CIS 255 • Java Programming
Laboratory Assignment 10: Arrays

In this laboratory assignment you will work with Java array syntax to read in a location's high temperature and low temperature for each month and display these values in columns with the average between the two.

Background

A local weather enthusiast has recorded his home's highest and lowest temperature for each month and would like to display this data in a table, including the month number, the highest temperature, the lowest temperature, and the average of the two on each line. Since all of the input must be maintained to generate the output in a report-like form, the program needs the capacity to store at least 24 values. Instead of using 24 individual variables, the program can use two arrays: one for the highs, and one for the lows.

Example output after the user enters the temperatures:

```
Month      High       Low        Average
-----     ------     -----       -------
   1       81.1       -6.3         37.4
   2       83.4        1.7         42.5
   3       90.0        2.3         46.2
   4       92.5       25.8         59.2
   5       98.6       35.5         67.1
   6      106.5       42.5         74.5
   7      106.5       50.5         78.5
   8      105.1       50.5         77.8
   9      105.9       36.7         71.3
  10       93.8       27.5         60.7
  11       84.9        4.7         44.8
  12       80.4        0.7         40.5
```

Step 1: Download the Starter File and Launch TextPad

On Blackboard, click the link for the starter file to download it. Make sure you insert your first initial and last name and the laboratory assignment number where indicated (**FLast_Lab10.java**) and that you specify the file location. Once you have downloaded the file, open it in TextPad (or a similar Java code editor).

Step 2: Customize Your Code and Add Import Statements

Edit the comments at the top of the code to include your name, the course number, the laboratory assignment number, and a brief description of the program's contents. You may also wish to include the date (perhaps after the laboratory assignment number). Also, modify the class name to match your file name. As you continue your code, you may go ahead and add comments to the statements or groups of statements that you believe deserve additional explanation, or you may wait until after you have written your entire class definition to add comments.

Since a report-like display lends itself more to the console than to dialogs, add an `import` statement for the `Scanner` class, and if you choose to format the values with `DecimalFormat` instead of `printf`, add that `import` statement as well.

Step 3: Declare Variables, Including Arrays

Declare the following variables at the beginning of the `main` method:

Description	Data Form	Initial Value
High temperatures	Array of 12 real numbers	From input
Low temperatures	Array of 12 real numbers	From input
Subscript variable	Whole number	0

Declare a `Scanner` object to read input from the console, and if you decide to use `DecimalFormat`, declare that object here as well (make sure at least one digit is shown before the decimal point and exactly one digit is shown after the decimal point). Finally, you may want to store the calculated average for each month, either in a single variable or in a separate array (you also have the option of calculating the average in the output statement itself, in which case no additional variable or array would be necessary). Write the declarations for these variables in your `main` method.

Even though the highest subscript in each array will be 11, the value in the second set of square brackets in each array declaration needs to be 12, as that is the number of elements in the array.

Step 4: Use a Loop to Read Input from the Console

Use a loop to prompt the user for the highest and lowest temperature for each month in a manner similar to the following (assume that the user enters values similar to the ones shown in **bold**):

```
Enter the highest temperature for month #1:  81.1
Enter the lowest temperature for month #1:  -6.3
Enter the highest temperature for month #2:  83.4
Enter the lowest temperature for month #2:  1.7
```
(Assume that the pattern continues for months 3 through 11)
```
Enter the highest temperature for month #12:  80.4
Enter the lowest temperature for month #12:  0.7
```

As you obtain each value from the user, store it in the appropriate array element.

You can process multiple arrays in a single loop. The counter variable used to access an element at a particular subscript in one array can also be used to access an element at the same subscript in other arrays. This works best when all of the arrays being processed have the same size.

The month number you display to the user should match what the user expects (1 for January, 2 for February, etc.), but the subscript where the value should be stored should match what the computer expects (0 for January, 1 for February, etc.). Account for this difference in your code.

Each loop you use to process all of the elements in an array must stay strictly less than the size of the array, not less than or equal to the size.

Step 5: Display Column Headings

The program will display the values in columns, so headings are appropriate. Display text similar to this:

```
Month      High       Low        Average
-----     ------     -----       -------
```

There are two ways to get values to line up. One is to use the escape sequence `'\t'` to move to the next tab stop. The other is to set the field width for each placeholder in the `printf` string.

Step 6: Display the Results

Use a loop to display the values for all twelve months. For each month of data, display the month number (one greater than the subscript where the values for that month are stored), the high (from the array), the low (from the array), and the average of the high and low. The program may calculate this average within the output statement, or it may store the average in a single variable or an array element. Make sure that the temperatures show one digit after the decimal point and that they line up near the column headings.

The concept of using more than one array to store multiple values for each item in a list is known as parallel arrays; the subscript links the values in the arrays together.

Step 7: Compile Your Code and Execute the Byte Code

Compile your Java program, correcting any syntax errors as the compiler finds them, and then execute the compiled application byte code. Enter some test values to verify that the results match what you believe they should be; see the **Background** section for an example set of values. To test multiple sets of input, you'll need to run the program multiple times.

Step 8: Add Inline Comments and Submit Your Code

Make sure that your code has **three or more inline comments** near the statements that you believe warrant the additional description (especially those that deal with arrays). These can be simple line comments at the end of a line of code that needs explanation, or block comments where each precedes a series of statements described in the comment.

Follow the instructions on the course website to attach and submit your source code.

Make sure you complete the submission process! If you attach your file without submitting the assignment, your file will not be received!