The Hong Kong Polytechnic University
Department of Electronic and Information Engineering

EIE3320 Lab 1: Object-Oriented Programming using Java

(Deadline for Submission: Check the course information)

Important Note: This is a group project. Two persons form a group. If you insist to do it on your own, one person per group is also fine.

Expected Outcomes

- Understand the principles of Object-Oriented design.
- Apply Java in Object-Oriented software development.
- Apply UML in Object-Oriented software modelling.
- Apply Object-Oriented approach to developing computer software.
- Learn independently and be able to search for the information required in solving problems.
- Present ideas and finding effectively.
- Work in a team and collaborate effectively with others.

Assessment Criteria

Your report should contain (but not limited to) the following:

1. Source code of your solutions (include the parts that you have modified)
2. The format of report should include the followings:
   - **2.1 Introduction**: A detailed description of the objectives and requirements of the program, and a brief descriptions of the methodology.
   - **2.2 Methodology**: The methodology when implementing the program. It contains
     - How your team divides the work among the team members?
     - The schedule of implementing the program
     - The program structure of the program developed, including
       - The specifications of the classes defined, and the public/private member functions/variables included
       - The flow of execution such as class diagram or flow chart.
   - **2.3 Program Testing**
     - The validations of your program and confirmed that it is running correctly.
       - Include the execution results of your program captured from the screen.
   - **2.4 Conclusion**
     - Summarize the experience gained in the program
   - **2.5 Future Development**
     - Indicate how your program can be extended.

General Descriptions

- After finished the program, each team should upload their program(s) and report to Blackboard.
- Each team member (student) must submit his/her own copy to Blackboard. It is expected that, if two students form a team, the team will submit two set of programs and reports to Blackboard. If a student does not submit his/her copy, they will score **no marks**.
- Each team member should declare his/her responsibility in the report. Each member will be individually assessed based on the declared responsibility and the result obtained.
- The report should be in PDF format. It is NOT required to include the complete source code in the report.
- It is compulsory to use a word processing tool to write your report. The font size must not be bigger than 12 or smaller than 10. Use 1.5 lines spacing on both sides of a page. (including all diagrams/tables, the length of the report should not be shorter than 15 pages.)

A. Brief Overview of Abstract Classes and Interface

Study the following URL

http://java.sun.com/docs/books/tutorial/java/IandI/abstract.html
http://java.sun.com/docs/books/tutorial/java/IandI/createinterface.html

B. Problem Statement

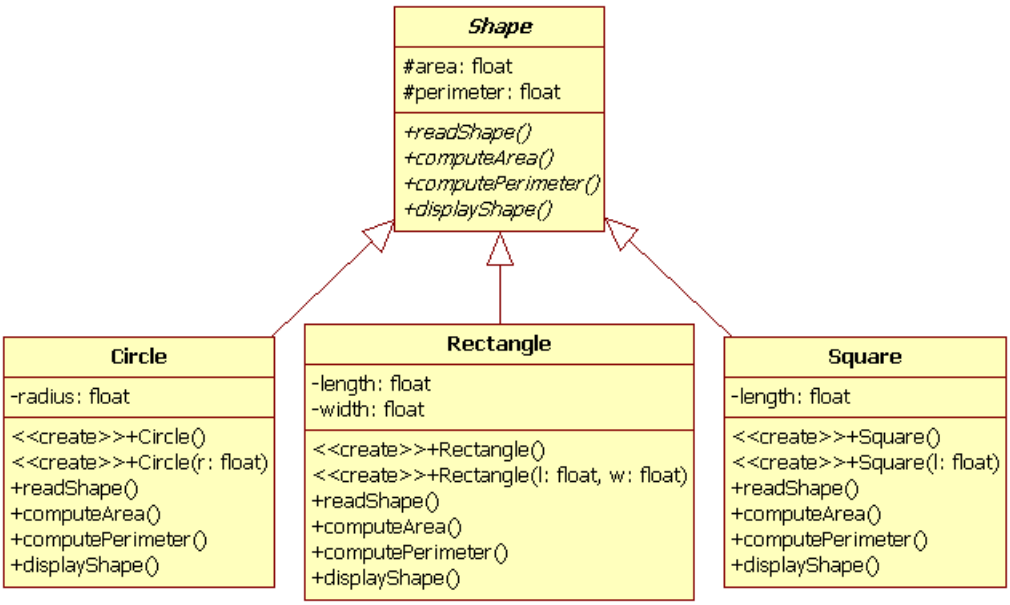
Write a program that can compute and display the area and perimeter of circles, squares, and rectangles. To achieve this task, you may create an abstract class called **`Shape`** that contains the following abstract methods.

```
// To read the shape information from users
abstract public void readShape();

// To compute the shape's area
abstract public void computeArea();

// To computer the shape's perimeter
abstract public void computePerimeter();

// To display the area and perimeter of the shape
abstract public void displayShape();
```

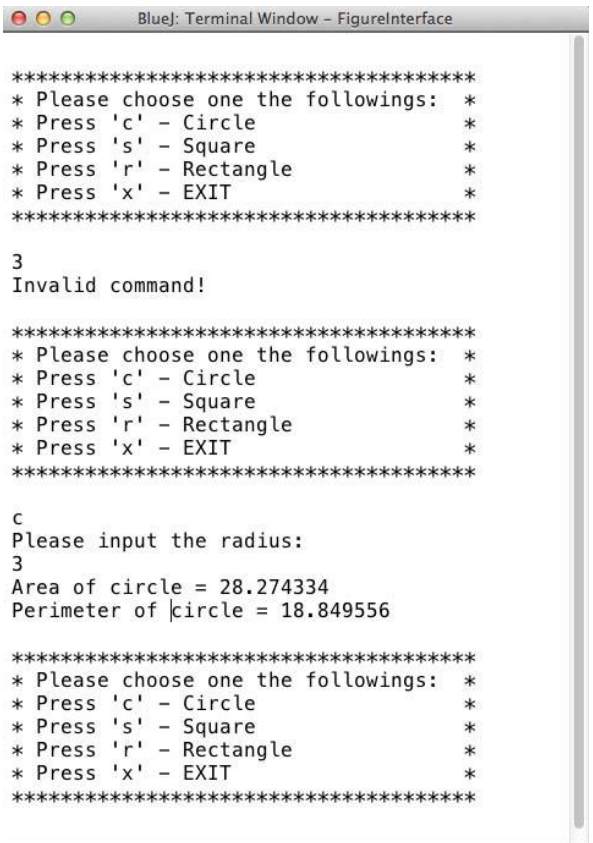
The **`Shape`** class also contains the protected member **`area`** and **`perimeter.`** You should use inheritance and polymorphism to make the code reusable and to reduce code complexity. Refer to the lecture notes on "Polymorphism" for the details of protected members.

C. Procedure

1. Use BlueJ or Eclipse to create the classes: **`Shape`**, **`Rectangle`**, **`Circle`**, and **`Square`**. These classes should have the relationship shown as follows. ***Hint*:** Some attributes may be unique to some classes only.



2. Create a Java file called "ShapeTester.java" to test the implementation. Your program output should look something like the following.



3. Create a Java file called "Picture.java". The class `Picture` should contain an `ArrayList` called `shapes` that stores a collection of `Circle`, `Square`, and `Rectangle` objects.

a) Write a public method with signature and return type as follows:
```
void addShape(Shape s);
```
The method shall add a `Shape` object `s` to the ArrayList `shapes`;

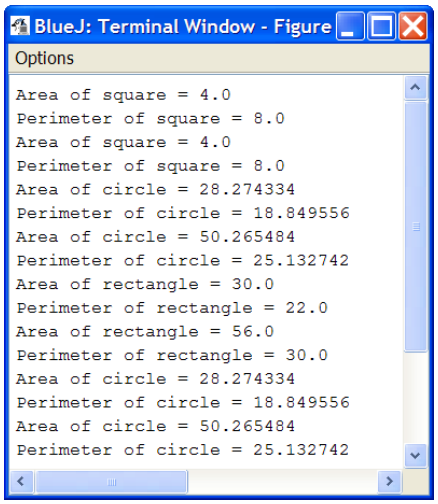
b) Write a public method with signature and return type as follows:
```
void computeShape();
```
The method shall compute the areas and perimeters of all objects in the ArrayList `shapes`;

c) Write a public method with signature and return type as follows:
```
public void listAllShapeTypes();
```
The method shall invoke `displayShape()` to display the areas and perimeters of all objects in the ArrayList `shapes`.

d) Write a public method with signature and return type as follows:
```
public void listSingleShapeType(String className);
```
The method shall display the areas and perimeters of all objects belonging to `className`. Note that `className` can be either `Circle`, `Square`, or `Rectangle`.

e) Create a class called `PictureTester` to test your codes in (a)–(d). Your "PictureTester.java" and console window should look like the following:

```
// PictureTester.java
public class PictureTester
{
  public static void main(String[] args) {
    Picture p = new Picture();
    p.addShape(new Square(2));
    p.addShape(new Square(2));
    p.addShape(new Circle(3));
    p.addShape(new Circle(4));
    p.addShape(new Rectangle(5,6));
    p.addShape(new Rectangle(7,8));
    p.computeShape();
    p.listAllShapeTypes();
    p.listSingleShapeType("Circle");
  }
}
```



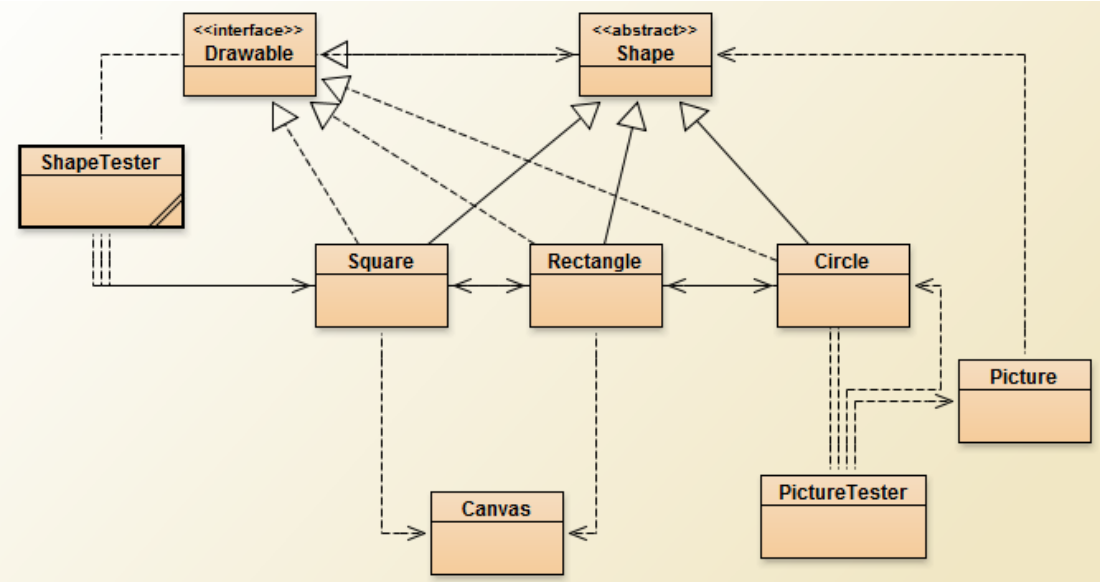
Hints:
1. Use the class `Class` detailed in http://docs.oracle.com/javaee/6/api/ .
2. Search the Internet using keywords: Java classname
3. Search the Internet using keywords: Java string comparison

4. Extend the program to draw the figure on the screen. You may use the "implements" keyword to produce the multiple-inheritance relationship as shown below. Note also that you may need to use the `Canvas` class in http://www.eie.polyu.edu.hk/~enhylin/BlueJProjects.zip (under Projects/Chapter1/Shapes). ***Note*:** To avoid name crashes between the AWT classes and the classes that you defined in this lab exercises, you may need to prefix the classes `Rectangle` and `Shape` with `java.awt.` in "Canvas.java", i.e., replace `Rectangle` with `java.awt.Rectangle`.

The class Drawable has the code looks like the following:

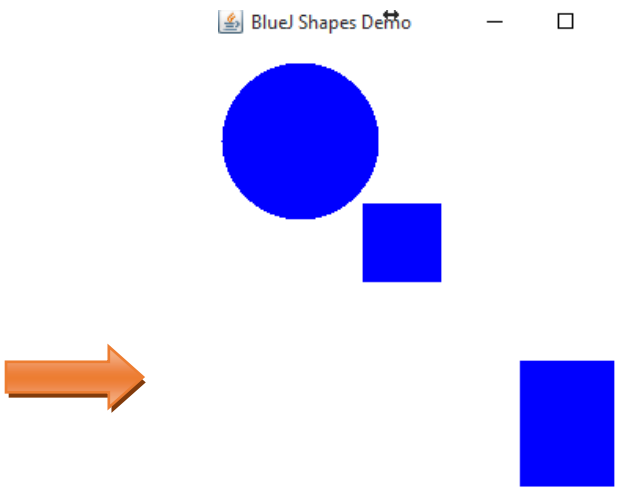
```
public interface Drawable
{
   void draw();
}
```

Note that you also need to change the code in "Shape.java" so that it implements the `Drawable` interface. The following shows the class diagram of program,



The result should draw the figure on the screen when input value in “ShapeTester.java”, the program output looks like the following.

```
c
Please input the radius:
50
Area of circle = 7853.982
Perimeter of circle = 314.15927

**************************************
* Please choose one the followings:  *
* Press 'c' - Circle                 *
* Press 's' - Square                 *
* Press 'r' - Rectangle              *
* Press 'x' - EXIT                   *
**************************************

s
Please input the length:
50
Area of square = 2500.0
Perimeter of square = 200.0

**************************************
* Please choose one the followings:  *
* Press 'c' - Circle                 *
* Press 's' - Square                 *
* Press 'r' - Rectangle              *
* Press 'x' - EXIT                   *
**************************************

r
Please input the length:
50
Please input the width:
60
Area of rectangle = 3000.0
Perimeter of rectangle = 220.0

**************************************
* Please choose one the followings:  *
* Press 'c' - Circle                 *
* Press 's' - Square                 *
* Press 'r' - Rectangle              *
* Press 'x' - EXIT                   *
**************************************
```



Lawrence Cheung
August 2017