Steve Cooper
CS 106A

Handout #15
January 19, 2011

Section Handout #2—Simple Java

Portions of this handout by Eric Roberts

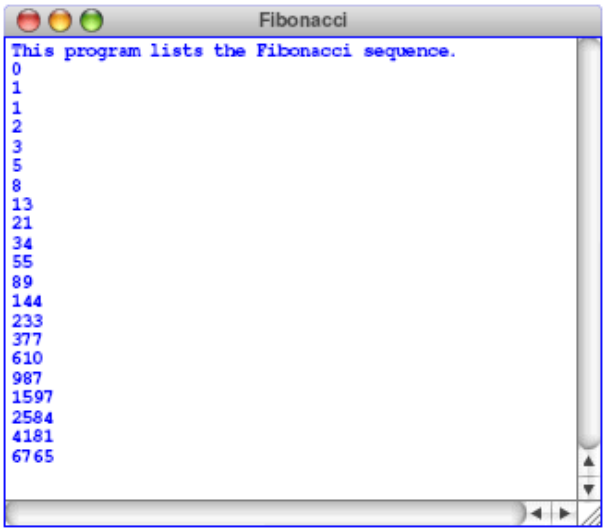
Note: Beginning with this section assignment, one of the questions will have asterisks by it. This means that you are to try to solve this question (preferably on paper) prior to section. Your section leader will collect your solution.

1. The Fibonacci sequence

In the 13th century, the Italian mathematician Leonardo Fibonacci—as a way to explain the geometic growth of a population of rabbits—devised a mathematical sequence that now bears his name. The first two terms in this sequence, **`Fib`**(0) and **`Fib`**(1), are 0 and 1, and every subsequent term is the sum of the preceding two. Thus, the first several terms in the Fibonacci sequence look like this:

`Fib`(0)	=	0	
`Fib`(1)	=	1	
`Fib`(2)	=	1	(0 + 1)
`Fib`(3)	=	2	(1 + 1)
`Fib`(4)	=	3	(1 + 2)
`Fib`(5)	=	5	(2 + 3)

Write a program that displays the terms in the Fibonacci sequence, starting with **`Fib`**(0) and continuing as long as the terms are less than 10,000. Thus, your program should produce the following sample run:



This program continues as long as the value of the term is less than the maximum value, so that the loop construct you need is a **`while`**, presumably with a header line that looks like this:

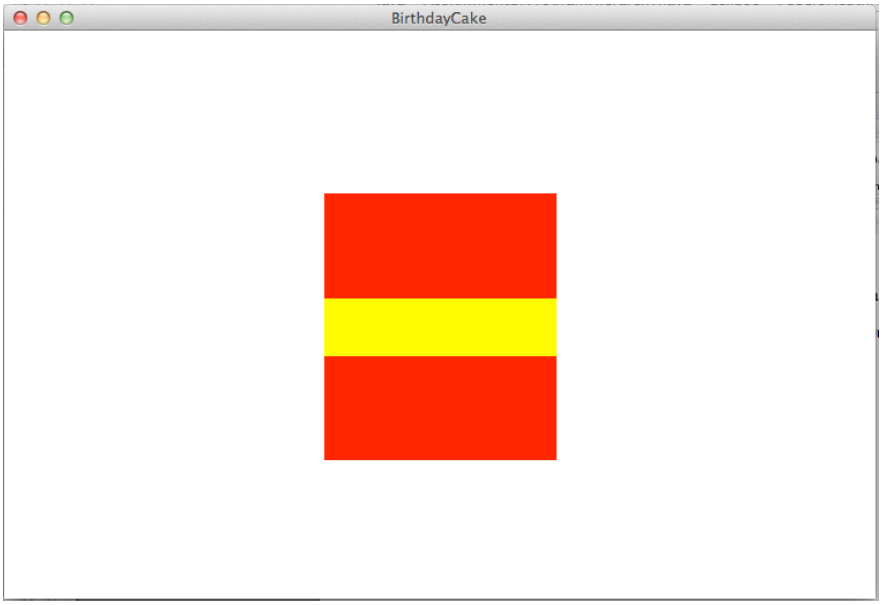
```
while (term < MAX_TERM_VALUE)
```

Note that the maximum term value is specified using a named constant.

2. ****Birthday Cake***

It's your friend's birthday, and you promised that you'd make her a cake. Unfortunately for your friend, you are a better programmer than baker, so you decide to program a digital greeting card instead of busting out the flour.

You decide to make a three-layer cake that looks something like this:



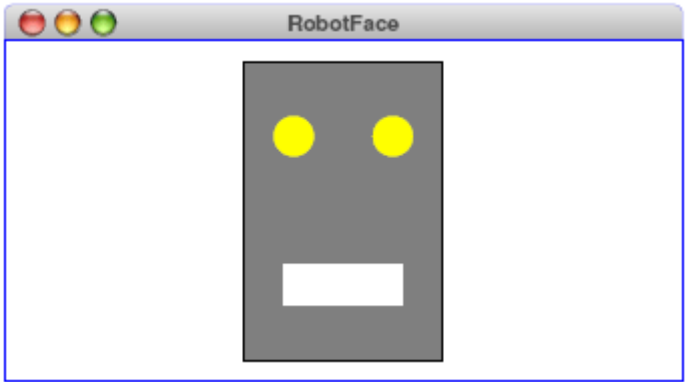
The entire cake must be centered both horizontally and vertically in the window. Recall that **`getHeight()`** and **`getWidth()`** return the height and width (respectively) of the graphics window.

Constants are provided for the measurements – **`INNER_LAYER_HEIGHT`**, **`OUTER_LAYER_HEIGHT`**, and **`CAKE_WIDTH`**.

Hint: Think about how you center one object in a window. How can you adapt that to center a group of three objects?

3. Drawing a face

Your job is to draw a robot-looking face like the one shown in the following sample run:



This simple face consists of four parts—a head, two eyes, and a mouth—which are arranged as follows:

- *The head*. The head is a big rectangle whose dimensions are given by the named constants **`HEAD_WIDTH`** and **`HEAD_HEIGHT`**. The interior of the head is gray, although it should be framed in black.
- *The eyes*. The eyes should be cricles whose radius in pixels is given by the named constant **`EYE_RADIUS`**. The centers of the eyes should be set horizontally a quarter of the width of the head in from either edge, and one quarter of the distance down from the top of the head. The eyes are yellow.
- *The mouth*. The mouth should be centered with respect to the head in the *x*-dimension and one quarter of the distance up from the bottom of the head in the *y*-dimension. The dimensions of the mouth are given by the named constants **`MOUTH_WIDTH`** and **`MOUTH_HEIGHT`**. The mouth is white.

Finally, the robot face should be centered in the graphics window.