CSE 2001: Data Structure & Algorithms

Programming Assignment-II

(Object-Oriented Design)

Question-1:

A phone number can be thought of as having three parts: the area code, the exchange code and the number.

Example: A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767) and the number (8900).

Define a class **Phone** to store these three parts of a phone number separately as instance variable (*area_code*, *exchange*, *number*). The class consists of two member methods: *input ( )* and *display ( )*. Create two class objects of type phone. Initialize one directly (by using dot (.) operator and instance variable with value `area_code: 212, exchange: 767, number: 8900`), and initialize other from the user through member method *input ( )*. Display both the numbers.

Sample Run:

```
My number is (212) 767-8900

Your number is (415) 555-1212
```

Question-2:

Define a class called **Complex** with instance variables *real*, *imag* and instance methods *setData()*, *display()*, *add()*. Write a Java program to add two complex numbers.

The prototype of add method is:

public Complex add(Complex, Complex).

Question-3:

Define a class called **Student** with instance variables *Roll*, *Name*, *DSA_Mark*. Also, the class consists of instance methods *getdata( )*, *showdata( )* to provide input to the instance variable and to display the value of instance variable. Write a program to create the details of 5 students. Display the information of the students who has secured the highest *DSA_Mark*.

Question-4:

Define a class called **product**. Two instance variables of the class are *pid* (product-id) and *price* (product price). It consists of one static variable *tot_price* (total price). Initialize the value of instance variables through parameterized constructor. It consists of a *display ( )* method to display the value of instance variables. A person went to market and purchase 5 different products. Using the above mentioned class, display the details of products that the person has purchased. Also, determine how much total amount the person will pay for the purchase of 5 products.

Question-5:

Define a class **Deposit**. The instance variable of the class **Deposit** are mentioned below.

Instance variable	Datatype
Principal	Long
Time	Integer
rate	Double
Total_amt	Double

Initialize the instance variables Principal, Time, rate through constructors. Constructors are overloaded with the following prototypes.

Constructor1: `Deposit ( )`
Constructor2: `Deposit (long, int, double)`
Constructor3: `Deposit (long, int)`
Constructor4: `Deposit (long, double)`
Apart from constructor, the other instance methods are (i) `display ( )`: to display the value of instance variables, (ii) `calc_amt( )` to calculate the total amount.

```
Total_amt = Principal + (Principal×rate×Time)/100;
```

Question-6:

Create an abstract class **Shape** and the derived classes **Square**, **Triangle** and **Circle**. Write a java program to display area of different shapes.

Question-7:

Define a base class **Person** with instance variable name, age. The instance variables are initialized through constructors. The prototype of constructor is as below.
Person (string, int)
Define a derived class **Employee** with instance variables Eid, salary. The instance variables are initialized through constructors. The prototype of constructor is as below.
Employee (string, int, int, double). Another instance method of Employee class is empDisplay() to display the information of employee details.

Question-8:

Define an interface **DetailInfo** to declare methods *display ( )* & *count ( )*. Another class **Person** contains a static data member *maxcount*, instance member *name* & method *display ( )* to display name of a person, count the no. of characters present in the name of the person.

Question-9:

Write a Java program to declare a Class named as **Student** which contains *roll number, name* and *course* as instance variables and *input_Student ()* and *display_Student ()* as instance methods. A derived class **Exam** is created from the class **Student**. The derived class contains *mark1*, *mark2*, *mark3* as instance variables representing the marks of three subjects and *input_Marks ()* and *display_Result ()* as instance methods. Create an array of objects of the **Exam** class and display the result of 5 students.

Question-10:

Design a package that contains two classes **Student** & **Test**. The Student class has data members as name, roll and instance methods input ( ) & output ( ). Similarly the **Test** class has data members as mark1, mark2 and instance methods input ( ), output ( ), **Student** is extended by **Test**. Another package carry interface **Sports** with 2 attributes score1, score2. Find grand total mark & score in another class.

Home Assignment

Question-1:

A sales person is paid commission based on the sales he makes as shown by the following table:

SALES	COMMISSION
Under Rs. 100	2% of SALES
Rs 500 and under Rs 5000	5% of Sales
Rs 5000 and above	8% of sales

Write a class, **Commission**, which has:

- An instance variable, *sales*; an appropriate constructor; and a method, *getCommission()* that returns the commission.

Now write a **Demo** class in Java 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 *getcommission()* method to get and print the commission.

If the sales are negative, your **Demo** class should print the message “*Invalid Input*”.

Question-2:

Define a class called Book with instance variables BName, BEdition, BPrice. Use constructor to initialize the instance variables of the class. Another instance method display ( ) to display the book information. A person purchase 5 different books. Display the book details which has the maximum price.

Question-3:

Write a Java code that would represent Distance object (meters, centimetres) using classes. The class definition include a parameterized constructor, display ( ) to display the instance variables, sum (Distance, Distance) to add two distances. Now write a main function that creates a couple of Distance objects and demonstrates the addition of two distances.

Question-4:

A point in the *x-y* plane is represented by its *x-coordinate* and *y-coordinate*. Design a class, **PointType** in Java, that can store and process a point in the *x-y* plane. You should then perform operations on the point, such as showing the point, setting the coordinates of the point, printing the coordinates of the point, returning the *x-coordinate*, and returning the *y-coordinate*. Every circle has a centre and a radius. Given the radius, we can determine the circle's area and circumference. Given the centre, we can determine its position in the *x-y* plane. The centre of a circle is a point in the *x-y* plane. Design a class, **CircleType** that can store the radius and centre of the circle. Because the center is a point in the x-y plane and you designed the class to capture the properties of a point from **PointType** class. You must derive the class **CircleType** from the class **PointType**. You should be able to perform the usual operations on a circle, such as setting the radius, printing the radius, calculating and printing the area and circumference, and carrying out the usual operations on the center.

Question-5:

Create a **RestaurantMeal** class that holds the name and price of a food item served by a restaurant. Its constructor requires arguments for each field. Create a **HotelService** class that holds the name of the service, the service fee, and the room number to which the service was supplied. Its constructor also requires arguments for each field. Create a **RoomServiceMeal** class that inherits from both RestaurantMeal and HotelService. Whenever you create a **RoomServiceMeal** object, the constructor assigns the string "*room service*" to the name of the service field, and Rs .24.00 is assigned to the service fee inherited from *HotelService*. Include a RoomServiceMeal method that displays all of the fields in a RoomServiceMeal by calling display functions from the two parent classes. Additionally, the display function should display the total of the meals plus the room service fee. In a main() function, instantiate a RoomServiceMeal object that inherits from both classes.

For example:
A "steak dinner" costing Rs. 600 is a "room service" provided to room 1202 for a Rs 24.00 fee.

****