Programming Questions (II)

1. A class called **Circle** is designed as given below. It contains one private instance variable: radius (of type double). Two constructors, Two public methods: getRadius() and findArea().Define the main class and test the program.

2. Create a **Line** class with 2 points (say, x and y) as data members. Write suitable constructors. Write a method to compute the length of the line. Use the formula-
Length = $\sqrt{(x2 - x1)^2 + (y2 - y1)^2}$

3. Develop a Java application to generate Electricity bills. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading, type of EB connection (i.e. domestic or commercial). Compute the bill amount using the following tariff.
If the type of the EB connection is domestic, calculate the amount to be paid as follows:
   - First 100 units - Rs. 1 per unit
   - 101-200 units - Rs. 2.50 per unit
   - 201 -500 units - Rs. 4 per unit
   - > 501 units - Rs. 6 per unit

   If the type of the EB connection is commercial, calculate the amount to be paid as follows:
   - First 100 units - Rs. 2 per unit
   - 101-200 units - Rs. 4.50 per unit
   - 201 -500 units - Rs. 6 per unit
   - > 501 units - Rs. 7 per unit

4. Define a class '**Student_Results'** with the following attributes: Reg.No, Name, branch & CGPA. You have to define appropriate constructors and methods. A Software company directly calls the first two toppers for the HR round in each branch. Write a Java program to display the shortlisted students' details.

5. A group of 'n' candidates has applied for faculty recruitment. Their Name, qualification, experience, dob and gender are stored in a class "Recruitment". Write a java program to sort the objects based on their experience and display their details. If the experience is equal, use the name as the second sorting criteria.

6. Create a class **ComplexArithmetic** to perform Complex number arithmetic operations. A Complex object should be created in the main method enclosed within a class ComplexMain. The complex arithmetic operations should be executed in a menu-driven way.

7. Develop a class **TelephoneIndex** with two String objects as members. One should hold subscribers' names, and the other should hold their phone numbers. The class

should have appropriate constructor, input and display methods. Create an array of objects for TelephoneIndex and do the following:

a) Your program should ask the user to enter a name or the first few characters of a name to search for it in the array and display the corresponding phone number.

b) The program should display all the names that match the user's input and corresponding phone numbers.

8. Implement profit/loss calculation for each of the products provided and also the total profit/loss using inheritance in java. The following classes and class members have to be part of the solution 1) ProfitLossCalculation -- child class of productData which implements the function calculate() to calculate the profit or loss 2) super class -- ProductData -- which stores the product information and implements printProduct() to print product information.

9. Create a class **CARD** with cardno, cust_name, bank_name as data members. Provide suitable constructors to create a CARD object. Create another class **Creditcard** that inherits the properties of CARD. In addition to the CARD properties, **Creditcard** will have an additional 'limit' data member. Provide suitable constructors. Include methods 'display' and 'use'. Display method should display the details of a credit card. The use method in Creditcard should decrease the limit by the amount used. Make sure that the user never exceeds the available limit. Create a demo class to test for 'n' Creditcard objects.

10. Write an inheritance hierarchy for classes Quadrilateral, Trapezoid, Parallelogram, Rectangle and Square. Use Quadrilateral as the superclass of the hierarchy. Create and use a Point class to represent the points in each shape. Make the hierarchy as deep (i.e., as many levels) as possible. Specify the instance variables and methods for each class. The private instance variables of the Quadrilateral should be the x-y coordinate pairs for the four endpoints of the Quadrilateral. Write a program that instantiates objects of your classes and outputs each object's area (except Quadrilateral).

11. The interface GCD contains an abstract method computeGCD(int num1, int num2). Class APPROACH1 implements the interface by following Euclid's algorithm and class APPROACH2 implements the interface by listing all the factors (need not be prime factors) of the two numbers and choosing the highest common factor. Write a Java program to do the above-said operations.

12. Write an abstract class Special with an abstract method double process (double P,double R). Create a subclass Discount and implement the process() method. Return the process() method with the following formula: total=P+P*R/100. Return the total.

13. Create a package called pack1. Add two classes Sum and Difference (calculate the sum and difference of two numbers) to it. Create a subpackage called subpack1. Add two classes Product and Quotient (calculate the product and quotient of two numbers).

Write a program to read values from the user and perform the arithmetic operations using the package classes.

14. Create an interface with methods add () and sub () in a package called 'pack1'. Create another package, 'pack2' with an interface with methods multiply () and divide (). By implementing both interfaces, write a main class to perform arithmetic operations on integer numbers.