Programming Assignment-V

(Class, Object & Methods)

1. A sales person is paid commission based on the sales as shown:

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()* which returns the commission amount.

Write a Demo class in Java to test the Commission class by reading a sales amount from the user and use this sales amount to create a Commission object.

Finally call the *getCommission()* method to get and print the commission amount. 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).

Note: Complex number is of the form $a+ib$, where a is real part and b is imaginary part.

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 objects of the Exam class and display the result of 3 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. Perform operations on the point such as 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 Student and Teacher.

Class Student contains data members like *course, Roll Number* and *Marks* and method *display()* to display data related to student. Similarly class Teacher contains data members like *subject_assigned* ,

contact_hours and method *display ()* to display data related to teacher. Implement this program using base class constructor in derived class.

6. Write a program in Java to count and display the number of objects created for a given class using constructor.

7. Write a program in Java to compute the sum of two time objects and also sum of three time objects using method overloading and display the time in the form of hour, minute and seconds.

8. Create a class **Book** to represent a book's details. The class should include:

1. **Attributes**:
   title (String): Title of the book.
   author (String): Author of the book.
   price (double): Price of the book.
2. **Constructor**:
   Accepts **title** and **author** to initialize the object.
3. **Method setPrice(double price)**:
   Sets the price of the book.
   Call **this.showDetails()** from within this method.
4. **Method showDetails()**:
   Prints the book **title**, **author**, and **price**.

Write a main class to create a Book object, set the book price, and display all the details.

9. Define a class **Loan** with instance variables: **principal**, **rate**, **time**, and **simple_interest**. Implement:

1. **Constructor**: Initializes **principa**l, **rate**, and **time** using the **this** keyword.
2. **calculateInterest()**: Computes **simple_interest = (principal * rate * time) / 100** and stores it in the **simple_interest** variable and prints the value.

Write a main class to test the Loan class.

10. Write a Java program to create a **BankAccount** class using **constructor overloading** to model different types of bank accounts:

1. **Basic Account** – Initialized with account holder's name and balance.
2. **Savings Account** – Initialized with name, balance, and interest rate.
3. **Checking Account** – Initialized with name, balance, minimum balance, and overdraft limit.
4. **Business Account** – Initialized with name, balance, interest rate, and overdraft limit.

Each constructor should set appropriate defaults based on the account type.

Add methods to:

- Withdraw and deposit money with appropriate balance checks.
- Display account details.

Write a BankTest class to create different accounts and demonstrate the functionality.

11. Create a Product class with private fields for *productId, productName, price*, and *quantity*. Use:

- **Constructor Overloading**: One constructor with productId and productName, another with price and quantity as well.
- **Method Overloading**: setPrice to set either a price or adjust by a percentage, and setQuantity to set or add stock.
- **this Keyword**: Use this to reference fields.

Write a ProductTest class to demonstrate these features and display product details.

12. Create a Book class with private fields for title, author, isbn, and availableCopies.

- **Constructor Overloading**: Default constructor and one that sets all fields.
- **Method Overloading**: checkOut to reduce copies, and returnBook to increase copies, with one version of each for single and multiple copies.
- **Encapsulation**: Private fields with **get** and **set** methods.

Write a LibraryTest class to demonstrate book checkouts, returns, and display book details.

13. Create a class Employee with instance variables*: name, eid,* and *gross_salary*. Implement:

1. **Constructor**: Initializes name and eid.
2. **GrossSalary(int basic)**: Calculates gross_salary = basic + HRA + DA (where HRA is 30% of basic and DA is 24% of HRA).
3. **display()**: Displays employee details.

Write a main class to test the Employee class.