CS 1331 Homework 6

Due Thursday October 4th, 2012 8:00 PM

Introduction

This homework will cover arrays, interfaces, and static.

You will be creating program that performs common functions on arrays.

Be sure to name your classes as required by the instructions. Also be sure to use good coding style and **indentation**, and to use appropriate and descriptive variable names. Seriously, indentation. I'm not helping anyone debug their code if I can't follow it because of their indentation.

If you are feeling comfortable with Java, now is probably a good time to switch to a more powerful IDE. **Eclipse** is a great IDE, and is actually required to develop in some frameworks (Android, Hadoop, …).

Do not forget about javadocing and commenting detailed in previous homeworks.

6.1: ArrayServices.java

Write an `ArrayServices` class that implements `ArrayServicesInterface`.

Do not edit the `ArrayServicesInterface` file.

Notice that the methods take in arrays to work with and **do not use instance variables.**

1. The instructions for each method are detailed in its javadoc. Note that the javadoc is *above* its respective method.

2. Add the main method below to test your methods. Just because your methods pass these tests do not necessarily mean they are correct. Feel free to add your own tests.

3. You will need to start writing more of your own test cases. For each test case you write, make sure that the test case and result prints out in some discernible fashion – you must be able to identify the test case from the print out without having to refer to the code! In addition, your test methods should either show the expected output along with the actual output **and/or** the output along with a pass/fail message

    - Add test for findIndexOf
    - Add test for standard deviation
    - Add test for increment
    - Add test cases for **all** remaining methods

```
public static void main(String[] args) {

      ArrayServicesInterface as = new ArrayServices();

      int[] array = {1,1,2,2,2,4,6,-1,10,5};

      System.out.println("Testing average:");
      System.out.println("Expected value: 3.2");
      System.out.println("Computed value: " + as.average(array));

      System.out.println();

      System.out.println("Testing indexOfMax:");
      System.out.println("Expected value: 8");
      System.out.println("Computed value: " + as.indexOfMax(array));

      System.out.println();

      System.out.println("Testing indexOfMin:");
      System.out.println("Expected value: 7");
      System.out.println("Computed value: " + as.indexOfMin(array));

      System.out.println();
```

```
        System.out.println("Testing contains:");
        System.out.println("Expected value: true");
        System.out.println("Computed value: " + as.contains(array, 4));

        System.out.println();

        System.out.println("Testing update:");
        System.out.println("Expected value:
                              [1, 1, 3, 3, 3, 6, 9, -1, 15, 7]");
        as.update(array, .5);
        System.out.println("Computed value: " + as.arrayToString(array));

        System.out.println();

}
```

You **cannot import java.util.Arrays;**

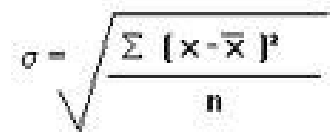
If your file does not compile, or your methods give an unexpected value, then something is incorrect with your methods.

You may just copy the javadoc's into your file, modifying where necessary (especially the class javadoc).

Usually, methods like these would be static, as they don't use any instance data. However, since you are implementing an interface the methods must be instance methods. If you are ever writing something like this in the future that does not use instance variables, it would probably work better as a static method.

Hints and Reference Material

- **Standard Deviation**: The mathematical formula for standard deviation for a set of elements called “X” whose length is “n” is:



σ = standard deviation

$\sum$ = sum of

x = each value in the data set

$\overline{x}$ = mean of all values in the data set

n = number of value in the data set

- This can be read as “For each element, subtract from it the average of all the elements and square the difference. Add these squared differences together, divide the sum by the total number of elements, and then take the square root.”
- Reminder – Do not modify ArrayServicesInterface!

Turn-in Procedure

Turn in the following files on T-Square. When you're ready, double-check that you have *submitted* and not just saved as draft.

- ArrayServices.java
- ArrayServicesInterface.java

All .java files should have a descriptive javadoc comment.

Verify the Success of Your HW Turn-In

Practice "safe submission"! Verify that your HW files were truly submitted correctly, the upload was successful, and that the files compile and run. It is solely your responsibility to turn in your homework and practice this safe submission safeguard.

1. After uploading the files to T-Square you should receive an email from T-Square listing the names of the files that were uploaded and received. If you do not get the confirmation email almost immediately, something is wrong with your HW submission and/or your email. Even receiving the email does not guarantee that you turned in exactly what you intended.
2. After submitting the files to T-Square, return to the Assignment menu option and this homework. It should show the submitted files.
3. Download copies of your submitted files from the T-Square Assignment page placing them in a new folder.
4. Recompile and test those exact files.
5. This helps guard against a few things.
    a. It helps insure that you turn in the correct files.
    b. It helps you realize if you omit a file or files.**
    (If you do discover that you omitted a file, submit all of your files again, not just the missing one.)
    c. Helps find last minute causes of files not compiling and/or running.

**Note: Missing files will not be given any credit, and non-compiling homework solutions will receive few to zero points. Also recall that late homework (past the grace period of 2 am) will not be accepted regardless of excuse. Treat the due date with respect. The real due date and time is 8 pm Thursday. Do not wait until the last minute!