Practice Lab Assignment 3

For this Practice Lab Assignment, you will write programs based on the concepts of Inheritance.

Instructions

- There are 5 questions in this assignment.
- Do not share your work with anyone.
- Discuss with TA in case of any further clarifications.

Due Date: 12th September 2024, Midnight.

Submission Guidelines

1. You will submit (upload) this assignment in Blackboard. Email submissions will not be accepted.
2. Upload the .zip file containing all questions.
3. Name the pdf file as "John_Doe_2010110999" in case your name is "John Doe" and your roll number is "2010110999".

Questions

1. Implement a class of shapes, make subclasses of circle, rectangle and triangle, using respective getter and setter functions. Make the respective functions of each subclass to find areas. Initialize three variables of type "shape" and initialize respective variables with a radius, a length & breadth, and a set of three sides and print respective areas.

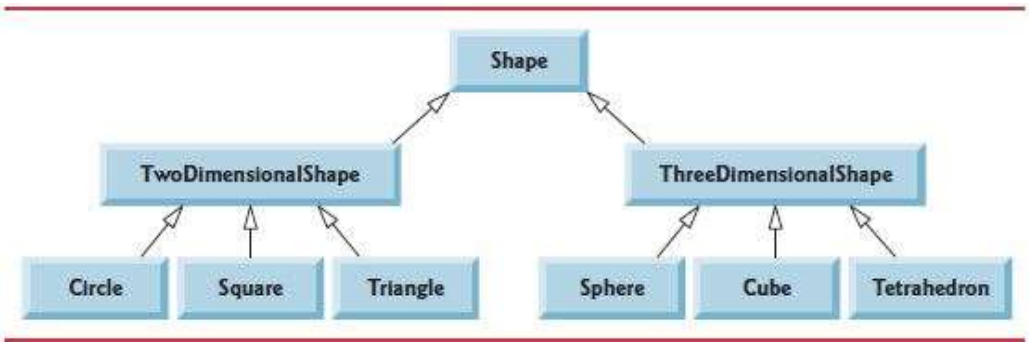
Comment on the runtime behavior of the variables of type "shape".

2. Design a class named **Triangle** to represent a triangle. The class contains:

- Three **double** data fields named **side1**, **side2**, and **side3** that specify the three sides of the triangle. The default values are **1** for all the sides.
- A no-arg constructor that creates a default triangle.
- A constructor that creates a triangle with the specified sides.
- The accessor (get) and mutator (set) methods for all the data fields.
- A method named **getArea()** that returns the area of this triangle.
- A method named **getPerimeter()** that returns the perimeter.

Implement the class. Write a test program that creates two **Triangle** objects. Assign sides **4**, **5**, and **6** to the first object and **1.5**, **2.5**, and **3.5** to the second object. Display the properties of both objects and find their areas and perimeters.

3. Implement the Shape hierarchy shown in the figure.



Each **TwoDimensionalShape** should contain method **getArea()** to calculate the area of the two-dimensional shape. Each **ThreeDimensionalShape** should have methods **getArea()** and **getVolume()** to calculate the surface area and volume, respectively, of the three-dimensional shape. Create a program that uses an array of Shape references to objects of each concrete class in the hierarchy. The program should print a text description of the object to which each array element refers. Also, in the loop that processes all the shapes in the array, determine whether each shape is a **TwoDimensionalShape** or a **ThreeDimensionalShape**. If it's a **TwoDimensionalShape**, display its area. If it's a **ThreeDimensionalShape**, display its area and volume.

4. Design a class named **Fan** to represent a fan. The class contains:

- Three constants named **SLOW**, **MEDIUM**, and **FAST** with values **1**, **2**, and **3** to denote the fan speed.
- An **int** data field named **speed** that specifies the speed of the fan (default **SLOW**).
- A **boolean** data field named **on** that specifies whether the fan is on (default **false**).
- A **double** data field named **radius** that specifies the radius of the fan (default **5**).
- A string data field named **color** that specifies the color of the fan (default **blue**).
- A no-arg constructor that creates a default fan.
- The accessor (get) and mutator (set) method for all four data fields.
- A method named **tostring()** that returns a string description for the fan. If the fan is ON, the method returns the fan speed, color, and radius in one combined string. If the fan is not ON, the method returns fan color and radius along with the string "fan is off" in one combined string.

Implement the class. Write a test program that creates two **Fan** objects. Assign maximum speed, radius **10**, color **yellow**, and turn it on to the first object. Assign medium speed, radius **5**, color **blue**, and turn it off to the second object. Display the objects by invoking their **toString** method.

5. A small airline has just purchased a computer for its new automated reservations system. You've been asked to develop the new system. You're to write an application to assign seats on each flight of the airline's only plane (capacity: 10 seats). Your application should display the following alternatives: Please type 1 for First Class and

Please type 2 for Economy. If the user types 1, your application should assign a seat in the first class section (seats 1–5). If the user types 2, your application should assign a seat in the economy section (seats 6–10). Your application should then display a boarding pass indicating the person's seat number and whether it's in the first-class or economy section of the plane.

Use a one-dimensional array of primitive type boolean to represent the seating chart of the plane. Initialize all the elements of the array to false to indicate that all the seats are empty. As each seat is assigned, set the corresponding element of the array to true to indicate that the seat is no longer available. Your application should never assign a seat that has already been assigned. When the economy section is full, your application should ask the person if it's acceptable to be placed in the first-class section (and vice versa). If yes, make the appropriate seat assignment. If no, display the message "Next flight leaves in 3 hours".