Lab-02

Simple assignments on classes and objects:

1. You need to create a class called “Student” which contains a variable called studentName, enrollNo, collegeName, mobileNo, and city. You need to assign the value of studentName, enrollNo, collegeName, mobileNo, and city by making 3 objects of Student class.

E.g., Student 1 --- studentName --- Anil Gupta

enrollNo, --- 1

collegeName, --- LNMIIT

mobileNo, --- 9991230987

city --- Delhi

Case (1). You need to pass your values in “hardcoded fashion”.

Case (2). You need to pass your values through user input.

You need to print in the following format

The details of first student are:

Anil Gupta

1

LNMIIT

9991230987

Delhi

The details of second student are:

xxxx

The details of third student are:

xxxx

2. Write a java program to print the area of a rectangle by creating a class named “ComputeArea,” which contains two functions/methods. The first function/method is

“setDimensions,” which takes length and breadth of a rectangle as an argument. The second function/method is “getAreaofTriangle,” which will return the area. The length and breadth of a rectangle should be entered through the keyboard.

3. Write a java program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle'. Also, you need to create another class called Test which will call your area of triangle compute function.

4. Write a program that would print the information (name, year of joining, salary, address) of three employees by creating a class named 'Employee'. The output should be as follows:

Name	Year of joining	Address
Robert	2004	1C- HallStreat
James	2006	68D- WallStreat
Jack	2007	26B- Nagels

5. Write a java program to add two complex numbers.

6. Write a Java program that performs the following steps:

1. declare int variable called i

2. declare a double variable called piValue and initialize it with value of pi

3. Calculate the smallest integer that is bigger than the square of the value in piValue, and assign the resulting value to i.

7. Determine what will be shown on screen if the following statements are executed.

```
float f = 500F, g = 500F;
System.out.println(f/0);
System.out.println(-f/0);
System.out.println((f-500)/(f-g));
System.out.println(-(f-500)/(f-g));
```

8. Write a Java program that prompts for and accepts two numbers, a and b, via keyboard, and prints out the results of the following numeric computation:

$$a+b,\ a\times b,\ \frac{a}{b},\ a^b,\ \text{and}\ \sqrt[b]{a}$$

9. Write a class, Commission, which has an instance variable, sales; an appropriate constructor; and a method, commission() that returns the commission. Now write a demo class to test the Commission class by reading a sale from the user, using it to create a

Commission object after validating that the value is not negative. Finally, call the commission() method to get and print the commission. If the sales are negative, your demo should print the message “Invalid Input”.

10. Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as instance variables—a part number (type String), a part description (type String), a quantity of the item being purchased (type int) and a price per item (double). Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable. In addition, provide a method named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a double value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0.0. Write a test application named InvoiceTest that demonstrates class Invoice’s capabilities.

11. There is a class called Account that contains below-mentioned features:

(A) Two constructors:

(1). First one can be utilized for initializing: - account name holder, account number, and initial amount at which an account is open.

(2). Second one will be used to initialize: - account name holder, account number, address, type of account, and current balance.

(B). Methods:

(1). Deposite (), Withdraw (), and getBalance()

Make necessary assumptions for data members and return types of methods. Create objects of this class and use them. Also, utilize the concept of array of references to store the Account objects and print those objects on the screen by creating another method called printCust_Info().