

CSC1016S Assignment 8

GUI Programming I

Assignment Instructions

This assignment concerns (i) inheritance in Java, and (ii) the basics of graphical user interface (GUI) programming.

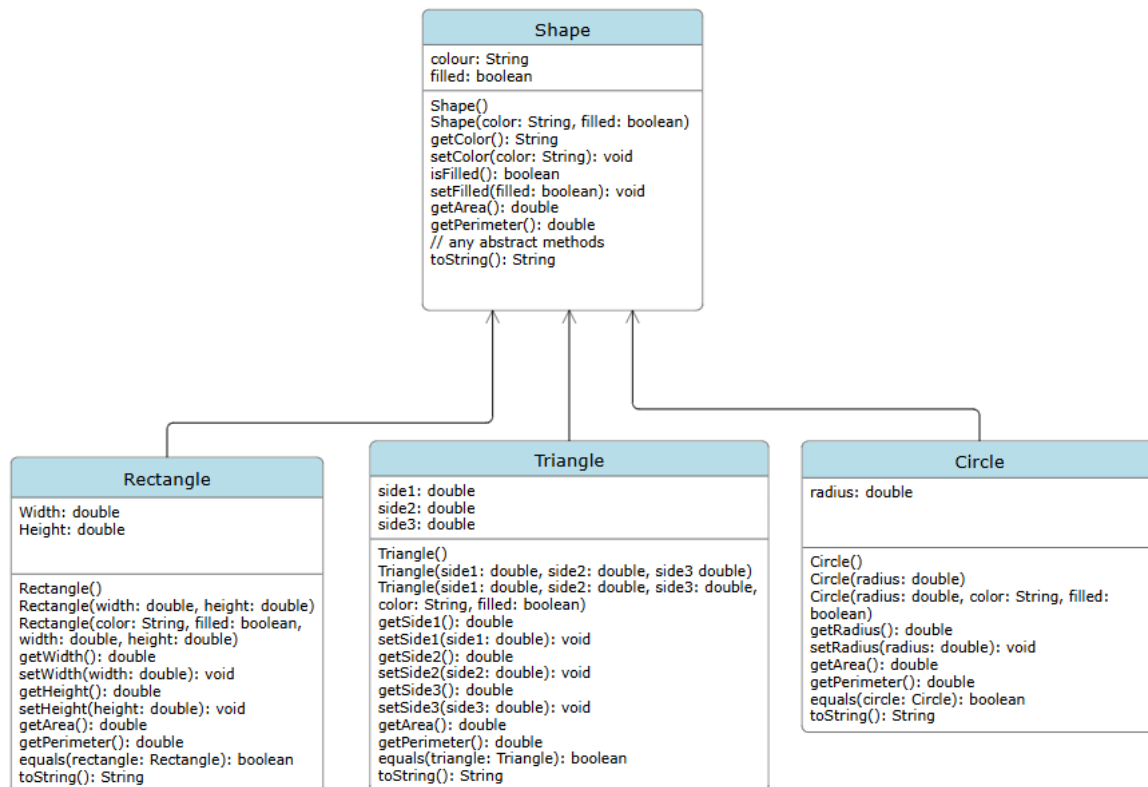
Question 1 [40 marks]

Consider geometric **Shapes**. Suppose you want to design the classes to model geometric shapes such as circles, triangles and rectangles. Geometric shapes have many common properties and behaviours. They can be drawn in a certain colour, filled or unfilled. Thus, a general class **Shape** can be used to model all geometric objects. This class contains the properties **colour** and **filled**, and their appropriate **get** and **set** methods. This class also contains a **toString()** method, that returns a string representation for the object.

- A circle, a triangle and a rectangle are special types of geometric shapes.
- The **Circle** class has a new data field, **radius**, and its associated **get** and **set** methods. It also contains the **getArea()**, **getPerimeter()**, and **getDiameter()** methods for returning the area, perimeter, and diameter of the circle.
- The **Rectangle** class has the data fields **width** and **height** and the associated **get** and **set** methods. It also contains the **getArea()** and **getPerimeter()** methods for returning the area and perimeter of the rectangle.
- The **Triangle** class contains three double data fields named **side1**, **side2**, and **side3**, with default values 1.0 to denote three sides of the triangle, and its associated **get** and **set** methods. It also contains the **getArea()** and **getPerimeter()** that returns the area and perimeter of this triangle.
- The class diagram depicting the relationship between the classes is given on page 2.

*Each class should contain appropriate **toString()** and **equals()** methods.*

Your task is to implement the three subclasses: **Circle**, **Rectangle**, and **Triangle** of the super class **Shape**.



Question 2 [60 marks]

This question concerns the construction of the GUI program for the geometric shapes in Question 1.

Design and implement a class named `ShapeMaker` that allows the user to draw circles, rectangles and triangles. The interface should allow the user to specify all the necessary information for the selected shape (**and create instances of these using the classes created above**), as well as specify the location of the shape and the size of the shape. The interface will then draw the shape, as well as display any other information regarding the shape (such as area, perimeter).

- `ShapeMaker` is a subclass of the `JFrame` class.
- Your GUI should allow the user to select the fill colour of the shape, the dimensions of the shape (or the radius in the case of a circle). For simplicity, fill colours can be limited to black, red, blue and green. **Use the appropriate getter and setter methods from question one to achieve this.**
- The user should also be able to specify the shape they want to draw. Then the shape should be displayed with the appropriate fill colour chosen.
- Use any other HCI concepts to come up with an intuitive user interface.
- You may choose to display the area and perimeter of a shape either on the top or bottom of the shape.
- **Hint:** you may use a **dropdown list**, **menus** and a **button to trigger the draw command**. Feel free to use any other options that serve to make navigation of your interface easy. See **Appendix** on the last page.

Your work will be manually assessed by tutors so make sure that you comment your source code where necessary.

Marking and Submission

Submit `Circle.java`, `Triangle.java`, `Shape.java`, `Rectangle.java`, and `ShapeMaker.java` in a single .ZIP folder to the automatic marker.

The zipped folder should have the following naming convention:

yourstudentnumber.zip

Appendix

JRadioButton

JRadioButton is a Swing component that represents an item with a state selected or unselected. Usually, a group of radio buttons is created to provide options to the user, but only one option can be selected at a time.

Here's the code to create a simple program that uses JRadio buttons:

```
import java.awt.FlowLayout;
import javax.swing.ButtonGroup;
import javax.swing.JFrame;
import javax.swing.JRadioButton;
import javax.swing.SwingUtilities;

public class SwingJRadioButtonDemo extends JFrame {

    public SwingJRadioButtonDemo() {
        super("Swing JRadioButton Demo");

        JRadioButton option1 = new JRadioButton("Linux");
        JRadioButton option2 = new JRadioButton("Windows");
        JRadioButton option3 = new JRadioButton("Macintosh");

        ButtonGroup group = new ButtonGroup();
        group.add(option1);
        group.add(option2);
        group.add(option3);

        setLayout(new FlowLayout());

        add(option1);
        add(option2);
        add(option3);

        pack();
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {

            @Override
            public void run() {
                new SwingJRadioButtonDemo().setVisible(true);
            }

        });
    }
}
```

JComboBox

JComboBox is a Swing component that renders a drop-down list of choices and lets the user select one item from the list.

Details on how to create a JComboBox with sample Java code can be found at <https://www.codejava.net/java-se/swing/jcombobox-basic-tutorial-and-examples>.

Feel free to explore other Swing components listed on the menu on the left pane.