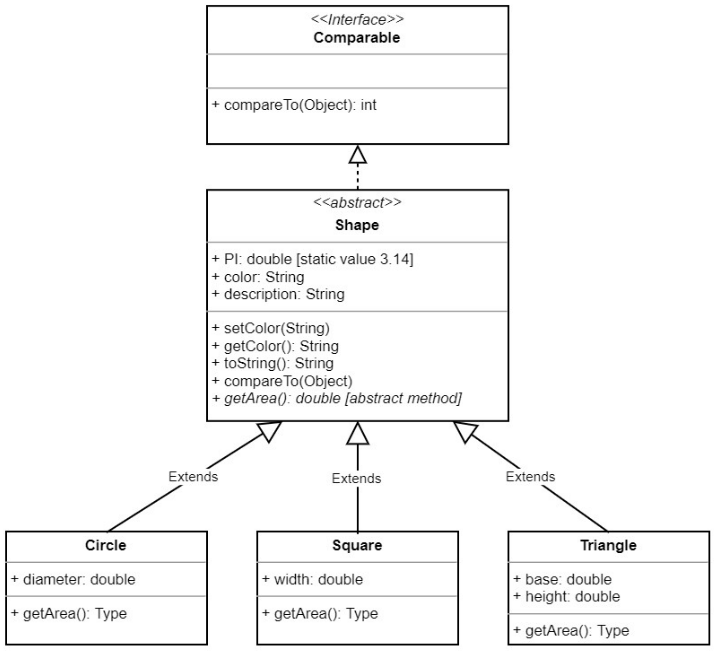
ITCS 209 Object Oriented Programming	Name:	**Lab**	**Challenge Bonus**	**Peer Bonus**
	ID:			
	Sec:			

Lab09: Interface, HashMap, and Regular Expression



In this lab, you are provided with the five java files as follows:

- **Comparable.java**: This is an *interface* class. **DO NOT MODIFY THIS CLASS!**
- **Circle.java**, **Square.java**, and **Triangle.java**: extend the Shape class. Each class contains its unique attributes and overridden the getArea() method. **DO NOT MODIFY THESE CLASS**
- **ShapeTester.java**: This is the main program for testing. **You have to complete this class**.

Task 1: Implement "Shape" Abstract Class (As shown in the class diagram above)

Create *Shape.java* file contains an abstract Class Shape. This class is an *abstract class* that implements **Comparable** interface with the following properties and methods:

- **Instance Fields (Attributes):** static variable PI with value 3.14, color, and description.
- **Constructor Method:** Shape (String color, String description)
- **Methods:**
  - *setColor*(String color): to set new color to this shape.
  - *getColor*(): return color of this Shape.
  - *toString*(): return information of this shape in the following pattern:

    "`description (color=____, area=____)`" (see expected output for more examples)

  - *compareTo*(Object shape): This method is used for comparing the size between this shape and the given shape obtained from the parameter. If this shape has a larger area than the given shape, return 1. If this shape has the same size as the given shape, return 0. Otherwise, return -1.
  - *getArea():* is an abstract method. You just have to define the method but do not have to implement the body of this method. This getArea() method will be implemented within each subclass (Circle, Square, and Triangle).

Task 2: Complete the “ShapeTester” class

Complete two static methods. Please see the comment for more details

- static void printAllShapes(HashMap<String, Shape> shapes)
- static int countInvalidName(HashMap<String, Shape> shapes)

In the main method, put two more objects into the shapeMap. One object must have a valid name; another one must have an invalid name

Expected output

```
my.circle is smaller than triangle x
Square* is the same size as triangle x
triangle2 is larger than triangle x
--------------------
triangle2->Triangle with base 20.0, height 40.0 (color=red, area=400.0)
circle_2->Circle with diameter 10.0 (color=yellow, area=314.0)
Square*->Square with width 10.0 (color=green, area=100.0)
2square->Square with width 5.0 (color=green, area=25.0)
my.circle->Circle with diameter 2.0 (color=yellow, area=12.56)
--------------------
The number of invalid shape’s name is 3
```

Challenge Bonus (Optional): Working with JFrame, Graphics, JPanel, etc.

Can you really draw Circle, Square, or Triangle shape on the screen? Here is the sample UI. However, feel free to make it more beautiful! You can use any libraries that you want to complete this task.