Lab 1

Objectives

- Exposure to writing and using tests in Java

Associated Lecture Videos

- Command-line
- First Java Program
- Variables, Types, Casting
- Methods
- If and switch statements
- Loops
- Debugging
- Arrays

CHECKPOINTS

- **CHECKPOINT** areas are found in each week's lab outline. **CHECKPOINTS** are places where it might be a good idea to check in with your lab TA before progressing if there are things that are unclear to you.
- You **do not need** to show the TA your progress each **CHECKPOINT**; you are only required to submit the file specified at the end of the lab to receive lab credit.

Part I

1. Download Lab1Tester.java and Lab1.java
2. Lab1Tester.java is a program written to test the Lab1.java program**, BUT** there are logic errors in it. You will need to fix the errors using the Lab1Tester output to help identify them.
    a. Using the command line (cmd in Windows, Terminal in Mac OSX) go to the directory you saved the files to, and then compile and run the Lab1Tester program

        To compile: javac Lab1Tester.java
        To run: java Lab1Tester

        Additional resources for compiling, executing, and debugging a Java Program:
        - The lecture videos listed above
        - Resources found on CSC Assistance Centre web-page:
            - select the **"Compiling Java programs on Windows"** link
            - scroll to heading "**Writing a Simple Java Program**"

CHECKPOINT (Ungraded) – If you are struggling with compiling, running, or debugging your Java program, you should get TA help before proceeding to the next section

    b. Identify the first test that is failing. Uncomment the print statement before the test to give you extra information on what the method is returning relative to what it should return.
        - **DO NOT CHANGE** how the tester calls the method (in Lab1Tester.java), you can assume each method is always called correctly.
    c. Fix the method causing the error in Lab1.java, save the file after you have made a change, recompile and rerun the Lab1Tester.java file until the test passes.
    d. Repeat steps b and c until all tests pass one method at a time for methods `getString`, `countAbove`, and `getAverage`.

CHECKPOINT (Ungraded) – If you are unable to find where the bug is in the method, please don't hesitate to ask a TA for assistance. Make sure you have fixed all three methods before proceeding.

3. Within Lab1Tester.java the testGetMax method needs tests to be added.
    a. Given the arrays defined at the top of the program, what should the maximum value returned be for each array?
    b. Write tests for the getMax method found in Lab1.java by calling the dispayResults method similar to how the tests methods you worked with earlier do.
    c. Once you have written tests in Lab1Tester.java, save, compile, and run the file to determine whether the getMax method was implemented correctly.
    d. Based on the test results, identify and then fix getMax if necessary, until all of the tests pass.
        i. Remember, any time you make a chance to either file, you, must save, compile, and then execute in order to see the result of the changes.

CHECKPOINT (Ungraded) – When you run Lab1Tester.java there should now be more test results showing (ie. at the bottom it should say Passed x/y tests) for the tests you added to testGetMax. If the number of test results is not higher than before, when you simply tested the first three methods, or there are tests that are still failing, be sure to check in with a TA.

4. Uncomment the call to the testIsSorted method in the main method of Lab1Tester.java.
    a. Save, compile, and run Lab1Tester.java to view the test results for the isSorted method
    b. Based on the test results, identify and then fix the isSorted method if necessary. until all of the tests pass within the testIsSorted method.
        i. Remember, any time you make a chance to either file, you, must save, compile, and then execute in order to see the result of the changes.

SUBMISSION (graded) – Submit the Lab1.java file into the Lab1 submission page on ConneX.