

Lab Assignment 07, Object-Oriented Programming, CSE 271, Spring 2020
Department of Computer Science and Engineering, Miami University

Interface

In this lab, you will practice how to define and implement interfaces. You will also implement a Tester class with the main method and write Javadoc comments for the classes and methods. In this programming assignment, we have one interface, Shape, which defines the expected behaviors (public interface) of all interested classes (Rectangle and Triangle). The classes implement the interface, Shape, to provide actual implementation of the expected behaviors. Today, we are going to define one interface, Shape, and two classes, Rectangle and Triangle which implement the interface, Shape. Create a project in Eclipse named **Lab_07**.

Important Note: Look at the slide 81 of the “lecture_inheritance_interface” slides on Canvas to see how to declare an interface and implement it in a class.

- **Interface Shape:**
Create an interface named **Shape** which has the following abstract methods:
Important Note: The following methods are abstract methods, which means you do not need to provide any method definition. End the method header/signature with a semicolon. The classes which implements this method is going to provide their own definition for this method. Also, if you write Javadoc comment for this method in the interface then you don't need to make Javadoc comments in the class where you implement this interface.
 - **double getArea()**
It computes the area of a geometric shape. Write Javadoc comment for this method.
 - **void scale(double factor)**
It scales the shape's measurements (width, length, base, height, etc.) by the factor received as a parameter.
- **Class Rectangle:**
Create a class named **Rectangle** implements the interface **Shape** and has the following information:
 - **int x**, the x coordinate of the upper left corner of the rectangle
 - **int y**, the y coordinate of the upper left corner of the rectangle
 - **double length**, the length of the rectangle.
 - **double width**, the width of the rectangle.The class should also have the following methods:
 - **public Rectangle(int x, int y, double length, double width)**
One parameterized constructor, which initializes the instances variables using the received parameters.
 - **public double getArea()**
Provides a definition of the getArea() method. It calculates the area of the rectangle based on the length and width. Area of a rectangle is *length* times *width*.
 - **public void scale(double factor)**
Provides a definition of the scale() method. It makes a scaling transformation based on the received scaling factor. Multiply *width* and *length* by factor to perform scaling transformation.
 - **public boolean equals(Object otherObject)**
Override the equals() method. It returns true if the two rectangle objects have the same length and width. False, otherwise.
Important Note: You have to first check whether the received parameter is null or not. Then check to see whether the current object and the received parameter are objects of the same class. If yes, then cast it to desired class object and compare using the instance variables.

Lab Assignment 07, Object-Oriented Programming, CSE 271, Spring 2020

Department of Computer Science and Engineering, Miami University

Interface

- **public String toString()**
Override the toString() method. The method returns a String representing the rectangle object that includes x, y, length and width. It returns a String like "Rectangle [x = 10, y = 15, length=60.0, width=75.0]".
- **Class Triangle:**
Create a class named **Triangle** implements the interface **Shape** and has the following information:
 - **double base**, the base of the triangle.
 - **double height**, the height of the triangle.The class should also have the following methods:
 - **public Triangle(double base, double height)**
One parameterized constructor, which initializes the instances variables using the received parameters.
 - **public double getArea()**
Provides a definition of the getArea() method. It calculates the area of the triangle based on the base and height. Area of a triangle is *0.5 times base times height*.
 - **public void scale(double factor)**
Provides a definition of the scale() method. It makes a scaling transformation based on the received scaling factor. Multiply *base* and *height* by *factor* to perform scaling transformation.
 - **public boolean equals(Object otherObject)**
Override the equals() method. It returns true if the two triangle objects have the same base and length. False, otherwise.
Important Note: You have to first check whether the received parameter is null or not. Then check to see whether the current object and the received parameter are objects of the same class. If yes, then cast it to desired class object and compare using the instance variables.
 - **public String toString()**
Override the toString() method. The method returns a String representing the triangle object that includes length and width. It returns a String like "Triangle [base=30.0, height=45.0]".

Javadoc Style Comments:

You have to make Javadoc style comments for all methods, interface, and classes including parameter and return description. Once you write all the Javadoc style comments generate Javadoc using Eclipse's "Generate Javadoc" option from the "project" menu. Include the Javadoc folder "doc" with your code submission.

Tester Class:

Now write another class named **ShapeDriver** with a main method to test your classes. In the **driver** class you have to test all the methods (constructor, getArea, scale, equals, toString) of the classes you defined (Rectangle and Triangle).

Your test code should include the following:

- **Testing Rectangle Class**
 - Create two rectangle objects with different widths and heights
 - Print information using toString() method for both rectangles
 - Print out the areas of the two rectangles
 - Scale each rectangle by different factors (0.5 and 2)
 - Print information using toString() method for both scaled rectangles

Lab Assignment 07, Object-Oriented Programming, CSE 271, Spring 2020
Department of Computer Science and Engineering, Miami University

Interface

- Print out the areas of the scaled rectangles
- Test equals() method by doing the following:
 - Compare two rectangle objects with different lengths and widths
Expected output: False
 - Compare two rectangle objects with same length and width
Expected output: True
 - Compare rectangle object with triangle object
Expected output: False
 - Compare rectangle object with null
Expected output: False

Repeat the above steps but this time for testing the triangle Class.

- **Testing Triangle Class**
 - Create two triangle objects with different widths and heights
 - Print information using toString() method for both triangles
 - Print out the areas of the two triangles
 - Scale each triangle by different factors (0.5 and 2)
 - Print information using toString() method for both scaled triangles
 - Print out the areas of the scaled triangles
 - Test equals() method by doing the following:
 - Compare two triangle objects with different bases and heights
Expected output: False
 - Compare two triangle objects with same base and height
Expected output: True
 - Compare triangle object with rectangle object
Expected output: False
 - Compare triangle object with null
Expected output: False

Important Note:

Make sure the file name is correct and code is well commented. No late submission. You will get zero for late submission.

Submission:

Submit java files and Javadoc folder "doc" to the appropriate submission folder on the Canvas by the due time. You can zip all the java files and "doc" folder together and submit one zip file.

Lab Assignment 07, Object-Oriented Programming, CSE 271, Spring 2020

Department of Computer Science and Engineering, Miami University

Interface

Grading Rubric:

Interface Shape	
Interface declaration	2
getArea() method declaration	2
scale() method declaration	2
Class Rectangle	
Implements the interface Shape	4
Constructor	2
getArea() method	5
scale() method	5
equals() method	5
toString() method	5
Class Triangle	
Implements the interface Shape	4
Constructor	2
getArea() method	5
scale() method	5
equals() method	5
toString() method	5
Testing through driver class	
Test constructors (each worth 1 points)	2
Test equals and toString methods (each worth 1 points)	4
Test getArea, scale methods (each worth 2 points)	8
Javadoc style comments	6
Submitted correct Javadoc files	2
Total	80