Department of Computer Science & Engineering

Object Oriented Programming with JAVA LAB Course Code: 23CP201P

Sr.No.	List of experiments	CO
1.	Set up and get familiar with Java programming environment; i. Install JDK, setup Java environment and write a program to print ―CODING IS FUN, ENJOY IT!. ii. Write a Java program to print the sum of two numbers.	**CO1**
2.	Study language features of Java (variables, data types, declarations, loop and branch constructs, etc.) i. You are developing a mathematical tool that requires generating a list of prime numbers. How would you implement a Java program to generate the first n prime numbers? ii. Write a program to enter two numbers and perform mathematical operations on them. iii. Write a program in Java to find maximum of three numbers using conditional operator. iv. You're working on a text analysis feature that counts the number of vowels and consonants in a given line of text. Write a program to accept a line and check how many consonants and vowels are there in line. v. 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 am vi. Java Program to Find Largest Number in an array vii. Write a java program to perform addition and multiplication of Two Matrices	**CO2**
3.	Class and Objects: study and implement classes based application using Java i. Write a program to create a "distance" class with methods where distance is computed in terms of feet and inches, how to create objects of a class. ii. Modify the "distance" class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object. iii. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword iv. Write a program that implements two constructors in the class. We call the other constructor using 'this' pointer, from the default constructor of the class. v. Write a program in Java in which a subclass constructor invokes the constructor of the super class and instantiate the values. vi. Write a program in Java to develop overloaded constructor. Also develop the copy constructor to create a new object with the state of the existing object.	**CO3**
4.	Inheritance: study and implement various types of inheritance in Java.	**CO3**

	i. Write a program in Java to demonstrate single inheritance, multilevel inheritance and hierarchical inheritance. ii. Java Program to demonstrate the real scenario (e.g., bank) of Java Method Overriding where three classes are overriding the method of a parent class. Creating a parent class. iii. Write a java program for the use of super and this keyword. iv. Write a java program for the use of final keyword.	
5.	Polymorphism: study and implement various types of Polymorphism in java. i. Write a program that implements simple example of Runtime Polymorphism with multilevel inheritance. (e.g., Animal or Shape) ii. Write a program to compute if one string is a rotation of another. For example, pit is rotation of tip as pit has same character as tip.	**CO3**
6.	Study and implement Abstract class and Interfaces in Java i. Describe abstract class called Shape which has three subclasses say Triangle, Rectangle, Circle. Define one method area() in the abstract class and override this area() in these three subclasses to calculate for specific object i.e. area() of Triangle subclass should calculate area of triangle etc. Same for Rectangle and Circle. ii. Write a Java program to create an abstract class Employee with abstract methods calculateSalary() and displayInfo(). Create subclasses Manager and Programmer that extend the Employee class and implement the respective methods to calculate salary and display information for each role. iii. Write a Java program to create an interface Shape with the getArea() method. Create three classes Rectangle, Circle, and Triangle that implement the Shape interface. Implement the getArea() method for each of the three classes.	**CO3**
7.	Study and implement Exception handling in Java i. Write a Java program for try-catch block in exception handling. ii. Write a Java for multiple catch block in exception handling. iii. Write a java program for nested of try in exception handling. iv. Write a small application in Java to develop Banking Application in which user deposits the amount Rs 1000.00 and then start withdrawing of Rs 400.00, Rs 300.00 and it throws exception "Not Sufficient Fund" when user withdraws Rs. 500 thereafter. v. Write a java program for finally block in exception handling.	**CO4**
8.	Study and implement File Handling in Java i. Read a content from file: calculate number of sentences, words and characters from the file. ii. Read content from a file convert it to uppercase and save it into another file. iii. Remove duplicate lines from a File. iv. Create a class called Student. Write a student manager program to manipulate the student information from files by using FileInputStream and FileOutputStream v. Refine the student manager program to manipulate the student information from files by using the BufferedReader and BufferedWriter vi. Write a program to manipulate the information from files by using the Reader and Writer class. Assume suitable data.	**CO4**
9.	Study and implement multi-threaded application in Java i. Write a Java program to demonstrate how to create and start a thread using both theThread class and the Runnable interface. ii. Write a Java program that illustrates thread synchronization by ensuring multiple threads can safely access a shared resource without causing data inconsistency.	**CO6**

	iii. Write a Java program to demonstrate inter-thread communication using wait(), notify(), and notifyAll() methods, allowing threads to communicate and coordinate their actions. iv. Write a Java program to show how thread priority affects the execution order of threads, highlighting the use of setPriority() and getPriority() methods. v. Write a Java program to implement the producer-consumer problem, ensuring the handling of potential deadlock conditions using proper synchronization techniques.	
10.	GUI programming using Java Applet/Swings Components and Event Handling i. Write a Java program to demonstrate various window handling events such as windowOpened(), windowClosing(), windowClosed(), windowIconified(), windowDeiconified(), windowActivated(), and windowDeactivated(). ii. Write a Java program to demonstrate various mouse handling events including mouseClicked(), mouseEntered(), mouseExited(), mousePressed(), mouseReleased(), and mouseDragged(). iii. Write a Java program to demonstrate different keyboard handling events such as keyPressed(), keyReleased(), and keyTyped(). iv. Write a Java program to create a simple GUI that includes a button and a label. When the button is clicked, the text of the label should change accordingly.	**CO5**