CSE 2001: Data Structure & Algorithms

Programming Assignment-II

(Object-Oriented Design)

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

SALES	COMMISSION
Under Rs. 500	2% of SALES
Rs. 500 and under Rs. 5000	5% of SALES
Rs. 5000 and over	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*".

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).*

3. 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.

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 center and a radius. Given the radius, we can determine the circle's area and circumference. Given the center, we can determine its position in the *x-y* plane. The center of a circle is a point in the *x-y* plane. Design a class, **CircleType** that can store the radius and center 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.

5. Let a class **Person** contains data members ***name*** and ***age***. A constructor with two arguments is used to assign name and age. Person is of two types a) Student and b) Teacher. class **Student** contains data members i)*course* ii) *Roll Number* and iii)*Marks* and method *display()* to display data related to student. Similarly, class **Teacher** contains data members i) *subject_assigned* (May take this as a String) ii) *contact_hour* and method *display ()* to display data related to teacher. Implement this program using base class constructor in derived class.

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

7. Define an interface to declare methods *display ( )* & *count ( )*. Another class 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.

8. Define an interface **EmpInterface** (*void displayEmp(), void giveBonus(double amount)*). Define an abstact class **Employee** (*empID, fName, lName, salary*). Define a concrete class **Manager** (*bonus*) subclass of **Employee** and define the interface methods. Perform the followings:

    a. Define the appropriate constructor in class hierarchy

    b. Ensure the bonus amount should not be negative and zero using exception handling

    c. Create an array of interface reference variables and populate with manager objects.

    d. Write a **Test** class to implement the above said requirements of interface implement ation and exception handling.

9. 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.

****