

## Lab 5

### What this Lab Is About:

- 1) Basic use of arrays

### Problem Description: Read a list of numbers, Print and Sum them

For this lab, you will create a basic Array of numbers, fill in the elements of that array by prompting the user for input, display the elements of the array back to the user, and then calculate and display the sum of those array elements.

#### Step 1: Getting Started

Create a class called **Lab5**. Use the same setup for setting up your class and main method as you did in previous labs and assignments. Be sure to name your file **Lab5.java**. Do not use a package.

#### Step 2: Declare Variables

For this section of the lab, you will need to declare four local variables: one integer for the length of the array, two doubles for the current array element and sum of array elements, and one Scanner to read input from the user. Declare these variable immediately after the function definition. You will declare the array in Step 4 after you know how big it should be.

#### Step 3: Request Array Size from User

Next, use the Scanner object declared in the previous step to request from the user the number of elements the array should have. Remember that you should always prompt the user with a message (e.g. "How many elements in the array? ").

#### Step 4: Declare the Array

Next, use the integer value previously requested from the user to declare a new array of this length. The array should be type double.

```
// Declare a new array of size equal to the size requested in
// Step 3.
-->

// For reference, the following is an EXAMPLE declaration of an
// integer array of a fixed size.
// DO NOT USE THIS ARRAY. Yours will be an array of type double
```

```
// int[] integerArray = new int[25];
```

### **Step 5: Fill in the Array**

Using a for loop, we will now fill in the elements of the array. This loop will iterate as many times as we have elements in the array. Notice that this loop starts at 0 and its final iteration occur when i is equal to the number of elements - 1. This is important because arrays in Java are zero-indexed, which means that the first element in the array is stored at the 0<sup>th</sup> position in the array. Inside the for loop, prompt the user for the next element and store that value in the appropriate position of the array.

```
for(int i = 0; i < <numberofelements>; i++) {  
    // Display the message: "Please enter the next value: "  
    -->  
  
    // Request the next element (double) from the user using  
    // the Scanner object declared in Step 2.  
    -->  
  
    // Store this element at the ith position of the array  
    -->  
}
```

### **Step 6: Display and Sum the Array's Elements**

Again using a for loop, we will now display back to the user the elements of the array in reverse order of their input (format the output to three places after the decimal point) and sum up the array's values. The array's elements should be printed with 8 values on each line with a tab between each element, and a single newline separating each line. Again, the construction of the for loop must respect that arrays in Java are zero-indexed, so the last element of the array is stored at the n-1<sup>th</sup> position, where n is the length of the array.

```
// Construct a for loop that runs backwards through the array,  
// starting at the last element and ending at the first.  
for (/* Loop logic */) {  
    // Inside this for loop, print the ith element of the array  
    // and a tab, with NO newline characters.  
    -->  
  
    // If this element is the 8th one on its line, print a  
    // newline character to advance to the next line.  
    // Also inside this for loop, add the value of the ith  
    // element to the current value of the double for the sum  
    // of elements declared in Step 2.  
    -->  
}
```

### Step 7: Display the Sum

Lastly, print out the sum of the array's elements to the user and close the main function and Lab5 class. Format the output of the double so that it displays three places after the decimal point. Note that your print statement should include a preceding newline character to ensure that this message appears on a different line than the list of elements. Also note that as usual, <sum> indicates that the message should show the value of the variable used to store the sum of elements, not the word "sum".

```
// Print the following message to the user, preceded by a
// newline character.
// "The sum of the array's elements is: <sum>"
-->
```

See the sample output below and make sure to display in the same format.

Below is an example of what your output should roughly look like when this lab is completed. All text in **green** represents user input.

#### Sample Run 1:

```
How many elements in the array? 10
Please enter the next value: 0.11
Please enter the next value: 0.10
Please enter the next value: 0.9
Please enter the next value: 0.8
Please enter the next value: 0.7
Please enter the next value: 0.6
Please enter the next value: 0.5
Please enter the next value: 0.4
Please enter the next value: 0.3
Please enter the next value: 0.2
0.200 0.300 0.400 0.500 0.600 0.700 0.800 0.900
0.100 0.110
The sum of the array's elements is: 4.610
```

#### Sample Run 2:

```
How many elements in the array? 3
Please enter the next value: 3.14
Please enter the next value: 1.414
Please enter the next value: 2.718
2.718 1.414 3.140
The sum of the array's elements is: 7.272
```

### Step 8: Submit your Lab

Once your program is working correctly, submit your lab on Canvas. Only upload the files Lab5.java.