CS5004-5005-SV – Lab 2 – May 14, 2020
Java Primitive Types and Strings Practice

The goal of today's lab is to become familiar with Java Data Types and key classes such as String, including their methods.

0. Report on and resolve any issues with access to CS5005 Canvas. If you did not upload your lab1.txt file already, please wrap that up. Please check for feedback from one of the TAs, on Canvas-CS5005.

1. Complete homework PS1. Submit the homework to **Canvas for 5004** as **problem1.txt.** Be sure to read the Oracle docs re. the remainder (%) operator. NB: You may find **jshell** helpful in working Problem 1. https://docs.oracle.com/javase/specs/jls/se8/html/jls-15.html#jls-15.17.3 (Be sure to turn this in on for CS5004. Only the lab portions below should be submitted to CS5005.)

2. Create a Class file called **DataTypes.java**, with a main method. It should declare and initialize one variable for each of the 8 primitive types described in Lecture. It should print to the console the name of the primitive type, the name of your variable, and the value assigned, one per line.

3. Create a Class file called StringFun.java, with its own **main** method. Note that when using an IDE such as VSCode or IntelliJ, you will want to keep things in separate folders for each project. In particular, a given folder should only contain *one* Class with a **main** method. Your program should demonstrate the following String operations:
    a. **concatenation** (see slide 5 from lecture)
    b. length
    c. 0-origin indexing and that blanks and periods count, using the **charAt** method
    d. **substring**, with either 1 or two inputs. Show that the character at the first input is included in the substring, but the character at the second input is not included.

   e. **Equals** and **equalsIgnoreCase**, demonstrating the difference
   f. **toLowerCase** and **to UpperCase**
   g. **trim**
   h. **indexOf,** including an example with >1 occurrence and an example with no occurences
   i. **lastindexOf**, also illustrating above cases
   j. **compareTo**, showing various cases including mixed case strings
   k. **compareToIgnoreCase**
   l. Explain**:** Since Strings are immutable, why am I able to I reassign the value of a variable declared as a String?
   m. Imagine an application that requires a string with the double-quote character inside the String (perhaps to teach a Java class?). How can I create a string with a quote character in it?

4. Variations on System.out.println (extension of part 2).
   a. Create a simple Class and main method using System.out.print as well as System.out.println
   b. Use System.out.printf, illustrating format specifications for at least primitive datatypes

5. Create a Class Echo, to read and echo user input from its **main** method. Start with **import java.util.scanner**. It is sufficient for this lab to read and print strings, as well as trying out the **nextLine** method. See slides 39-45. Vary your example from what was in the lecture!

6. [OPTIONAL!] Download the Code Format Sample from Piazza, illustrating proper Java documentation and other formatting. Note that the block comments that start with /** instead of just /* are special “javadoc” comments that can be used to automatically create a documentation web page for your program. From the terminal, create a folder called javadocs in the same folder as the downloaded .java code and cd into that folder. Execute from Terminal: Javadoc Calculator.java. It will generate a bunch of files ending in .html, .css, .js, and so on. Open index.html with a web browser to see where this is going! If you succeed, please capture a screen shot of the resulting .html page and upload with the rest of the lab materials. If not, no worries!

In a future lab we will look at how to install extensions to VSCode to create javadoc output from *within* that editing environment and how to better direct the resulting output.

Please submit the *Lab* portions of this handout as Lab2 on Canvas for CS5005. Tonight if you are done, or at least before next week's lab session. The sooner you submit, the sooner our TAs can provide helpful feedback.