

Section Handout #2

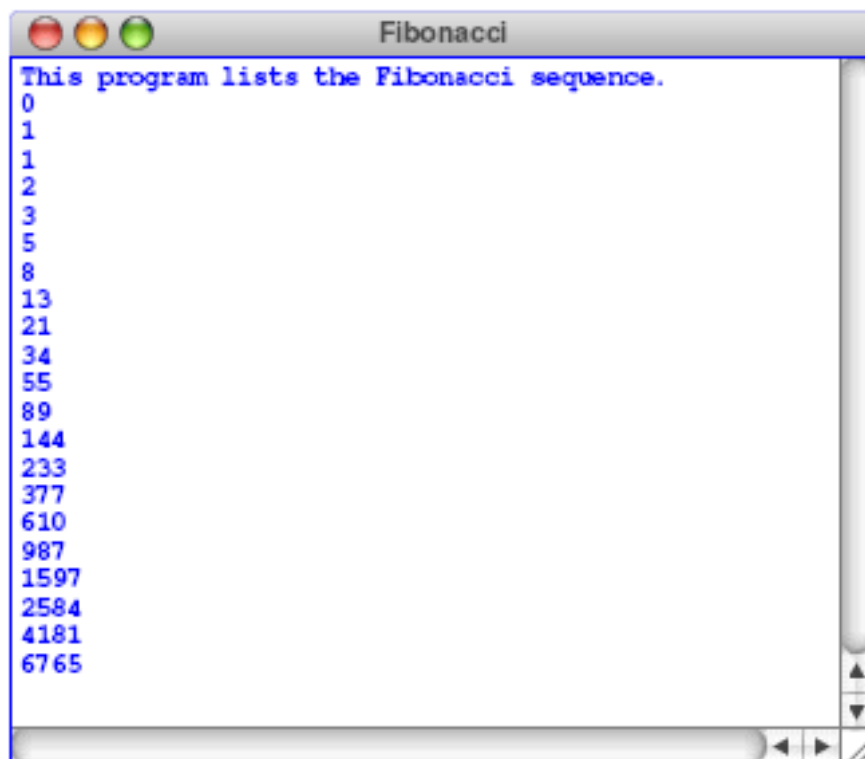
Based on a handout by Eric Roberts and Mehran Sahami

Problem One: The Fibonacci sequence

In the 13th century, the Italian mathematician Leonardo Fibonacci—as a way to explain the geometric 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:

<code>Fib(0)</code>	=	0	
<code>Fib(1)</code>	=	1	
<code>Fib(2)</code>	=	1	(0 + 1)
<code>Fib(3)</code>	=	2	(1 + 1)
<code>Fib(4)</code>	=	3	(1 + 2)
<code>Fib(5)</code>	=	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:



```
Fibonacci
This program lists the Fibonacci sequence.
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
6765
```

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)
```

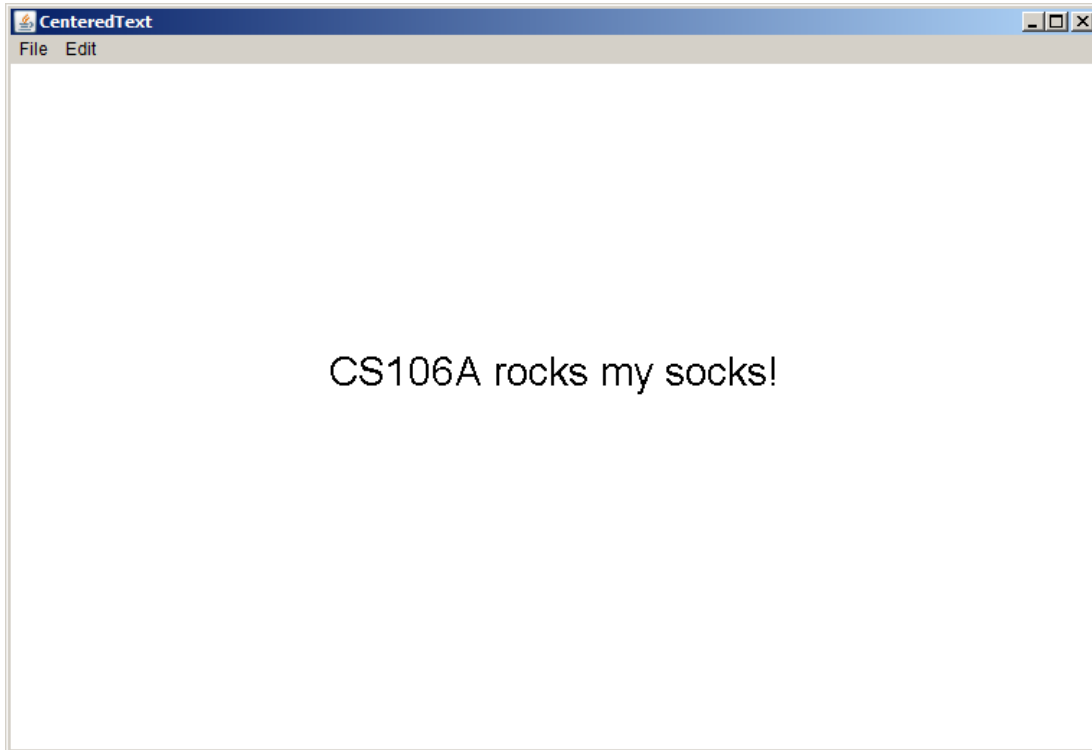
Note that the maximum term value is specified using a named constant. An aside: The Fibonacci sequence has numerous applications in computer science and shows up in surprising places. It's used to efficiently compute logarithms, index and retrieve data, and as a building block in some route-planning algorithms.

Problem Two: Drawing Centered Text

Your job is to write a `GraphicsProgram` that displays the text message (i.e., `GLabel`):

CS106A rocks my socks!

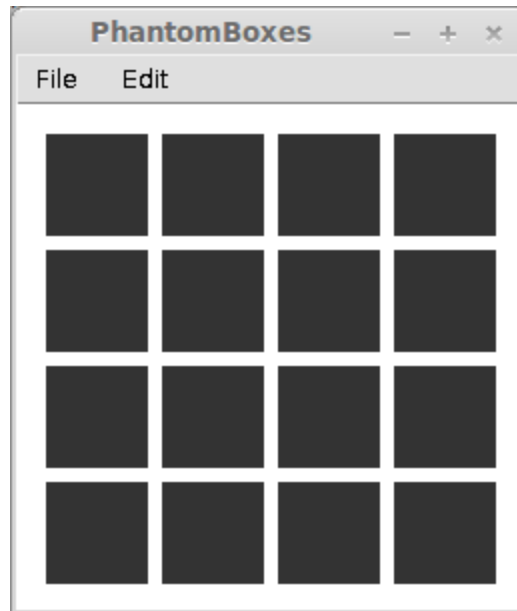
The text should be displayed in SansSerif 28-point font, and centered horizontally and vertically in the middle of the graphics window, looking something like this:



You can find the width of a label by calling `label.getWidth()` and the height it extends above the baseline by calling `label.getAscent()`. If you want to center a label, you need to shift its origin by half of these distances in each direction.

Problem Three: Optical Illusions

In this problem, you'll write a program that produces an optical illusion. By drawing a grid of black squares with small amounts of spacing in-between them, your brain will trick you into thinking that there are small grayish areas in the corners between those squares. The output of the program is shown here:



When writing this program, you should make sure to do the following:

- The figure should be centered both horizontally and vertically in the window.
- The number of boxes on each side should be controlled by the `BOXES_PER_SIDE` constant.
- The width and height of each box should be controlled by the `BOX_SIZE` constant.
- The spacing between the boxes should be controlled by the `BOX_SPACING` constant.

You may find it easier to do the math to position each box by computing the x and y coordinates of the box and the upper-left corner of the grid, then determining for the box in row i and column j how much to offset that box as a function of the given constants, along with i and j .