Week 10 Lab Exercises
Inheritance and Animation

In this lab, we will explore inheritance and simple animation in Java.

Lab setup:

- Create a project named Week-10-Lastname-Firstname
- Use the *default* package or create appropriately named packages for your classes

Problem 1. ***Shape Hierarchy*** **(50 points).**

Implement the **`Shape`** class hierarchy shown in the figure below. Each shape should implement the **`toString`** method that returns a text description of the object as a string. For example, the Shape class can return "I am a Shape" string. Both **`Circle`** and **`Square`** should extend the base Shape class and should override its **`toString`** method with a more appropriate message. They should also contain the method **`getArea`** to calculate the area of the shape, and implement appropriate constructors, getters, and setters.

Write a main method to test your classes by creating some circle and square objects and computing their area.

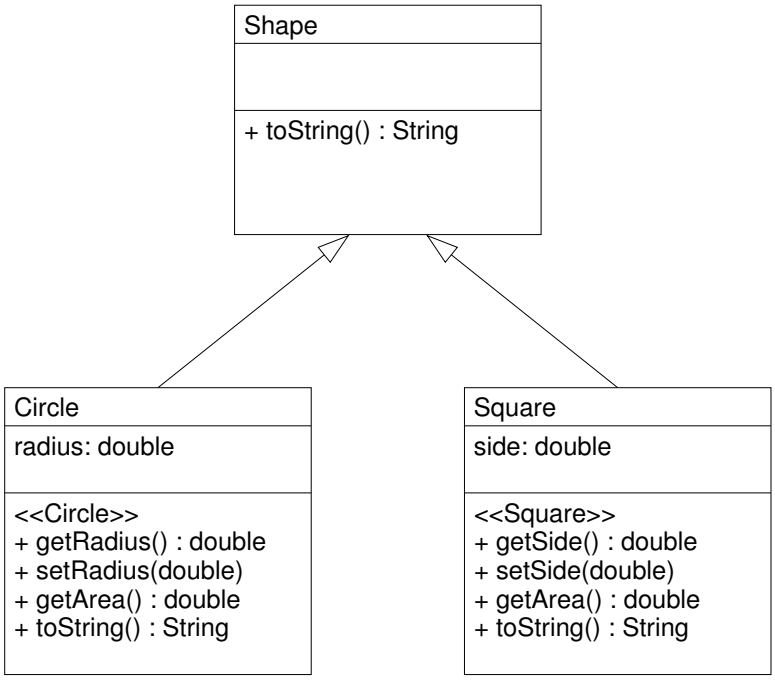
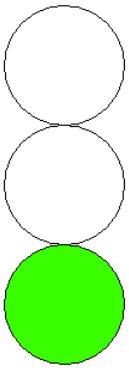


Figure 1. Shape class hierarchy diagram.

Problem 2. Traffic light simulation (50 points).

Write a program that does a simple traffic light simulation. The traffic light should cycle through red, yellow, and green colors as filled-in circles (see the figure to the right).



Use the code provided below which triggers a panel repaint every second. You can use variable **`count`** to keep information about the state of the traffic light. Feel free to experiment and structure your code as you wish.

```
import java.awt.Color;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JPanel;
import javax.swing.JFrame;
import javax.swing.Timer;

public class Semaphore extends JPanel implements ActionListener {

    private int count = 0;
    private Timer animationTimer;

    // constructor starts the timer to trigger every 1 second
    public Semaphore() {
        animationTimer = new Timer(1000, this);
        animationTimer.start();
    }

    // repaint semaphore with each timer trigger event
    @Override
    public void actionPerformed(ActionEvent e) {
        repaint();
    }

    public void paintComponent( Graphics g ) {
        super.paintComponent(g);

        // add your traffic light code here

    }

    public static void main(String args[]) {
        JFrame window = new JFrame();

        Semaphore s = new Semaphore();
        window.add(s);

        window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        window.setSize(200, 360);
        window.setVisible(true);
    }

}
```