

# INTRODUCTION TO COMPUTER SCIENCE I

## LAB 4

### Loops!

## 1 Setup

Grab two Java files:

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

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.

**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 `Patterns.java`. Currently, it prints a of shapes and takes a selection from that menu; then it prompts for a size for the chosen shape. A sequence of mutually exclusive cases are controlled by an `if-then-else` chain. Your task is to **fill in the code that prints each shape**.

Shown below are examples of each shape, given a size of 5:

1. A *square* has `size` plus-signs to a side:

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

2. A *left-aligned triangle* has a base and a height of `size` plus-signs, with the height aligned to the left:

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

3. A *right-aligned triangle* also has a base and a height of `size` plus-signs, but with the height aligned to the right:

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

4. A *non-sectarian Festivus bush*<sup>1</sup> has an asterisk on top (for the star), an absolute-value bar on the bottom (for the trunk), and plus-signs for the body of the tree, like so:

```
  *
  +++
  +++++
  +++++++
  ++++++++
  ++++++++
  |
```

As written, the program will accept and value for the *size*, including negative numbers. A negative size does no harm, but also makes no sense. **Modify the input of the size** such that the user gets repeated opportunities to enter a non-negative size value.

### 3 Submit your work

Submit your modified `BrokenLoop.java` and your `Patterns.java` source code files using either the submission web site or the `cssubmit` command on `remus/romulus`. To submit multiple files, enter them all on the same line after the word `cssubmit`:

```
[sfkaplan@remus ~/lab-4]$ cssubmit BrokenLoop.java Patterns.java
```

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

---

<sup>1</sup>OK, OK, it's a Christmas tree.