

INTRODUCTION TO COMPUTER SCIENCE I

LAB 4

Method calls

This lab exercises your ability not only to write methods and call them, but also to know when and how to write them. There will be, within this assignment, particular tasks that are best encapsulated in a method.

1 Getting started

Begin in the usual way:

1. **Login** to `remus` or `romulus`.
2. **Open a terminal** to get a shell prompt.
3. **Make a directory** named `lab-4` (using `mkdir`) and change into it (using `cd`).
4. **Copy the initial source code** as follows:

```
$ cp ~sfkaplan/public/COSC-111/lab-4/Triangles.java .
```

5. **Open the source code for editing** (using `emacs`).

2 Your assignment

2.1 Printing a pattern

Your program should do something (seemingly) simple: Take a single number from the user, and then print a pattern to the screen. Now for the details...

First, ask the user for a single, positive (that is, non-negative and non-zero) integer. Your program should repeatedly request integer values until the user provides a positive one.

Given that integer (let's call it n), you will print a sequence of n "triangles" to the screen, where each triangle is composed of stars (that is, asterisk characters). The first triangle will be $n \times n$, the second (assuming $n > 1$) will be $(n - 1) \times (n - 1)$, etc., down to the 1×1 triangle. So, if $n = 4$, then the output would be:

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

*
**
***

*
**

*
```

2.2 What you must do

In the code I've provided, `main()` calls on two methods:

- `getPositiveInt()`: This method should require the user to input a positive integer. Once it has obtained that integer, it should return it.
- `printPattern()`: This method should print the full pattern of triangles, where each triangle is one size smaller than the previous one, down to a triangle that is just one asterisk. The number of triangles to print (and thus the size of the first, largest triangle) is provided by the caller.

In addition to these two methods, which must exist for the code to compile and run, you should write the following two supporting methods to make the task easier:

- `printTriangle()`: This method should print an entire triangle, where the size of the triangle (that is, the number of rows) is provided by the caller.
- `printRow()`: This method should print the number of asterisks (*) provided by the caller on a single row.

That is, `printPattern()` should rely on `printTriangle()` to print each triangle in the pattern; in turn, `printTriangle()` should rely on `printRow()` to print each row of a given triangle. You must **write these methods** and get the program to do its thing.

3 How to submit your work

Use the CS submission systems to submit your work. Specifically, you will need to submit your `LeapYear.java` file. You may use either of the following two methods, while connected to `remus` or `romulus`, to use the submission system:

- **Web-based:** Visit the submission system web page.
- **Command-line based:** Use the `~lamcgeoch/submit` command at your shell prompt.

This assignment is due on Thursday, Feb-25, 11:59 pm, before it becomes Friday, Feb-26.