

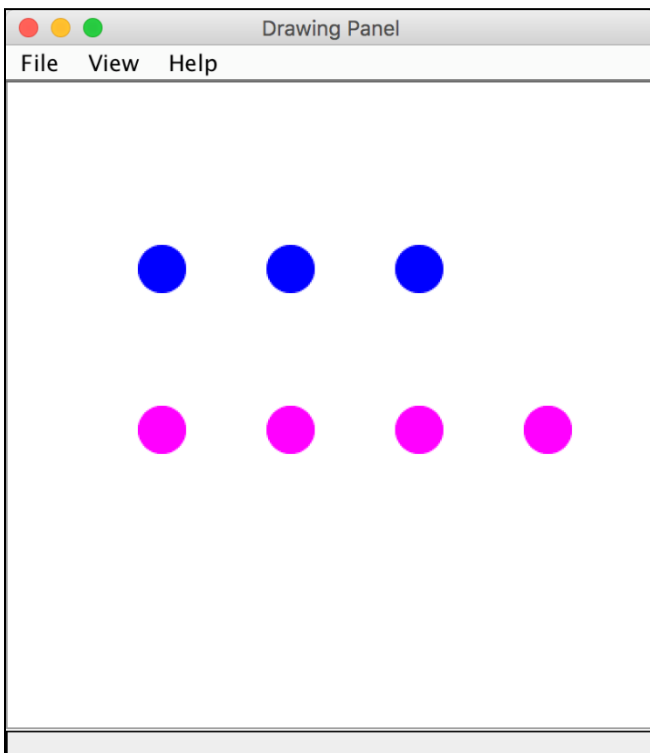
## CSE 142 Section Handout #3 Style Sheet

Consider the following program:

```
import java.awt.*;
public class StyleProb {

    public static int circleDiameter = 30;
    public static void main(String[] args) {
        DrawingPanel panel = new DrawingPanel(400, 400);
        Graphics g = panel.getGraphics();

        g.setColor(Color.BLUE);
        threeCircles(g, circleDiameter, circleDiameter / 2, 100);
        g.setColor(Color.MAGENTA);
        fiveCircles(g, circleDiameter, circleDiameter / 2, 200);
    }
    // draws three circles
    public static void threeCircles(Graphics g, int diam, int radius, int yCoord) {
        for (int i = 1; i <= 3; i++) {
            g.fillOval(80 * i, yCoord, diam, radius * 2);
        }
    }
    // draws five circles
    public static void fiveCircles(Graphics g, int diam, int radius, int yCoord) {
        for (int i = 1; i <= 5; i++) {
            g.fillOval(80 * i, yCoord, diam, radius * 2);
        }
    }
}
```



When ran, this program produces the following image:

While this program would receive full external correctness by producing the desired output, it would not receive full internal correctness. List all style issues you can find.

## CSE 142 Section Handout #3

### Style Sheet Solutions

Corrected:

```
// Donald Knuth
// TA: Alan Turing
//
// This program outputs a piece of abstract art.
// The art consists of 2 rows of colored, filled-in circles
// (blue and magenta) with a white background

import java.awt.*;

public class StyleProb {
    public static final int CIRCLE_DIAMETER = 30;

    public static void main(String[] args) {
        DrawingPanel panel = new DrawingPanel(400, 400);
        Graphics g = panel.getGraphics();

        g.setColor(Color.BLUE);
        drawCircles(g, 100, 3);
        g.setColor(Color.MAGENTA);
        drawCircles(g, 200, 5);
    }

    // Draws the given number of circles in a row at a height of yCoord, using the
    // Graphics object
    public static void drawCircles(Graphics g, int yCoord, int numCircles) {
        for (int i = 1; i <= numCircles; i++) {
            g.fillOval(80 * i, yCoord, CIRCLE_DIAMETER, CIRCLE_DIAMETER);
        }
    }
}
```

- **Class constant declaration** – Class constants should be declared 'final'. The 'final' keyword signifies that this value will not be changed during the duration of this. Class constants should also be named in all capital letters with underscores in between words (e.g., CIRCLE\_DIAMETER).
- **Redundant methods** – The methods `threeCircles` and `fiveCircles` are redundant. The only difference between the two methods is the number of times the loops iterate, and this difference can be expressed in a parameter.
- **Parameters** – The radius does not need to be passed as a parameter, as it can be calculated from the diameter class constant. Also notice that it is never used as its own value, and so is even more unnecessary (it is only used in `radius * 2` which is equivalent to `circleDiameter`). The class constant's scope is the entirety of the program, and so it should not be passed as a parameter to the `threeCircles` and `fiveCircles` methods.
- **Comments** – Method comments should describe what that method does, as well as any parameters the method accepts. Each method's comments should provide enough information for someone to know how to call the method without looking at its actual implementation. Also missing a class constant, name, and TA. Describe at a high level what the program does in the class comment.
- **Indentation** – The indentation of a program should increase by one tab every time a curly brace ( { ) is opened and decrease by one tab every time a curly brace ( } ) is closed. The class constant declaration should be indented by one more tab, and method comments should be aligned with the corresponding method header on the line below.
  - Incorrect indentation may result in point deductions on Homework 3.
- **Blank lines between methods** – The spacing between methods also matters. Remember to leave a blank line between each method for readability.
  - Failure to separate methods with blank lines may lose style points in Homework 3.