing- JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons – The JButton class, Check boxes, Radio buttons, Combo boxes,Tabbed Panes, Scroll Panes, Trees, and Tables.

Course Outcomes

CO-1: Demonstrate the behavior of programs involving the basic programming constructs like controlstructures, constructors, string handling and garbage collection.

CO-2: Demonstrate the implementation of inheritance (multilevel, hierarchical and multiple) by usingextend and implement keywords

CO-3: Use multithreading concepts to develop inter process communication.

CO-4: Understand the process of graphical user interface design and implementation using AWT orswings.

CO-5: Develop applets that interact abundantly with the client environment and deploy on the server.

TEXT BOOKS:
1. Java the complete reference, 7th edition, Herbert schildt, TMH.
2. Understanding OOP with Java, updated edition, T. Budd, Pearson education.

REFERENCE BOOKS:
1. An Introduction to programming and OO design using Java, J.Nino and F.A. Hosch, John wiley& sons.
2. An Introduction to OOP, third edition, T. Budd, Pearson education.
3. Introduction to Java programming, Y. Daniel Liang, Pearson education.
4. An introduction to Java programming and object-oriented application development, R.A. Johnson- Thomson.
5. Core Java 2, Vol 1, Fundamentals, Cay.S. Horstmann and Gary Cornell, eighth Edition,Pearson Education.
6. Core Java 2, Vol 2, Advanced Features, Cay.S. Horstmann and Gary Cornell, eighth Edition,Pearson Education
7. Object Oriented Programming with Java, R.Buyya, S.T.Selvi, X.Chu, TMH.
8. Java and Object Orientation, an introduction, John Hunt, second edition, Springer. 9. Maurach'sBeginning Java2 JDK 5, SPD.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

B.Tech. II Year I Sem.   L T P C
                                                                        0 0 3 1.5

Course Objectives:
  - To write programs using abstract classes.
  - To write programs for solving real world problems using the java collection framework.
  - To write multithreaded programs.
  - To write GUI programs using swing controls in Java.
  - To introduce java compiler and eclipse platform.
  - To impart hands-on experience with java programming.

Note:
1.   Use LINUX and MySQL for the Lab Experiments. Though not mandatory, encourage the useof the Eclipse platform.

2.   The list suggests the minimum program set. Hence, the concerned staff is requested to addmore problems to the list as needed.

List of Experiments:
1.   Use Eclipse or Net bean platform and acquaint yourself with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

2.   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. Handle any possible exceptions like divided by zero.

3.   A) Develop an applet in Java that displays a simple message.
B) Develop an applet in Java that receives an integer in one text field, and computes its factorialValue and returns it in another text field, when the button named "Compute" is clicked.

4.   Write a Java program that creates a user interface to perform integer divisions. The user enters twonumbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 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.

5.   Write a Java program that implements a multi-thread application that has three threads. First threadgenerates a random integer every 1 second and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.

6.   Write a Java program for the following:Create a doubly linked list of elements.

Delete a given element from the above list. Display the contents of the list after deletion.

7.   Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in the selected color. Initially, there is no message shown.

8.   Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

9.   Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, andthe remaining lines correspond to rows in the table. The elements are separated by commas.
Write a java program to display the table using Labels in Grid Layout.

10.   Write a Java program that handles all mouse events and shows the event name at the center of thewindow when a mouse event is fired (Use Adapter classes).

11.   Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint:use hash tables).

12.   Write a Java program that correctly implements the producer – consumer problem using the concept of inter thread communication.

13.   Write a Java program to list all the files in a directory including the files present in all its subdirectories.

Course Outcomes:
      CO-1:   Able to write programs for solving real world problems using the java collection framework.
      CO-2:   Able to write programs using abstract classes.
      CO-3:   Able to write multithreaded programs.
      CO-4:   Able to write GUI programs using swing controls in Java.

REFERENCE BOOKS:
1.   Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition Pearson education.
2.   Thinking in Java, Bruce Eckel, Pearson Education.
3.   Java Programming, D. S. Malik and P. S. Nair, Cengage Learning.
4.   Core Java, Volume 1, 9th edition, Cay S. Horstmann and G Cornell, Pearson.