Mini-Lab: Loops

Goal:

To demonstrate the usage of loops. Most programs require you to use loops, since you will most likely be doing things multiple times. Understanding how loops work, loop syntax, and when to use which type of loops becomes very important.

Why loops?

Suppose you want to write out the multiplication tables for 1 to 20. You could do out all the math and then write a bunch of System.out.println()'s, or you could use loops to do the work for you. For instance:

This prints out the multiplication tables for 1 to 20.

```
for(int i = 0; i <= 20; i++) {
  for(int j = 0; j <= 20; j++) {
    System.out.println(i*j + " ");
  }
  System.out.println();
}
```

This is much easier than taking the time to copy all of the correct numbers into a System.out.println().

Mainlines:

In java, sometimes you don't want to use Applets. Suppose you are writing a program that doesn't require any graphics; the Applet will just come up empty, and this is a waste of resources. Instead, we just want to run our program without having to use an appletviewer, so we need a mainline. A mainline is what gets called when you run a java program. For instance, the mainline that you usually use is in the super class of the GP.Containers.Applet, so it has been written for you. This time, you need to write the mainline yourself. Don't worry, it's really easy.

Getting Started:

So, to get started go to a shell in your account. Type in the following line "cs015_install loops" Now type "cd loops," without the quotes. Now type xemacs &, this should bring up xemacs. Click on open, and select the file loops.java. Now you're ready to go.

The Lab:

Now that you have the lab up in front of you, read through the comments. You need to fill in 3 methods:

DrawTriangle1():

Should Draw:

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

You may use (1) System.out.println() and (1) System.out.print().

DrawTriangle1():

Should Draw:

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

You may use (1) System.out.println() and (2) System.out.print()’s.

DrawTriangle1():

Should Draw:

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

You may use (1) System.out.println() and (2) System.out.print()’s.