

## CS 122 – Lab #2 – Practice with if-statements and loops

In today's lab, you will write two programs. The first one focuses on Java's `if` statement. The second program practices with loops (and also if statements).

Before we dive in, there is a little housekeeping you should do. Last week, you became acquainted with your computer account on the server `cs.furman.edu`. You will be using it all term, and over time you will accumulate a lot of work. So, let's organize your files into folders.

Log in to your `cs.furman.edu` account. If you have not done so already, create a directory to contain all the work for this class. You could, for example, simply call the directory `cs122`. If you use this account for other CS classes or research projects, you could similarly create directories for them as well.

Use the `cd` command to enter your `cs122` directory. In here, you should also create subdirectories. I recommend a new directory for each lab and for each homework assignment. You should also a place set aside for other programs that you write, and for programs that you download and study. With this in mind, you should have a directory called `lab01` and another one called `lab02`. Put all the files you did in last week's lab into the `lab01` directory. Today's work will go into the `lab02` directory.

One more thing you should do is to protect your account from prying eyes. By default, your account is set up so that other people can see your files. To prevent this and make your files only accessible by you, type this command: `chmod 700 ~`. You only have to do this once.

Have the instructor or lab aide check your work when you are finished. Enjoy!

### First program: Converting military to civilian time

Write a program called `Time.java`. It will take a single command-line argument: the military time expressed as an integer. The program should print out the equivalent civilian time. For example, if the user runs the program as `java Time 1900`, then the output should read `7:00 pm`.

Military time is based on a 24-hour clock. The time is expressed without any am/pm designation and also without the colon. The possible values of military time range from 0000 to 2359.

Hints:

- After reading the input, it will be helpful for you to create separate variables for the hour and minute. You should divide and mod by 100 in order to separate out the hour and the minute from the input.
- When you are ready to print the output, you should do so in three phases. First, print the correct civilian hour, which is a number 1-12. Second, you should print a "0" as needed if the number of minutes is small. Finally, you should reconcile the am versus pm based on the military hour.

Run your program with several different inputs to verify that it works correctly.

## Second program: Number triangles

Write a program called `Count.java`. The purpose of this program is to practice writing *nested loops*. Let's have the program print a triangle of numbers like this one:

```
1
1 2
1 2 3
```

The program will take one command-line argument, which we can call `n`. The number `n` will tell us how big the triangle should be. The above example corresponds to `n = 3`. There are 3 rows, and on each row, we print one more number than the one before it.

Once you are able to print the triangle correctly, modify your program so that it prints two additional triangles. If we run as `java Count 4`, then `n = 4`, and the final output should look like this:

```
1
1 2
1 2 3
1 2 3 4

1 2 3 4
1 2 3
1 2
1

4 3 2 1
4 3 2
4 3
4
```

You should handle each of the three triangles with its own nested loop. Please print a blank line between each triangle.

Hint: To print a triangle of numbers, you need a nested loop. Customarily we use the variable `i` in the outer loop and `j` in the inner loop. The value of `i` should have something to do with how one row of the triangle is different from the other rows. For example, `i` could represent the last number that should be printed on each row. The value of `j` refers to the numbers printed within a single row. For example, to print the row "1 2 3" means we want `j` to range from 1 to 3, inclusive, in that order.