

Assignment 1

Objectives

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

Introduction

In this assignment, you will write and test a set of methods that operate on an array (or arrays). 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 ArrayOperations.java to the same directory.
2. Read A1Tester.java and ArrayOperations.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. The output should show that some tests 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 ArrayOperations.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 ArrayOperations.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. This is done for you in the arrayProduct method. The method is not correct, but does return a value so that the tester can at least test if (even if the tests fail).
 - c. Compile and run to ensure your tests and stub are correct
 - d. Implement the function by completing the stub
 - e. Compile and run (repeat steps d and e until all of your array tests pass)

CRITICAL: You **must** name the methods in ArrayOperations.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.

Submission and Grading

Submit **ArrayOperations.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. 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 ArrayOperations.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:

```
{}  
{8}  
{2,3,2,0}  
{-1,2,4,1,3}  
Passed test: testSumArray  
Passed test: testSumArray  
Passed test: testSumArray  
Passed test: testSumArray  
Passed test: testProductArray  
Failed test: testProductArray at line 102  
Failed test: testProductArray at line 106  
Failed test: testProductArray at line 110  
Passed 5/8 tests
```

The first 4 lines are the output of the call to the `printArray` method to print out the values in `array0`, `array1`, `array2` and `array3` declared globally in the program.

The following 4 lines report that the `sumArray` method passed all of the tests (`sumArray` has been implemented for you). The `testSumArray` method in `A1Tester` tests the `sumArray` method in the `ArrayOperations` class.

The next 4 lines are reporting that your implementation is failing in the `testProductArray` method at lines 102, 106, and 110 and the last line says you passed 5 test of the 8 total tests that were run. This is because we have only provided you with a stub for the `testProductArray` method.

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("should be 2: " + result);
```

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