CS 1400 - Lab 6

Maximum Points: 10 pts.

Lab Topics

- The basics of Array object

Use the following Coding Guidelines

1) Download the template file Lab6.java from Canvas and fill-in-the-blanks to create your Java program.
2) Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
3) Keep identifiers to a reasonably short length.
4) Use uppercase for constants. Use upper camel case for classes. Use lower camel case for all other identifiers (variables, methods, objects).
5) Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with ifs, switches, and loops. Be consistent with the number of spaces or tabs that you use to indent.
6) Use white space to make your program more readable.
7) Use comments to explain how the parts of your program work.

Problem Description

For this lab, you will

1. create a basic Array of numbers,
2. fill in the elements of that array by prompting the user for inputs,
3. display the elements of the array to the user, and then
4. calculate and display the sum of those array elements.

Step 1: Getting started with the header

At the beginning of each programming assignment you must have a comment block with the following information:

```
/*------------------------------------------------------------
// AUTHOR: your name.
// FILENAME: title of the source file.
// SPECIFICATION: your own description of the program.
// FOR: CS 1400 - Lab #6
// TIME SPENT: how long it took you to complete the lab.
//-----------------------------------------------------------*/
```

Step 2: Declare the class Lab6 and a main method

In this lab you only need one class Lab6 and a main method. The program flow is included in the following sample code:

```
import java.util.Scanner;

public class Lab6 {
    public static void main(String[] args) {
        // Declaring variables
        // ...
        // Request array size from the user
        // ...
        // Declare the array
        // ...
        // Fill in the array
        // ...
        // Construct a for loop to do summation
        // ...
        // Display the result
    }
}
```

Step 3. Display the variables you need

To get input from the user, we need some variables to hold the data. Please, declare 4 variables as follows, you can choose the variable names you like:

• arraySize: int	the size of the array
• currentElement: double	the value of the current element
• sumOfElements: double	the sum of all elements in the array
• scan: Scanner	used to parse the user input

Step 4. Request the user input and declare a double array

Please use the Scanner object you defined to request an integer from the user. Store the result in the variable arraySize. To do so, you will need a println to prompt the user and scan.nextInt() method to parse an integer.

In Java, an array is a collection of variables. It reserves a sequence of memory space and collects data of the same type. For example, an integer array of size 5 can hold 5 integers in one place. In some cases, it we would like to keep relevant data together instead of using multiple separate variables.

After the array size is determined, we can declare an array of double values.

```
// Request the array size from the user
// -->

// Declare a new array of size equal to the size requested
//
// For reference, the following is an EXAMPLE declaration of an
// integer array of a fixed size. (DO NOT USE THIS CODE DIRECTLY).
//
// int[] integerArray = new int[25];
//
// -->
```

Step 5. Fill the array by the user inputs (for loop)

An array element can be accessed by using array index, which is the equivalent concept to the String index. To assign a value to our array, we may do

```
doubleArray[0] = 10.0;
```

It means we put the value 10.0 to the first element of doubleArray.

To initialize all elements in an array, we can utilize a for loop and make the loop variable iterate the available indices.

Please write a for loop with a variable goes from 0 to arraySize – 1. In each round, you need to use your Scanner object to get a new value for the array element.

Step 6. Display the array content backwards and find the sum (another loop)

To display and sum elements in an array, again we will use a for loop. Please write a for loop which can iterate all elements in your array backwards and find the sum.

Sample Output

Below is an example of what your output should look like when this lab is completed. Text in **RED** represents user input.

```
How many elements in the array?
5
Please enter the next value:
1
Please enter the next value:
2
Please enter the next value:
3
Please enter the next value:
4
Please enter the next value:
5
5.0  4.0  3.0  2.0  1.0
The sum of the array's elements is: 15.0
```

Submit Your Lab6.java to Gradescope

Please submit **ONLY** the file Lab6.java to the “Lab 6” link on Gradescope. Make sure it is compiling and producing the expecting outputs. You are done.