

Assignment 1

Objectives

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

Introduction

This assignment has two parts. Part I requires you to write and test a set of methods that operate on an array.

Many companies require their employees to travel for work using their own cars and staying in hotels. Those employees get these expenses reimbursed based on the time and distance have been are gone. Part II requires you to write a program that can calculate the expense reimbursement for a single person as well as the expense reimbursement of a population of multiple people.

In this assignment you are provided with a tester that tests the methods you are to write in both parts of this assignment.

Quick Start

Part I

1. Download ***ATTester.java*** and ***ArrayOperations.java*** to the same directory.
2. Read ***ATTester.java*** and ***ArrayOperations.java*** carefully.
 - a. Compile and run the ***ATTester*** program from the directory you downloaded the files to.
To compile: `javac ATTester.java`
To run: `java ATTester`
 - b. Fix the method, 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. Implement the function by removing the supplied stub code and completing the function according to the specifications given in the comments
 - b. Compile and run (repeat step a until all of your array tests pass)

You **cannot** use `java.util.Arrays` methods or you will receive a **zero grade** for that method.

Part II

1. Download ***expense_calculator_tester.py*** and ***expense_calculator.py*** for your reference. This is a full implementation of the Expense Calculator in Python.
2. Download ***ExpenseCalculator.java*** to the same directory as Part I.
3. Implement each method in ***ExpenseCalculator.java***, again following the steps:
 - a. Implement the function by completing the stub
 - b. Compile and run (repeat step a until all of your expense tests pass)

Understanding the test program

AlTester.java tests your implementation of ***ArrayOperations.java*** and ***ExpenseCalculator.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 AlTester.java`

Run the test program by typing: `java AlTester`

You should see the following output:

```
{2}
{2, 1, 3, 0}
{-1, 2, 4, 1, 3}
Failed test: testProductArray at line 82
Failed test: testProductArray at line 86
Failed test: testProductArray at line 90
Failed test: testProductArray at line 94
Failed test: testMaxArray at line 104
Failed test: testMaxArray at line 108
Failed test: testMaxArray at line 112
Failed test: testMinArray at line 120
Failed test: testMinArray at line 124
Failed test: testMinArray at line 128
Failed test: testEqualArrays at line 140
Passed test: testEqualArrays
Failed test: testEqualArrays at line 148
Passed test: testEqualArrays
Passed test: testEqualArrays
Passed test: testEqualArrays
Failed test: testShiftBy at line 175
Failed test: testShiftBy at line 182
Failed test: testShiftBy at line 189
Failed test: testShiftBy at line 196
Failed test: testShiftBy at line 203
Failed test: testMileage at line 213
Failed test: testMileage at line 218
Failed test: testMileage at line 223
Failed test: testMileage at line 228
Failed test: testCalcFood at line 238
```

```
Failed test: testCalcFood at line 243
Failed test: testCalcFood at line 248
Failed test: testCalcFood at line 253
Failed test: testCalcHotel at line 264
Failed test: testCalcHotel at line 270
Failed test: testCalcTotalExpense at line 278
Failed test: testCalcTotalExpense at line 282
Failed test: testCalcTotalExpense at line 286
Failed test: testCalcAllExpenses at line 299
Failed test: testCalcAllExpenses at line 309
Passed 5/37 tests
```

The first 3 lines are the output of the call to the **printArray** method to print out the values in a1, a2 and a3 declared globally in the program.

The remaining lines show that you are failing most of the tests – that isn't a surprise because you haven't written any code yet!

Work through the methods until all your tests are passing.

Note that due to the supplied stub methods, the tester is reporting that your code passes some of the **arraysEqual** tests – if you do not write to to meet the specification of the **arraysEqual** method, you will not receive points for tests passed by the stub.

Uncommenting the print statements in the test methods can help you to determine why the test is failing:

```
System.out.println("should be 2: " + result);
```

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

Submission and Grading

Submit the following files with your name and student ID at the top of each file using conneX:

ArrayOperations.java and **ExpenseCalculator.java**

If you submit files that do not compile, or that do not use the correct method names you will receive a **zero grade** for the assignment. It is your responsibility to ensure you follow the specification and submit the correct files.

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.

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, and you are encouraged to design solutions together, but each student must implement their own solution. We will be using plagiarism detection software on your assignment submissions.