Subject Code: 01CE0403

Subject Name: Object Oriented Programming with Java

B.Tech. Year - II

Objective: Java is a computer programming language having feature like object-oriented, polymorphism, inheritance and multithreading. It comprises of large third-party library using which we can develop software.

Credits Earned: 5 Credits

Course Outcomes: After completion of this course, student will be able to

- To understand object-oriented programming concepts and implement in java. (Understand)
- To comprehend building blocks of OOPs language, inheritance, package and interfaces. (Understand)
- To identify exception handling methods. (Apply)
- To implement multithreading in object-oriented programs. (Apply)
- To develop GUI based desktop application in project-based learning. (Create)

Pre-requisite of course: NA.

Teaching and Examination Scheme

Teaching Scheme (Hours)			Credits	Theory Marks			Tutorial/ Practical Marks		Total Marks
Theory	Tutorial	Practical		ESE (E)	Mid Sem (M)	Internal (I)	Viva (V)	Term work (TW)	
4	0	2	5	50	30	20	25	25	150

Contents:

Unit	Topics	Contact Hours
1	**Java Overview:** Java Introduction, Platform Independence, JVM & JDK, Data types, Operators, If, else statement, Switch condition, while, do-while, for loop, break and continue statement.	4

2	**Array and String:** Single Array & Multidimensional Array, Library Classes-String, String Buffer & Wrapper Class, Command line arguments and Various String Operations.	6
3	**Classes, Objects and Methods:** Class and Object, Object reference, Constructor: Constructor Overloading, Method: Method Overloading, Recursion, Passing and Returning object form Method, new operator, this and static keyword, finalize() method, Import statement, Static import, Access control, Nested class, Inner class, Anonymous inner class.	8
4	**Inheritance and Interfaces in Java:** Overview of Inheritance, inheritance in constructor, Inheriting Data members and Methods, Multilevel Inheritance – method overriding Handle multilevel constructors Explain super keyword, Stop Inheritance ,Explain Final keywords, Creation and Implementation of an interface, Interface reference, instance of operator, Interface inheritance, Dynamic method dispatch, Abstract class, Comparison between Abstract Class and interface, inside of System.out.println – statements.	8
5	**Exception Handling in Java:** Exception and Error, Use of try, catch, throw, throws and finally, Built in Exception, Custom exception, Throwable Class.	4
6	**Multithreaded in Java:** Introduction of Multithread programming, Thread classes and Runnable interface, Thread priority and synchronization, Thread communication and Deadlock.	6
7	**JAVA File Handling:** Overview of Different Stream (Byte Stream, Character stream), Readers and Writers class, File Class, File Input Stream, File Output Stream, Input Stream Reader and Output Stream Writer class, File reader and writer class, File Writer, Buffered Reader class.	6
8	**Collection Classes:** List Class (Abstract List), Array List class, LinkedList class, Enumeration Iterative Statement, Vector class.	4
9	**Applet, AWT and Swing:** MVC Architecture, Applet: Applet Fundamental, Applet Architecture, Applet Skeleton, Requesting Repainting, , Event Handling: various event handling mechanisms, Delegation Event Model, Events, Event Sources, Event Listeners, various classes related to event sources and event listeners, AWT: window fundamentals, creating frames, Adding removing various controls, Layout managers, Introduction To Swing,	10

Applications and Pluggable look and feel, Basic swing components : Text Fields, Buttons, Toggle Buttons, Checkboxes, and Radio Buttons.		
Total Hours		**56**

References:

1. Java 7 Programming Black Book by Kogent Learning Solutions Inc,DreamTech press
2. Java Fundamentals A comprehensive introduction By Herbert Schildt, Dale Skrien, McGraw Hill Education.
3. Programming with Java A Primer – E.Balaguruswamy,Mc Grawhill
4. The Complete Reference, Java 2 (Fourth Edition),Herbert Schild, - TMH.
5. Core Java Volume-I Fundamentals Horstmann & Cornell, - Pearson Education. - Eight Edition
6. Head First Java by Kathy Sierra, Bert Bates, O'Reilly publications

Suggested Theory distribution:

The suggested theory distribution as per Bloom's taxonomy is as per follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery and evaluation					
Remember	Understand	Apply	Analyse	Evaluate	Create
10%	20%	40%	10%	10%	10%

Suggested List of Experiments:

1. Demonstrate HelloWorld Application with Single and Multiple Main in a java program.
2. Write a console program to define and initialize a variable of type byte to 1, and then successively multiply it by 2 and display its value 8 times. Explain the reason for the last result.
3. Write a program that defines a floating-point variable initialized with a dollar value for your income and a second floating-point variable initialized with a value corresponding to a tax rate of 35 percent. Calculate and output the amount of tax you must pay with the Rs. and paisa stored as separate integer values (use two variables of type int to hold the tax, perhaps taxRs and taxPaisa).

4. Write a program that calculate percentage marks of the student if marks of 6 subjects are given.

5. Write a program to display a random choice from a set of six choices for breakfast (you could use any set; for example, scrambled eggs, waffles, fruit, cereal, toast, or yogurt).

6. When testing whether an integer is a prime, it is sufficient to try to divide by integers up to the square root of the number being tested. Write a program to use this approach.

7. A lottery requires that you select six different numbers from the integers 1 to 49. Write a program to do this for you and generate five sets of entries.

8. Write a program to generate a random sequence of capital letters that does not include vowels.

9. Write an interactive program to print a string entered in a pyramid form. For instance, the string "stream" has to be displayed as follows:

```
         S
        S t
       S  t  r
      S  t  r  e
     S  t  r  e  a
    S  t  r  e  a  m
```

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

```
      *
     * *
    * * *
     * *
      *
```

11. Create an array of String variables and initialize the array with the names of the months from January to December. Create an array containing 12 random decimal values between 0.0 and 100.0. Display the names of each month along with the corresponding decimal value. Calculate and display the average of the 12 decimal values.

12. Write a program to accept a line and check how many consonants and vowels are there in line.

13. Write a program to find length of string and print second half of the string.

14. Write a program to find that given number or string is palindrome or not.

15. Write a program that sets up a String variable containing a paragraph of text of your choice. Extract the words from the text and sort them into alphabetical order. Display the sorted list of words. Also count the number of words that start with capital letters. You could use a simple sorting method called the bubble sort. To sort an array into ascending order the process is as follows:

a. Starting with the first element in the array, compare successive elements (0 and 1, 1 and 2, 2 and 3, and so on).

b. If the first element of any pair is greater than the second, interchange the two elements.

c. Repeat the process for the whole array until no interchanges are necessary. The array elements will now be in ascending order.

16. Create a class which ask the user to enter a sentence, and it should display count of each vowel type in the sentence. The program should continue till user enters a word "quit". Display the total count of each vowel for all sentences

17. Define a class, mcmLength, to represent a length measured in meters, centimeters, and millimeters, each stored as integers. Include methods to add and subtract objects, to multiply and divide an object by an integer value, to calculate an area resulting from the product of two objects, and to compare objects. Include constructors that accept three arguments—meters, centimeters, and millimeters; one integer argument in millimeters; one double argument in centimeters; and no arguments, which creates an object with the length set to zero. Check the class by creating some objects and testing the class operations.

18. Define a class, tkgWeight, to represent a weight in tons, kilograms, and grams, and include a similar range of methods and constructors as the previous example. Demonstrate this class by creating and combining some class objects.

19. Put both the previous classes in a package called Measures. Import this package into a program that will calculate and display the total weight of the following: 200 carpets—size: 4 meters by 2 meters 9 centimeters, that weigh 1.25 kilograms per square meter; and 60 carpets—size: 3 meters 57 centimeters by 5 meters, that weigh 1.05 kilograms per square meter.

20. Define an abstract base class Shape that includes protected data members for the (*x, y*) position of a shape, a public method to move a shape, and a public abstract method show() to output a shape. Derive subclasses for lines, circles, and rectangles. You can represent a line as two points, a circle as a center and a radius, and a rectangle as two points on diagonally opposite corners. Implement the toString() method for each class. Test the classes by selecting ten random objects of the derived classes, and then invoking the show() method for each. Use the toString() methods in the derived classes.

21. Define a class, ShapeList, which can store an arbitrary collection of any objects of subclasses of the Shape class.

22. Implement the classes for shapes using an interface for the common methods, rather than inheritance from the superclass, while still keeping Shape as a base class.

23. Write a program that will generate exceptions of type NullPointerException, NegativeArraySizeException, and IndexOutOfBoundsException. Record the catching of each exception by displaying the message stored in the exception object and the stack trace record.

24. Add an exception class to the last example that will differentiate between the index-out-ofbounds error possibilities, rethrow an appropriate object of this exception class in divide(), and handle the exception in main().

25. Write a program that calls a method that throws an exception of type ArithmeticException at a random iteration in a for loop. Catch the exception in the method and pass the iteration count when the exception occurred to the calling method by using an object of an exception class you define.

26. Add a finally block to the method in the previous example to output the iteration count when the method exits.

27. Write a Main method that takes the name of a text file as a command line argument and prints every line in lower case.

28. Write a main() method that counts the number of words in a text file whose name is accepted from standard input. Also print the size of a file.

29. Write a program using BufferedInputStream, FileInputStream, BufferedOutputStream, FileOutputStream to copy Content of one file File1.txt into another file File2.txt.

30. Create a class called Student. Write a student manager program to manipulate the student information from files by using FileInputStream and FileOutputStream.

31. Refine the student manager program to manipulate the student information from files by using the BufferedReader and BufferedWriter

32. Refine the student manager program to manipulate the student information from files by using the DataInputStream and DataOutputStream. Assume suitable data

33. Write a complete multi-threaded program to meet following requirements:
    a. Two threads of same type are to be instantiated in the method main.
    b. Each thread acts as a producer as well as a consumer.
    c. A shared buffer can store only one integer information along with the source & destination of the information at a time.
    d. The information produced is to be consumed by appropriate consumer.
    e. Both producers produce information for both consumers.
    f. Each thread produces 5 information.
    g. Demonstrate kill and suspended scenario for Thread.

34. Develop an applet that draws a circle. The dimension of the applet should be 500 x 300 pixels. The circle should be centered in the applet and have a

radius of 100 pixels. Display your name centered in a circle. (using drawOval() method)

35. Draw ten red circles in a vertical column in the center of the applet.

36. Built an applet that displays a horizontal rectangle in its center. Let the rectangle fill with color from left to right.

37. Develop an applet that display the position of the mouse at the upper left corner of the applet when it is dragged or moved. Draw a 10x10 pixel rectangle filed with black at the current mouse position.

38. Develop an applet that contains one button. Initialize the label on the button to "start", when the user presses the button, which changes the label between these two values each time the button is pressed.

39. Develop an applet that uses the mouse listener, which overrides only two methods which are mousePressed and mouseReleased.

40. Develop a Demo swing application with different component
    a. Using JLabel and Icon – using JApplet
    b. Using JTextField and JTextArea – using JApplet
    c. Using JButton – using JApplet
    d. Using JToggleButton – Application (JFrame)
    e. Using JRadioButton and JCheckBox – Application (JFrame)

41. Write a java application to create a Calculator using JButton and JTextField. Use Grid layout manager.

Instructional Method:

a) 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.

b) The internal evaluation will be done on the basis of continuous evaluation of students in the laboratory and class-room.

c) Practical examination will be conducted at the end of semester for evaluation of performance of students in laboratory.

d) Students will use supplementary resources such as online videos, NPTEL videos, e-courses, Virtual Laboratory

Supplementary Resources:

a) http://www.oracle.com/technetwork/java/javase/downloads/index.html
b) http://docs.oracle.com/javase/specs/jls/se7/ html/index.html
c) http://docs.oracle.com/javase/tutorial/java/index.html
d) http://www.javatpoint.com/
e) http://www.tutorialspoint.com/java/
f) http://www.learnjavaonline.org/

g) http://www.c4learn.com/javaprogramming/

h) http://www.learn-java-tutorial.com/