

CSE 271 Lab 7 – Interfaces
Spring 2022
Assigned: 3/7/2022
Due: 3/15/2022

Introduction:

In this lab, we will practice utilizing interfaces. Also, you will implement a Tester class to ensure that your classes and interfaces are working as intended. Specifically, in this lab assignment, you will have one interface that describes a set of expected behaviors that the inherited classes must implement. The interface is called **Shape** and the two classes that will implement the interface are called **Rectangle** and **Triangle**. Create a project in your Eclipse IDE called Lab7 and code the following classes and interfaces.

Interface Shape:

Create an interface named **Shape** which has the following abstract methods. Recall that abstract methods do not have an implementation when you define them. Only the classes that implement the interface will provide the implementations. Note that if you provide JavaDoc in the interface for these abstract methods, you do not need to create JavaDoc in the class where you implement the method and simply use the { @inheritDoc } for the JavaDoc block.

- **public double getArea()** – Computes the area of a geometric shape.
- **public void scale(double factor)** – Scales the geometric shape's measurements (length, width, height, etc.) by the factor received as a parameter.

Rectangle Class Implements Shape:

Create a class named **Rectangle** which implements the interface **Shape** and has the following private instance properties and public interface.

1. Instance Properties:
 - a. **int x** – The x coordinate of the upper left corner of the Rectangle.
 - b. **int y** – The y coordinate of the upper left corner of the Rectangle.
 - c. **double length** – The length of the Rectangle.
 - d. **double width** – The width of the Rectangle.
2. Public Interface:
 - a. **public Rectangle(int x, int y, double length, double width)** – A workhorse constructor which initializes the instance properties using the received parameters.
 - b. **public double getArea()** – The implementation of the abstract getArea() method. It returns the area of the Rectangle based on the length and width. Note, the area of a Rectangle is length * width.
 - c. **public void scale(double factor)** – The implementation of the abstract scale(double factor) method. It scales the length and width instance properties by a scaled factor received from the parameter. Multiple the length and width instance properties by the factor (this is scale transformation).
 - d. **public boolean equals(Object o)** – This method returns true if two Rectangle objects have the same length and width and false, otherwise.

- e. **public String toString()** – Returns a String representation of the Rectangle object that include the x, y, length, and width instance properties. An example String is as follows: “Rectangle [x = 10, y = 15, length = 60.0, width = 75.0]”.
- f. **A set of getters and setters for each instance property.**

Triangle Class Implements Shape:

Create a class named **Triangle** which implements the interface **Shape** and has the following private instance properties and public interface.

1. Instance Properties:
 - a. double base – The base of the Triangle.
 - b. double height – The height of the Triangle.
2. Public Interface:
 - a. **public Triangle(double base, double height)** - A workhorse constructor which initializes the instance properties using the received parameters.
 - b. **public double getArea()** – The implementation of the abstract getArea() method. It returns the area of the Triangle based on the base and height. Note, the area of a Triangle is $0.5 * \text{base} * \text{height}$.
 - c. **public void scale(double factor)** – The implementation of the abstract scale(double factor) method. It scales the base and height instance properties by a scaled factor received from the parameter. Multiple the base and height instance properties by the factor (this is scale transformation).
 - d. **public boolean equals(Object o)** – This method returns true if two Triangle objects have the same base and height and false, otherwise.
 - e. **public String toString()** – Returns a String representation of the Triangle object that include the base and height instance properties. An example String is as follows: “Triangle [base = 30.0, height = 45.0]”.
 - f. **A set of getters and setters for each instance property.**

ShapeTester Class:

In the main method of your ShapeTester class, you should test the classes and their corresponding methods listed above. That is, you should test the constructors, getArea(), scale(), equals(), and toString() methods of both the Rectangle and Triangle classes.

Additional Notes:

- Make sure you include JavaDoc comments for all methods and classes including parameter and return descriptions.
- Make sure that all classes are named correctly.
- There is no explicit validation checking needed for this Lab assignment.

Submission Instructions:

After you have completed the lab assignment, locate your source code (**Shape.java**, **Rectangle.java**, and **Triangle.java**) in your workspace and submit it to the corresponding Lab 7 assignment’s CODE plugin.

Rubric:

Task	Grade
Interface Shape	
Correctly create the Shape interface	2
Declared abstract getArea() method	2
Declared abstract scale() method	2
Class Rectangle	
Implements the Shape interface	6
Constructor correctly implemented	6
Getters and setters correctly implemented	6
equals() method correctly implemented	6
toString() method correctly implemented	6
getArea() method correctly implemented	7
scale() method correctly implemented	7
Class Triangle	
Implements the Shape interface	6
Constructor correctly implemented	6
Getters and setters correctly implemented	6
equals() method correctly implemented	6
toString() method correctly implemented	6
getArea() method correctly implemented	7
scale() method correctly implemented	7
Adequate JavaDoc included for all classes and methods and followed the Miami University coding guidelines	6
Total	100