CS170 **Lab 5**
Fall 2012 **Simple for-loops**

Due: Midnight tonight

Objectives of this lab:

- Practice loops using the for-statements
- Use command line arguments

Exercise Preparation:

- Start a terminal application and prepare your `lab5` directory:
  - `mkdir ~/cs170/lab5`
  - `cp ~cs170003/share/lab5/* ~/cs170/lab5`
  - `cd ~/cs170/lab5`
  - `ls`
  - You should see 3 files: `Sample.java`, `Triangle1.java` and `Triangle2.java` If you do not see these files, ask the TA for help.

Sample.java:

- Open the `Sample.java` file using gedit:
  - `gedit Sample.java &`
- This file has a method called `indentCharLine(i, c, n).`
  - This method will first print `i` spaces
  - Followed by the character `c, n` times.
  - Examples (using – to represent a space):
    - `indentCharLine(4, 'A', 6)  will print:  ----AAAAAA`
      `indentCharLine(6, 'X', 3)  will print:  ------XXX`
- The `main` method shows you how to use the `indentCharLine` method to make a square.
- This program requires you to use a command line argument to specify the size (height and width in terms of the number of characters) you would like to make the square.
  - For example, using a command line argument of 3 will make the square 3 characters high and 3 characters wide.
- Compile run `Sample.java` to be sure you understand how it works:
  - `javac Sample.java`
  - `java Sample 3`
  - `java Sample 6`

Task 1: Complete `Triangle1.java`

- Open the file provided for this lab.:
  ```
  cd ~/cs170/lab5
  gedit Triangle1.java &
  ```
- You should see the "skeleton" of a (incomplete) program in your editor. If you see a blank

window, ask your TA for help.

- Your task is to complete the `main` method.
- The comments in the file will guide you in what to do.
- Make use of the `indentCharLine` method inside your for loop to make your code simple.
- Remember to make small changes, compile, and check your work. Don't try to write the entire program first, before you test it out.
- This program also take 1 command line argument: the height (number of rows) of your triangle.
  - Example:
    `java Triangle1 5`
    will output:
    ```
    X
    XX
    XXX
    XXXX
    XXXXX
    ```

Task 2: Complete `Triangle2.java`

- Open `Triangle2.java` for editing in `gedit`.
- Again, you should see the "skeleton" of an (incomplete) program. If you do not, ask your TA for help.
- The comments in the file will guide you in what to do in completing the `main` method. Several things will be very similar to `Triangle1.java`.
- The direction of the triangle will be changed for this program.
- Again, the program takes a command line argument specifying the height of the triangle.
  - Example:
    `java Triangle2 5`
    will output:
    ```
        X
       XX
      XXX
     XXXX
    XXXXX
    ```

Turning in your work:

- Save all your work and close `gedit`.
- Turn in your lab 4 work using these commands:
  ```
  cd ~/cs170/lab5
  /home/cs170xxx/turnin-lab Triangle1.java lab5a
  /home/cs170xxx/turnin-lab Triangle2.java lab5b
  ```
  - ***Note: you will need to replace 'xxx' with your section number.***
- These commands makes a copy of each of your files in the grader's account.
- Your turnin is successful when you see a success message for each file like:
  ```
  Program `Triangle1.java' has been turned in by YOUR_ID as
  lab4a
  ```