LAB 4

1. Create a class Work which consist of four methods with same name calculate (). If two parameter is passed calculate the sum between two parameter and display the result, if three parameter of type double is passed then find the multiplication between threeparameter and return the result, if two parameter of float is passed then calculate difference between two parameter and display the result. Now create the instance of Work and invoke all the method to demonstrate method overloading condition.
2. Write a java program that asks the user to enter number in an array of size ‘n’. then display only the numbers that are divisible by 2 and 3.
3. Make an array of integers of size 30 and store 30 integer in it then display the integers that are between 16 and 47;
4. Define string array of size 10 and insert name of 10 student in it. Then list the name of students that ends with letter ‘a’.
5. Create a class Student having data member roll, name, address, faculty, grade(string). Create suitable constructor to set the details of student and methods findDivision()that calculates the division of student based on grade achieved and return the division(if grade=A+ then distinction, if grade=A then very good, grade B+ then first division, grade B then second division , grade C then pass in individual subject and if grade D the fail), display() method that display the student details and also display the division achieved by the student in main method. Now create the instance of two students and demonstrate the above scenario.
6. How static polymorphism is achieved in java? Show a program
7. Create a String array that holds name of 5 birds and print the name of such birds whose first letter start with ‘s’.
8. Create an abstract class Calculation which has one normal method int add(int x, int y)which will calculate the sum of x and y and return the results and one abstract method int mul (int x, int y) that performs multiplication. Create a class Demo that inherit abstract class that implement all the abstract method, and it have its own method int calcDiv(int x, int y) that calculates the division between x and y and return the result. Now create the object of Demo and demonstrate the above scenario. After this access the method of abstract class using reference of abstract class.
9. Create an abstract class FMachine having methods getData() and putData(). Derive a class Airplane having instance variable code, name, capacity and methods getData()and putData() that overrides method of abstract class to read and display the result. Create some instances of Airplane and call the required methods.
10. Create an abstract class Calculator having abstract methods calcPrime(int x, int y), calcEven(int x, int y). Now create a class SimpleCalculator that inherit an abstract class and provide the implementation of abstract method