Assignment 4 – Art work with Java

Maximum Points = 50

The purpose of this lab is to continue your study of computer programming and algorithms with an introduction to object-oriented programming using the Java programming language. In this lab you will use several new features including – using and constructing objects of predefined classes to create a piece of “art.”.

Enter and run the following program containing the two classes (1) FrameViewer and (2) MyComponent:

```
import java.awt.Color;
import javax.swing.JFrame;
import javax.swing.JLabel;

/**
 * FrameViewer allows a frame of graphics to be
viewed
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class FrameViewer
{
    /**
     * main method that controls the program
     */
    public static void main(String[] args)
    {
      JFrame frame = new JFrame();
      frame.setSize(300, 400);
      frame.setTitle("My art work");
      MyComponent canvas = new MyComponent();
      frame.add(canvas);
      frame.setDefaultCloseOperation
(JFrame.EXIT_ON_CLOSE);
      frame.setVisible(true);
    }
}
```

```
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Rectangle;
import javax.swing.JComponent;

/**
 * object of class myComponent provide a "canvas" to
record your art work.
 *
 * @author (your name)
 * @version (a version number or a date)
 */

public class MyComponent extends JComponent
{
   /**
     * method used in awt to "paint" on the component
     *
     * @param  g   graphics used to "paint" with
     * @return     none
     */
    public void paintComponent(Graphics g)
    {
        // declares the 2D Graphics object and a Rectangle
        Graphics2D g2;
        Rectangle box;

        // recast the Graphics object as a 2D Graphics
        g2 = (Graphics2D) g;

        // Construct a rectangle and draw it
        box = new Rectangle (5, 10, 20, 30);
        g2.draw(box);

        // Move rectangle 15 units to right, 25 units down
        // change the color of graphics to blue
        box.translate(15, 25);
        g2.setColor(Color.BLUE);

        // draw the rectangle in a second location
        g2.draw(box);
    }
}
```

Your program must have these additional features:

1. Modify the frame size and frame title.

2. Draw additional graphics (e.g. Ellipse2D, line2D) using different shapes, sizes, and colors using at least one “custom color [see P2.7 on page 75]”. Be creative (see FaceComponent.java)
3. Use drawString to “write” your name and a message on the canvas.

CHALLENGES: a) Use ImageIcon class defined in the Java API to add one or more images to your canvas.

b) Use the Font and Color classes to create fonts and colors that can be used by the graphic to change the fonts and colors of the text.

c) Use JOptionPane to ask the user for a message and display the message on the canvas.

HAVE FUN!!!!

Make sure that your program uses proper indentation and complete documentation. See http://csc.columbusstate.edu/summers/NOTES/1301/style.htm for guidelines.

The program heading should occur at the top of the program and should include:

```
/**
 * PROGRAM SPECIFICATIONS
 * NARRATIVE DESCRIPTION:
 *
 * @author (your name)
 * @version(date)
 ==============================*/
```

(Due before 8 am on Wednesday, October 3, 2012) Submit your .java files containing your program and your timesheet documenting your time to the dropbox in WebCT.

Grades are determined using the following scale:

- Runs correctly..………………….:___/10
- Correct output……..…………….:___/10
- Design of output..……………….:___/8
- Design of logic………………….:___/10
- Standards……………………….:___/7
- Documentation………………...:___/5

Grading Rubric (Word document)