

## Section Handout 3

Based on a handout by Eric Roberts, Mehran Sahami, and Patrick Young

### Problem One: True or False?

For each of the following statements below, indicate whether it is true or false in Java:

1. The value of a *local variable* named `i` has no direct relationship with that of a variable named `i` in its caller.
2. The value of a *parameter* named `x` has no direct relationship with that of a variable named `x` in its caller.

### Problem Two: Method Trace

For the program below, trace through its execution by hand to show what output is produced when it runs.

```
/*
 * File: QuestionableJava.java
 * =====
 * A program that explores some of the trickier points of parameter passing.
 */
import acm.program.*;

public class QuestionableJava extends ConsoleProgram {
    public void run() {
        int marten = 137;
        int faye = 42;

        println("marten = " + marten);
        hannelore(faye);
        println("marten = " + marten);
        println("faye = " + faye);

        marten = angus(faye, marten + faye);
        println("marten = " + marten);

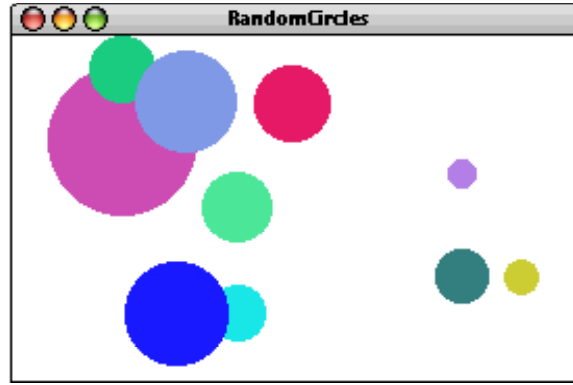
        marten = angus(marten, faye);
        println("marten = " + marten);
    }

    private void hannelore(int marten) {
        println("marten = " + marten);
        marten = 160;
    }

    private int angus(int martin, int faye) {
        int dora = faye - martin;
        println("dora = " + dora);
        return dora % 10;
    }
}
```

### Problem Three: Random Circles

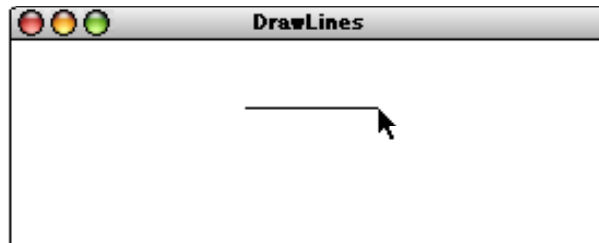
Write a `GraphicsProgram` that draws a set of ten circles with different sizes, positions, and colors. Each circle should have a randomly chosen color, a randomly chosen radius between 5 and 50 pixels, and a randomly chosen position on the canvas, subject to the condition that the entire circle must fit inside the canvas without extending past the edge. The following sample run shows one possible outcome:



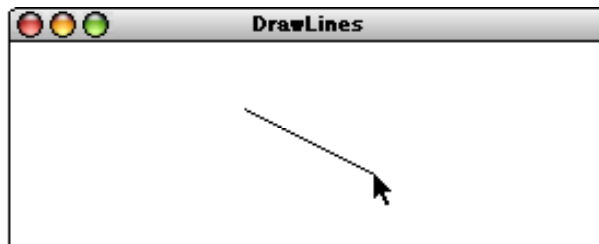
On some runs of this program you might not see ten circles. Why?

### Problem Four: Drawing Lines

Write a `GraphicsProgram` that allows the user to draw lines on the canvas. Pressing the mouse button sets the starting point for the line. Dragging the mouse moves the other endpoint around as the drag proceeds. Releasing the mouse fixes the line in its current position and gets ready to start a new line. For example, suppose that you press the mouse button somewhere on the screen and then drag it rightward an inch, holding the button down. What you'd like to see is the following picture:



If you then move the mouse downward without releasing the button, the displayed line will track the mouse, so that you might see the following picture:



Because the original point and the mouse position appear to be joined by some elastic string, this technique is called rubber-banding. Although this program may seem quite powerful, it is also simple to implement. The entire program requires fewer than 20 lines of code.