```java
package cisc181.mytestlab_0;

// Christopher Rasmussen, CISC181 spring 2019

import javax.swing.JComponent;
import javax.swing.Timer;

import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;

public class MouseAndTimerJComponent extends JComponent {

  private String printString;
  private int counter;
  private final int COUNTER_MAX = 1000;
  private Timer timer;
  private MyButton stopButton, startButton;

  class MyButton {
    int x, y, width, height;
    boolean pressedState;
    String text;
    MyButton(int x, int y, int width, int height, String text) {
      this.x = x;
      this.y = y;
      this.width = width;
      this.height = height;
      this.text = text;
      pressedState = false;
    }

    // is a mouse click at (mouseX, mouseY) inside this button?

    boolean checkPressed(int mouseX, int mouseY) {
      return (mouseX >= x && mouseX < (x + width) && mouseY >= y && mouseY < (y + height));
    }

    // show this button and associated text

    void draw(Graphics g) {

      g.setColor(Color.BLACK);
      g.drawRect(x, y, width, height);  // note that y is UPPER edge, not bottom

      Font currentFont = g.getFont();
      Font newFont = currentFont.deriveFont(currentFont.getSize() * 2F);
      g.setFont(newFont);
      int textWidth = g.getFontMetrics().stringWidth(text);
      int textHeight = g.getFontMetrics().getHeight();

      g.drawString(text, x + width/2 - textWidth/2, y + height/2 + textHeight/2);  // y here is BOTTOM of string
      g.setFont(currentFont);

    }
  }


  class MouseClickListener implements MouseListener {

    public void mouseClicked(MouseEvent e) { }
```

```
        public void mouseEntered(MouseEvent e) { }
        public void mouseExited(MouseEvent e) { }

        public void mousePressed(MouseEvent e) {
            if (startButton.checkPressed(e.getX(), e.getY())) {
                printString = "Start!";
                timer.start();
            }
            else if (stopButton.checkPressed(e.getX(), e.getY())) {
                printString = "Stop!";
                timer.stop();
            }
            else {
                printString = "Pressed!";
            }
            repaint();
        }

        public void mouseReleased(MouseEvent e) {
            printString = "Swing?";
            repaint();
        }
    }

    MouseAndTimerJComponent(int width, int height) {

        startButton = new MyButton(50, height - 100, 100, 50, "Start");
        stopButton = new MyButton(160, height - 100, 100, 50, "Stop");

        printString = "Swing?";

        addMouseListener(new MouseClickListener());

        counter = 0;
        timer = new Timer(1000, new ActionListener() {

            public void actionPerformed(ActionEvent ae) {
                counter++;
                if (counter >= COUNTER_MAX)
                    counter = 0;

                repaint();
            }
        });
    }

    public void paintComponent(Graphics g) {

        g.setColor(Color.RED);
        Font currentFont = g.getFont();
        Font newFont = currentFont.deriveFont(currentFont.getSize() * 5F);
        g.setFont(newFont);
        g.drawString(printString + " " + counter, 50, 100);
        g.setFont(currentFont);

        startButton.draw(g);
        stopButton.draw(g);
    }
}
```