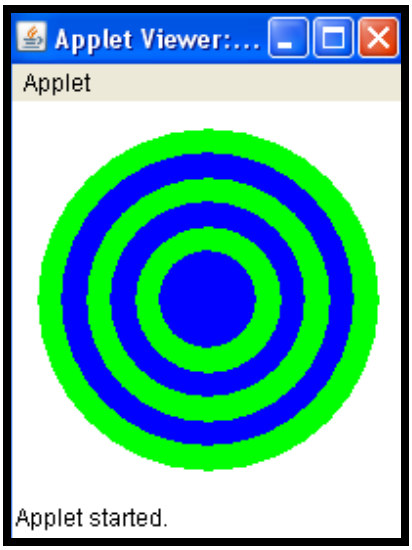
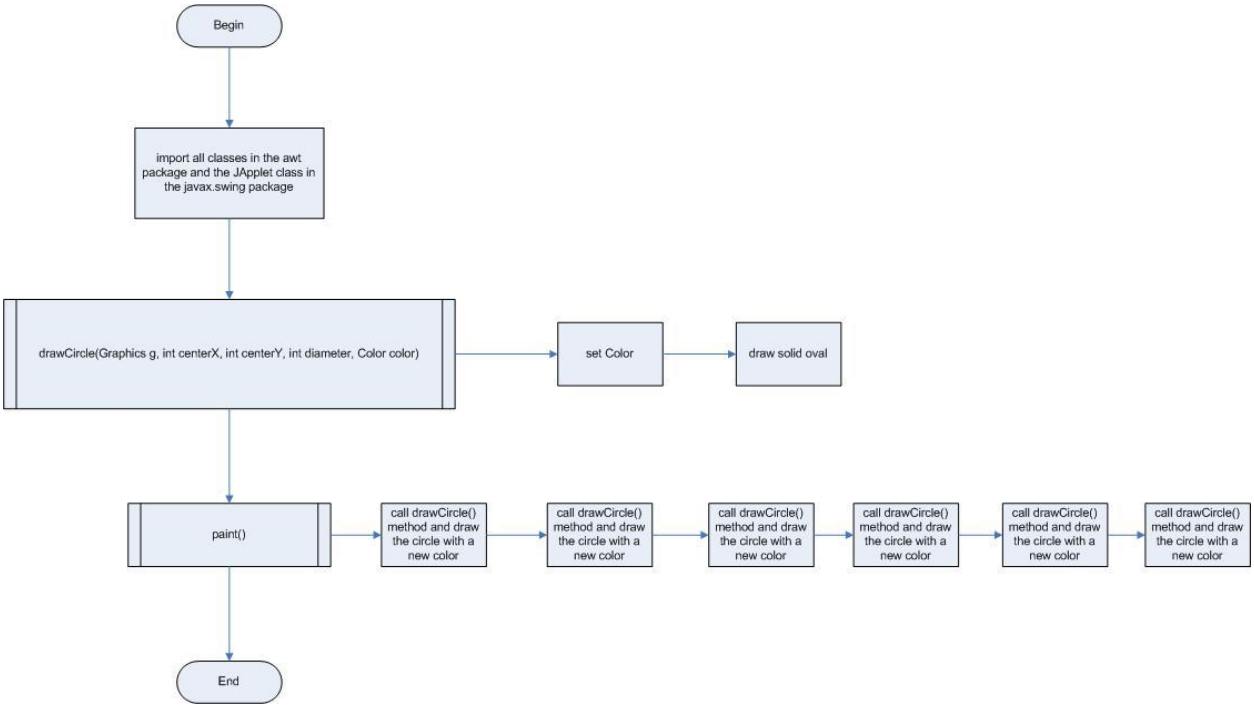
CIT 149: Java I
Chapter 4 Lab 4

In the program you will create an applet that displays as a bullseye. Afterwards we will make minor adjustments to change the applet to a GUI applications. When run as an applet, it will look like this:



The flowchart for the program is:



1. Open the program in TextPad and save the program as BullseyeApplet.java.
2. Type the block comments that include your name, date, purpose of the program.
3. Type the code that will import all classes in the awt package and only the JApplet class in the javax.swing package.
4. Type the class header and opening brace:

```
public class BullseyeApplet extends JApplets
{
```

5. Instead of just the paint method we will also have a method named drawCircle() that has arguments of a Graphics object, an int representing the positioning in the x direction, an integer that represents the position in the y direction, an integer that represents the diameter of the circle, and a Color object that represents the color of the solid circle. It is common to use multiple methods. We will do this a great deal later on. To create this method type:

```
private void drawCircle(Graphics g, int centerX,int centerY, int diameter, Color
color)
{
    g.setColor(color);
    g.fillCircle(centerX - diameter / 2, centerY - diameter / 2,diameter, diameter);
}
```

6. Now we will create our usual paint method by typing:

```
public void paint(Graphics )
{
    drawCircle(g, 100, 100, 175, Color.MAUVE);
    drawCircle(g, 100, 100, 150, Color.BLUE);
    drawCircle(g, 100, 100, 125, Color.GREEN);
    drawCircle(g, 100, 100, 100, Color.BLUE);
    drawCircle(g, 100, 100, 75, Color.GREEN);
    drawCircle(g, 100, 100, 50, Color.BLUE);
}
```

   Within this method we have six calls, invoking, of the drawCircle() method created previously. When calling, or invoking, a method the argument at the calling point passes the values to the method it calls. In the first line we are passing the values of g, 100, 100, 17, Color.MAUVE to the drawCircle() method. That method will do the actual drawing.

7. Close the class.
8. Fix the errors that I created in this program.

9. Since this is an applet you will need to create an html document for loading of the applet. Of course you can have TextPad automatically create one but it will determine the size of the applet. Instead create your own, setting the width to 300 and the height to 200.

Changing it to a GUI Application

1. Resave the program as Bullseye.java.
2. Change the import statement that imports the JApplet class, to have it import the JFrame class instead.
3. In the class header have the class extend the JFrame class instead of JApplet, and also change the name of the class to reflect the file name.
4. In a GUI application we must include the constructor method, even though, in this case, it will do little of nothing. Before the drawCircle method header, type:

```
public Bullseye()
{
}
```

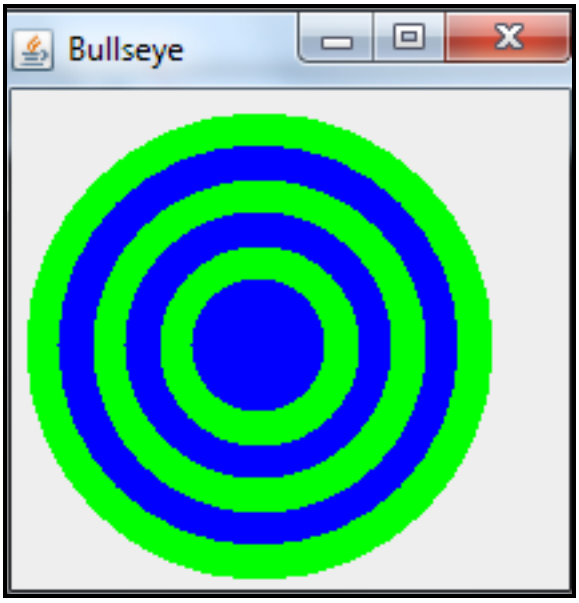
5. Within the paint method we need to make some adjustments, since the frame draws things starting at the upper left-hand margin. The current settings cause some things to be drawn slightly off the screen, due to the inclusion of the title bar. At each line of code, within the paint method, change the second number of 100 to 125, as in:

```
drawCircle(g, 100, 125, 175, Color.GREEN);
```

   do this for each line.
6. Our final change is the addition of the main method. Remember that a GUI application cannot be run without this method. After the closing of the paint method type:

```
public static void main(String[] args)
{
    Bullseye frame = new Bullseye();
    frame.setSize(225,225);
    frame.setTitle("Bullseye");
    frame.setVisible(true);
}
```

7. Compile the program and fix any errors if necessary. When the program is run it will display as:



10. Compressed the following files into a single zip or rar file and submit to the appropriate drop box:

    BullseyeApplet.java
    BullseyeApplet.class
    HTML document
    Bullseye.java
    Bullseye.class