INTRODUCTION TO COMPUTER SCIENCE I
LAB 4: LOOPS
Friday, September 29, 2017

1 Setup

Copy several Java files:

```
$ wget -nv -i https://goo.gl/dpWsvJ
```

Open the file `BrokenLoop.java` using emacs. You should see the following code in the `main` method:

```
int num = keyboard.nextInt();
int i = 0;
while(i < num) {
    System.out.print("*");
}
```

This piece of code is trying to print a line of `num` asterisks. Compile and run the program.

OH NO! Instead of printing `num` asterisks in a line, you're probably seeing your terminal window fill up with endless asterisks. You are stuck in an *infinite loop*: the condition `i < num` is never satisfied, so the computer will continue to print asterisks forever. Fortunately, there is a way to kill a program that is currently running. In your terminal window, type Ctrl+c (that is, hold down the Ctrl key and hit c). This will stop the program and break out of the infinite loop.

1. Fix the provided code so that instead of going into an infinite loop, the code actually does print `num` asterisks in a line.

2 Printing with `while` loops

In this section you will use `while` loops to print some cool designs. Open the file `While.java`. Currently the file reads an `int` from the keyboard and stores it in a variable called `size`.

2. Use `while` loops to print a `size` by `size` square. Your square should look something like:

```
+++++
+++++
+++++
+++++
+++++
```

3. Use `while` loops to print a triangle with `size` rows in which the first row has one `^`, the second row has two `^`s, and so on. Your triangle should look something like:

```
^
^^
^^^
^^^^
^^^^^
```

4. Use `while` loops to print another triangle with `size` rows, this time with the ^s aligned to the right. Your triangle should look something like:

```
    ^
   ^^
  ^^^
 ^^^^
^^^^^
```

3 Printing with `for` loops

Anything you can do with a `while` loop you can also do with a `for` loop (and vice versa). The main advantage of `for` loops is that they are specifically designed for cases in which you are counting, i.e., when you know in advance how many times you need to repeat a certain task. In this section you will use `for` loops to make some more designs.

Compile and run `For.java`. You should be prompted to enter an `int`, after which a triangle of the size you specified should print. It should look something like this:

```
^^^^^
^^^^
^^^
^^
^
```

5. Open the file `For.java`. Currently, the code uses `while` loops to print the triangle. Modify this code so that it does the same thing, but uses `for` loops instead of `while` loops.

6. Use `for` loops to print an `X`. Begin by prompting the user to enter an odd number. Then print an `X` with the entered number of rows that looks something like this:

```
X   X
 X X
  X
 X X
X   X
```

7. So far we are assuming that when told to enter an odd number, the user will be sensible and not enter 4 or 26 or 12. But at this point we have seen all the tools we need to check that the entered number is in fact odd. Before your code to print an X, write a loop to make sure that the user has entered an odd number. Continue prompting the user to enter a number until they enter an odd number. (Hint: Do you want a for loop or a while loop for this task? Do you know in advance how many times you will need to prompt the user?)

4 Submit your work

Submit your modified BrokenLoop.java, While.java, and For.java using either the submission web site or the cssubmit command. To submit multiple files, enter them all on the same line after the word cssubmit:

```
cssubmit BrokenLoop.java While.java For.java
```

This assignment is due on Thursday, October 5, 11:59 pm.