Assignment 1

Reminder: Your code is to be designed and written by only you and not to be shared with anyone else. See the Course Outline for details explaining the policies on Academic Integrity. Submissions that violate the Academic Integrity policy will be forwarded directly to the Computer Science Academic Integrity Committee.
All materials provided to you for this work are copyrighted, these and all solutions you create for this work cannot be shared in any form (digital, printed or otherwise). Any violations of this will be investigated and reported to Academic Integrity.

Objectives

- Introduction to Java
- Review methods, arrays, classes
- Exposure to testing

Getting Started

1. Download `A1Tester.java` and `A1.java` to the same directory.
2. Read `A1Tester.java` and `A1.java` carefully.
    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. Fix the method, recompile and rerun the tester until the tests pass. See the section below titled *Understanding the Test Programs* for more information.
3. Implement each method in `A1.java`, by repeating the following steps:
    a. Uncomment a test in the corresponding method in `A1Tester.java`
    b. Add a stub for the method in `A1.java`
    NOTE: a stub is an empty method with only the signature and a return statement that returns some "dummy" value if the function has a non-void return type
    c. Compile and run to ensure your tests and stubs are correct
    d. Implement the method by completing the stub
    e. Compile and run (repeat steps d and e until all of your tests pass)

CRITICAL: You **must** name the methods in `A1.java` as specified in the documentation (above the purpose) or you will receive a **zero grade** for that method.
You **cannot** use `java.util.Arrays` methods or you will receive a **zero grade** for that method.
The code you submit must compile

Submission and Grading

Submit the file named **`A1.java`** with the completed methods through the assignment link in BrightSpace.

- If you chose not to complete some of the methods required, you **must at least provide a stub for the incomplete method** in order for our tester to compile.
- If you submit files that do not compile with our tester (ie. an incorrect filename, missing method, etc) you will receive a **zero grade** for the assignment.
- Your code must **not** be written to specifically pass the test cases in the testers, 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.
- **ALL late** and **incorrect** submissions will be given a **ZERO** grade.

Understanding the test program

`A1Tester.java` tests your implementation of `A1.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:

```
Expected: 15
Returned: 0
Failed test: testSum with 2, 13 at line 88

Expected: -11
Returned: 0
Failed test: testSum with 2, -13 at line 95

Passed 0/2 tests
```

The first 3 lines are the output of the first test. The first line tells you what the function is expected to return and the second line tells you what the function is actually returning. For a test to pass, these values must match. The third line tells you the test has failed and that the values used in the test were 2 and 13 and that this failing test is at line 88 of `A1Tester`.

The next 3 lines are the output for the next test, which again is failing. Once you complete the implementation of the `sum` method within `A1.java`, those `Failed test` lines should change to `Passed...` and the last line of output will say: `Passed 2/2 tests`

See ***Video: automated testing*** in Week 2, Monday BrightSpace Module for a more in-depth explanation of this testing process.