

Assignment-7

Object Oriented Programming Lab

Java: Polymorphism, Abstract Class, and Interface

1. You are required to write a base class Student having details (name, registration number, father's name, address (current address, permanent address), and contact details (phone number, email Id). UGStudent, PGStudent, and PhDStudent extend the Student class they have details about the past qualifying degree and marks. Make each attribute protected and provide suitable getter and setter to access attributes. **Use the Java Abstract class concept.**

2. **Create an interface** Vehicle having three methods changeGear, speedup, and applyBrakes. Two other classes, Bike and Bicycle, implement the Vehicle interface. These classes have two fields, i.e., gear and speed. The changeGear method accepts the value of gear and assigns it to the gear field. The speedup method accepts the value by which the speed field should be increased. The applyBrakes method accepts the value by which the speed field should be decreased. Create a Test class to demonstrate each of these classes.

3. Write a class Employee having attributes: employeeID, name, department, dob, designation, yearOfJoining, and phoneNumber. Add a function to print details of employee using employeeID. Two classes, i.e., Faculty and **OfficeStaff**, extends the Employee class. The Faculty has a list of subjects and labs taught by them. **OfficeStaff** has a list of skills which stores skill such as Typing, Technician, etc. Each of the above classes has appropriate getter setter for inserting skill in the list and printing the list of skill. Demonstrate above class using a Test class which creates 5 employees of different types and print their details. **Use Java Abstract class concept.**

4. **Create an interface** Shape having two methods getArea and getPerimeter. Three classes, Circle, Triangle, and Rectangle, implement the Shape interface, and **override** the two methods. Create a Test class to demonstrate each of these classes.

5. Painting Shapes

Develop a class hierarchy of shapes and write a program that computes the amount of paint needed to paint different objects. The hierarchy will consist of a parent class Shape with three derived classes - Sphere, Rectangle, and Cylinder. For the purposes of this exercise, the only attribute a shape will have is a name and the method of interest will be one that computes the area of the shape (surface area in the case of three-dimensional shapes). Do the following.

1. Write an abstract class Shape with the following properties:
 - >An instance variable shapeName of type String
 - >An abstract method area()
 - >A toString method that returns the name of the shape

2. **Class Sphere** for a sphere which is a descendant of Shape. A sphere has a radius and its area (surface area) is given by the formula $4 \cdot \text{PI} \cdot \text{radius}^2$. Define similar classes for a rectangle and a cylinder. Both the Rectangle class and the Cylinder class are descendants of the Shape class. A rectangle is defined by its length and width and its area is length times width. A cylinder is defined by a radius and height and its area (surface area) is $\text{PI} \cdot \text{radius}^2 \cdot \text{height}$. Define the toString method in a way similar to that for the Sphere class.

3. **Class Paint** for a type of paint (which has a "coverage" and a method to compute the amount of paint needed to paint a shape). Correct the return statement in the amount method so the correct amount will be returned. Use the fact that the amount of paint needed is the area of the shape divided by the coverage for the paint.
(NOTE: Leave the print statement - it is there for illustration purposes, so you can see the method operating on different types of Shape objects.)

4. **Class Paintthings** Computes the amount of paint needed to paint various shapes. A paint object has been instantiated. Add the following to complete the program:

- >Instantiate the three shape objects: deck to be a 20 by 35 foot rectangle, bigBall to be a sphere of radius 15, and tank to be a cylinder of radius 10 and height 30.
- >Make the appropriate method calls to assign the correct values to the three amount variables.

Run the program and test it. You should see *polymorphism* in action as the amount method computes the amount of paint for various shapes.