

Section Handout #2—Console & Graphics Programs

Based on handouts by Marty Stepp and Keith Schwarz

1. Fibonacci Sequence

Write a ConsoleProgram that displays all of the numbers in the Fibonacci sequence that are less than 10,000, starting with 0. The first two terms in the sequence are 0 and 1, and every subsequent term is the sum of the previous two terms. Your program should produce the following output:

```
This program lists the Fibonacci Sequence for numbers less than 10,000.  
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765
```

2. Fizz Bazz Buzz

Fizz bazz buzz is a game for a group of people sitting in a circle. The group counts upward from 0, saying the numbers in ascending order. However, there are some restrictions – any time a person would say a number that's a multiple of three, they instead say “fizz.” Any time a person would say a number that's multiple of five, they instead say “bazz.” Finally, any time a person would say a number that's a multiple of three *and* a multiple of five, they instead say “buzz.” (0 counts as a multiple of every number, by the way.)

For example, the first rounds of fizz bazz buzz would be:

```
Buzz  
1  
2  
Fizz  
4  
Bazz  
Fizz  
7  
8  
Fizz  
Bazz  
11
```

Your job is to write a program that prompts the user for a number, then plays that many rounds of fizz bazz buzz. For example, if the user typed in 12, you should see the above output.

3. Mystery function trace

For each call to the following method, indicate what output is produced.

```
1    public void mystery(int n) {  
2        print (n + “ ”);  
3        if (n > 0) {  
4            n = n - 5;  
5        }  
6        if (n < 0) {  
7            n = n + 7;  
8        } else {  
9            n = n * 2;  
10       }  
11       println(n);  
12    }
```

Call

Output

mystery(8);

mystery(-3);

mystery(1);

mystery(0);

4. Optical Illusion

Write a GraphicsProgram that produces the optical illusion below. 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.

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 at 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 *i*, *j*, and the given constants.

