CS 342 Homework#1 Fall 2019

Maven and Java

Due Date: Sunday, September 15th, by 11:59PM

Description:

In this homework, you will run a Maven project in your development environment of choice. You will also create a new .java file with a few methods

Implementation Details:

What you have:

Download the zip file from Blackboard and go to the HomeworkOne directory. This is the Maven project you will be working with. It is a small program that can calculate some basic statistics (mean, median and standard deviation) from numbers read in from a text file.

Take a look in the src/main/java folder. You will find two files: **HomeworkOne.java** and **Statistics.java**. **HomeworkOne.java** just contains the main method. You will find some code there but no need to change it.

Statistics.java is a class that reads in values from a file into an array. This class makes calls to a **StatFormulas** class where the actual methods for calculating mean, median and standard deviation will be located. You should notice that these method calls are being made by the class name not a specific instance of that class. That means that every method in the **StatFormulas** class will be static; creating a so called “static” outer class. You should also notice that there is no **StatFormulas.java** file in your directory. *****You will be implementing that class; more on that in a bit.*****

If you take a look at src/main/resources, you will find the two text files, values.txt and values2.txt that this program uses to test the methods in the StatFormulas class.

Now take a look in the src/test/java directory and look at the **StatMethodTest.java** file. This is a Junit5 test file. We will be doing unit tests with Junit5 soon but for now, you just need to know that this is where the test cases are. Notice that for each file, 5 values and 6 values, we are testing the methods for mean, median and standard deviation (std) you will write in the **StatFormulas** class you create.

Finally, take a look at the pom.xml file in the root directory, **HomeworkOne**. This contains all of the dependencies this project needs to run.

CS 342 Homework#1 Fall 2019

What you need to do:

1) In the src/main/java directory, create a new java file called StatFormulas.java. This file will have one class defined in it: **public class StatFormulas**
2) Inside of the StatFormulas class you need to implement the following methods:

```
public static double mean(double values[])
public static double median(double values[])
public static double std(double values[])
```

These are basic statistics formulas that are easily found if you are not familiar with them. You will notice that your data is read into arrays of doubles. There are better data structures in Java but since we have not covered them yet, we will stick with arrays which everyone should be comfortable with. There are some helpful methods of the Java Array class if you choose to use them. For instances; you will notice this line in the constructor of the Statistics class:

```
Arrays.parallelSort(values);
```

This is a static method of the Array class that sorts an array. You will also find **length** helpful.

How to develop this project:

You will need to import this project, the HomeworkOne directory, as a Maven project in the IDE of your choice or just go to that directory from the command line. You can run the Maven commands:

clean: removes all class files and reports.
test-compile: compiles files in the src/test/java directory
compile: compiles files in the src/main/java directory
test: runs all test cases

For each method you write, you should compile your project and run the tests. If the method is correct, you will pass the two tests for that method (run with the values from the text files).

You must also add the ability to run main in the pom.xml file. You do this by adding the following line in the properties section:

```
<exec.mainClass>HomeworkOne</exec.mainClass>
```

Run main with the following Maven command: **exec:java**

CS 342 Homework#1 Fall 2019

Electronic Submission:

Put the Maven template folder with your files in a .zip and name it with your netid + MavenJava: for example, I would have a submission called mhalle5MavenJava.zip, and submit it to the link on Blackboard course website.

Assignment Details:

Late work on a homework is **NOT ACCEPTED.** Anything past the deadline will result in a zero.

We will test all homework on the command line using Maven 3.6.1. You may develop in any IDE you chose but make sure your homework can be run on the command line using Maven commands. Any homework that does not run will result in a zero. If you are unsure about using Maven, come see your TA or Professor.

Unless stated otherwise, all work submitted for grading *must* be done individually. While we encourage you to talk to your peers and learn from them, this interaction must be superficial with regards to all work submitted for grading. This means you *cannot* work in teams, you cannot work side-by-side, you cannot submit someone else's work (partial or complete) as your own. The University's policy is available here:

https://dos.uic.edu/conductforstudents.shtml.

In particular, note that you are guilty of academic dishonesty if you extend or receive any kind of unauthorized assistance. Absolutely no transfer of program code between students is permitted (paper or electronic), and you may not solicit code from family, friends, or online forums. Other examples of academic dishonesty include emailing your program to another student, copying-pasting code from the internet, working in a group on a homework assignment, and allowing a tutor, TA, or another individual to write an answer for you. It is also considered academic dishonesty if you click someone else's iClicker with the intent of answering for that student, whether for a quiz, exam, or class participation. Academic dishonesty is unacceptable, and penalties range from a letter grade drop to expulsion from the university; cases are handled via the official student conduct process described at https://dos.uic.edu/conductforstudents.shtml.