Assignment 1

Objectives

- Introduction to Java
- Review methods, loops, and arrays
- Exposure to testing

Introduction

In this assignment, you will write and test a set of methods. You're provided with a tester that tests the methods. Some of the methods are completed for you, but the majority of them you will need to write yourselves until all of the tests pass.

Quick Start

1. Download A1Tester.java and A1Exercises.java into the same directory.
2. Read A1Tester.java and A1Exercises.java carefully. A1Tester.java contains all of the tests; it is the file you will compiling and executing. A1Tester.java contains all of the methods you will need to implement to complete the assignment.
    a. Compile and run the A1Tester program from the directory you downloaded the files to.
    To compile: javac A1Tester.java
    To run: java A1Tester
    b. The output should show that some tests are passing, but some are failing. Find the method with failing tests, fix the method, then recompile and rerun the tester until the tests pass. See the section below entitled *Understanding the Test Programs* for more information.
3. Implement each method in A1Exercises.java, by repeating the following steps:
    a. Uncomment a test in the corresponding method in A1Tester.java
    b. Compile and run A1Tester to ensure you have defined the method correctly.
    c. Implement the method by completing the code within the method
    d. Compile and run until all tests for the method pass.

CRITICAL: You **must** name the methods in A1Exercises.java as specified in the documentation (above the purpose) or you will receive a **zero grade** for that method (as the tests will not compile correctly). You **cannot** use java.util.Arrays methods or you will receive a **zero grade** for that method.

Submission and Grading

Submit **A1Exercises.java** with your name and student ID at the top of the file using conneX**.**

If you chose not to complete some of the methods required, you **must provide a stub for the incomplete method(s)** in order for our tester to compile. I have provided some examples of a **stub** for you (**numFactors** and **isPrime**). Notice that these methods have a correct signature (name, return type, and parameter list), allowing the tester to call the methods, but their implementation is not correct. If you submit files that do not compile with our tester, you will receive a **zero grade** for the assignment. It is your responsibility to ensure you follow the specification and submit the correct files. Additionally, your code must **not** be written to specifically pass the test cases in the tester, instead, it must work on all valid inputs. We may change the input values when we run the tests and we will inspect your code for hard-coded solutions.

Be sure you submit your assignment, not just save a draft. **ALL late** and **incorrect** submissions will be given a **ZERO** grade. A reminder that it is OK to talk about your assignment with your classmates, but not to share code electronically or visually (on a display screen or paper). We will be using plagiarism detection software.

Understanding the test program

A1Tester.java tests your implementation of A1Exercises.java

The first things you should do after downloading the source files is to compile and run the test program:

Compile the test program by typing: `javac A1Tester.java`
Run the test program by typing: `java A1Tester`

You should see the following output:

```
Passed test: isFactor(3,3)

Passed test: isFactor(3,2)

Passed test: isFactor(8,2)

Passed test: isFactor(42,5)

Passed test: isFactor(102,17)

Passed test: calcPower(2,0)

Passed test: calcPower(2,3)

Passed test: calcPower(5,4)

Passed test: calcPower(13,2)

Failed test: numFactors(1) at line 102

Failed test: numFactors(6) at line 106

Failed test: numFactors(24) at line 111

Failed test: numFactors(51) at line 116

Passed 9/13 tests
```

The first 9 lines are the output from the two methods implemented for you. They report that all of the tests passed for the isFactor and calcPower methods. These two methods are tested in A1Tester.java but implemented in A1Exercises.java.

The next 4 lines are reporting that your implementation is failing in the testNumFactors method at lines 102, 106, 111, and 116. The final line reports that you passed 9 test of the 13 total tests that were run. This is because we have only provided you with a stub for the numFactors method (it compiles and runs correctly, but there is a logic error in the implementation).

Uncommenting the print statements in the test methods can help you to determine why the test is failing. They will print out what the result should be, and what your method is currently returning:

```
//System.out.println("Expected: "+expected+", result: "+result);
```

Once you complete the implementation of a method, the Failed test lines should change to Passed.