INTRODUCTION TO COMPUTER SCIENCE I
LAB 2
Conditional statements

This lab will allow us to use and experiment with *conditional statements* (a.k.a., `if-then-else` statements). Such statements will be indespensible for our task of creating a program that can determine whether a given year is (or is not) a *leap year*.

1 Getting started

In order to get started with this assignment, do the following:

1. **Login:** Just as you did for Lab-1, use Remote Desktop to connect to `remus` or `romulus`.
2. **Open terminal:** Right-click on the desktop (within Remote Desktop!) and use the drop-down menu that appears to open a Terminal window.
3. **Make a directory:** Create a new `lab-2` directory with the `mkdir` command, and then change into that new directory with the `cd` command.
4. **Copy the initial source code:** Obtain a Java source code file that contains the beginnings of your new program, named `LeapYear`. As always, be careful to include both the tilde (˜) before my username and the trailing space followed by a period (`.`):

```
$ cp ˜sfkaplan/public/COSC-111/lab-2/LeapYear.java .
```

5. **Open the source code for editing:** Use `emacs` to open this new program's code. It is here that you will complete the code. Remember that each time you change the code, you must *save, compile, and execute (run)* it in order to test your most recent work.

2 What makes a year a *leap year*?

The rules for what constitute a *leap year*—one that has the extra day of February 29—may be just a touch more complex than you realized. Specifically, a given year is a *leap year* if and only if...

1. the year is divisible by 4 (that is, division leaves no remainder)...
2. but **not** divisible by 100...
3. except if it **is** divisible by 400.

In other words, 2017 will not be a leap year, but 2016 is, while 2100 is not, yet 2400 will be.

One last caveat: leap years have only existed since 1753. Any year before then is not a leap year, although in the special way that the entire concept didn't apply.

3 Your assignment

Complete the source code provided in `LeapYear.java` such that, after it obtains a year number from the user, it then prints whether the given year *is* or *is not* a leap year. Should the user enter a year that is too early (pre-1753), the program should print that the question of leap-year-ness applies only to 1753 and later.

4 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:

1. **Web-based:** Open a browser[1] and point it at:

   ```
   https://www.cs.amherst.edu/submit
   ```

2. **Command-line based:** From the shell in your terminal window, submit your work via the following command:

   ```
   $ ~lamcgeoch/submit LeapYear.java
   ```

   You must, of course, be within your `lab-2` directory. Once you issue this command, you will be prompted to select the correct course and assignment for which you are submitting work; those choices should, I hope, be self-evident.

This assignment is due on Thursday, Feb-11, 11:59 pm, before it becomes Friday, Feb-12.

[1] Be sure to open it from within the server or the workstation! That is, within Remote Desktop, go to the start menu and select Firefox to open a browser that runs on `remus/romulus`.