CS161: Introduction to Computer Science
Lab Assignment 9

(Fun with) For Loops

At last we come to the `for` loop! Up until now, we've been using the `while` loop for all our looping needs. However, if you know the number of iterations in advance, a `for` loop is a much better choice than a `while` loop. The purpose of today's lab is to familiarize you with the `for` loop and to give you experience using `for` loops in various scenarios.

A few comments before we begin:

- The textbook does not use curly brackets when there is only one statement inside the body of the `for` loop. See the example below. *This is a bad habit to get into and is an easy way to introduce unintentional errors into your code.* Always use curly brackets with your `for` loops[1]!

```
// Bad
for(int count = 1; count <= 5; count++)
    System.out.println(count);
```

```
// Good
for(int count = 1; count <= 5; count++) {
    System.out.println(count);
}
```

- The variable `count` above controls the number of iterations of the loop. Historically, such variables are named in a similar manner as mathematical subscripting variables: i, j, k, etc. For example, in math you see notation like x_i or x_{ij}. In the same way, in Java it is common to see for loops like:

  ```
  for(int i = 0; i < N; i++){

  }
  ```

  or

  ```
  for(int i = 0; i < N; i++){
      for(int j = 0; j < N; j++){

      }
  }
  ```

  This is one of the only times where it is acceptable to have a variable with a non-descriptive name such as i or j.

[1] and if statements, while loops, etc

Warm Up

Today's lab is a series of standalone programming questions. The starter code has a Java class called `ForLoops` that has a `main` method along with other `private`, `static` methods.

Implement each method (one at a time) and then call the method from `main` to check that your output is correct. For example, after you finish implementing the `printHello()` method, you can call it in `main` using the following syntax:

```
public static void main(String[] args){
    printHello();
}
```

When you run the `main` method you should see "Hello world!" printed 10 times to the screen.

Asterisk Pictures

The first part of this exercise should be completed on a piece of paper. Read the following Java code:

```
for(int i = 0; i < 5; i++){
    for(int j = 0; j < 3; j++){
        System.out.print("*");
    }
    System.out.println();
}
```

On a sheet of paper, with your partner, step through this code and decide what it prints. *After you are done, call me or the lab assistant over to check your answer.*

This is the first time we have seen *nested loops* where one loop is inside of another loop. We can unroll the loops as follows:

```
Set i = 0
     Set j = 0
         print "*"
     Set j = 1
         print "*"
     Set j = 2
         print "*"
print "\n"
Set i = 1
     Set j = 0
         print "*"
     Set j = 1
         print "*"
     Set j = 2
         print "*"
...
```

Open up the second Java class in the starter code called `AsteriskPictures`. There are three methods inside: one for drawing a line, one for drawing a rectangle, and one for drawing a triangle. The first method (for drawing a line) requires only a single for-loop. However, the other two methods require you to use nested for-loops just like above.

Note: The third method asks you to draw a triangle of a given height to the screen (using nested `for` loops). For example, if `height=5` then your method should draw the following picture:

```
*
**
***
****
*****
```

Finally, together with your partner write a fourth method that prints a picture of your choosing. Feel free to experiment with nesting 3 `for` loops, or perhaps putting 2 `for` loops one after the other inside an outer `for` loop.

Submitting your lab assignment

Submit your `lab9` folder with both the `ForLoops` and `AsteriskPictures` classes inside.