

Overview

This week we are going to learn how to use array to hold multiple data items without having to declare a variable for each item.

Array Syntax

The general form for array *declaration* is:

```
<type>[] <var>;
```

Examples:

```
int[] intArr;  
double[] doubleArr;  
String[] stringArr;
```

The general form of array *construction* is:

```
<var> = new <type>[<int_expression>;
```

Here **int_expression** is any expression that evaluates to an integer. For example, **int_expression** can be an integer literal (like, 5), or a literal expression that evaluates to an integer (like, (8*15)%16), or an integer expression (like, a/3 + b, where a and b are integer variables). Examples:

```
int max = 20;  
intArr = new int[100];  
doubleArr = new double[max];  
stringArr = new String[max+100];
```

The general forms of array *initialization* are:

```
<type>[] <var> = new <type>[<int_expression>;  
<type>[] <var> = {<literal1>, <literal2>, ..., <literalN>;
```

where <literal1>, <literal2>, etc. are literals of the data type <type>. Examples:

```
int[] intArr = new int[100];  
String[] stringArr = {"Alpha", "Beta", "Gamma"};  
= new String[max+100];
```

To access each element of an array, state the array reference variable followed by the element's position inside square brackets. For example:

```
0th entry of intArr is intArr[0]  
1st entry of intArr is intArr[1]  
...  
Nth entry of intArr is intArr[n]
```

Each array element can be both read and written to just like any variable. For example:

```
intArr[5] = 27;           // Writing to an array entry  
int n = intArr[5] - 7;    // Reading an array entry
```

Getting Started

After starting Eclipse, create a new project called **Lab20_11**. Import the **SumAllArr.java** from the Lab 11 assignment page into the project and run it.

Part 1: Create – SumSquareArr.java

This program should ask the user for a number, **max**, create an array of size **max** and populate each element at index **i** with the result of the expression $1^2 + 2^2 + \dots + i^2$. The program must repeat the calculations

using 3 types of loops and populate 3 different arrays (as shown in **SumAllArr**). It should then output the contents of each array, along with the sum of all the square numbers as shown below (inputs are shown in **green**, with each run separated by a dashed-line):

```
Please enter the maximum number: 5
```

```
Contents of Arr1 (for-loop), Arr2 (while-loop), Arr3 (do-while-loop) are:
```

```
Arr1 0, Arr2 0, Arr3 0
```

```
Arr1 1, Arr2 1, Arr3 1
```

```
Arr1 5, Arr2 5, Arr3 5
```

```
Arr1 14, Arr2 14, Arr3 14
```

```
Arr1 30, Arr2 30, Arr3 30
```

```
Arr1 55, Arr2 55, Arr3 55
```

```
Sum of Squares of numbers from 1 to 5 is: 55
```

```
-----  
Please enter the maximum number: 0
```

```
No numbers to sum up. Exiting program.
```

```
-----  
Please enter the maximum number: 10
```

```
Contents of Arr1 (for-loop), Arr2 (while-loop) and Arr3 (do-while-loop) are:
```

```
Arr1 0, Arr2 0, Arr3 0
```

```
Arr1 1, Arr2 1, Arr3 1
```

```
Arr1 5, Arr2 5, Arr3 5
```

```
Arr1 14, Arr2 14, Arr3 14
```

```
Arr1 30, Arr2 30, Arr3 30
```

```
Arr1 55, Arr2 55, Arr3 55
```

```
Arr1 91, Arr2 91, Arr3 91
```

```
Arr1 140, Arr2 140, Arr3 140
```

```
Arr1 204, Arr2 204, Arr3 204
```

```
Arr1 285, Arr2 285, Arr3 285
```

```
Arr1 385, Arr2 385, Arr3 385
```

```
Sum of squares of numbers from 1 to 10 is: 385
```

Part 2: Create – AnyAverageArr.java

This program should ask the user for a number, **max**, a column width, **col** and do the following:

- Create an array of size **max**.
- Ask user for **max** numbers and store each number as successive array elements.
- Print the numbers entered with **col** elements in each line.
- Output the average of all the numbers.

Your program must produce an output as follows (inputs are shown in **green**, with each run separated by a dashed-line):

```
This program will find the average of any set of numbers.
```

```
Please choose number of values to average: 10
```

```
Please choose column width for output formatting: 4
```

```
Please enter value #1: 1
```

```
Please enter value #2: 2
```

```
Please enter value #3: 3
```

```
Please enter value #4: 4
```

```
Please enter value #5: 5
```

```
Please enter value #6: 6
```

```
Please enter value #7: 7
```

```
Please enter value #8: 8
```

Please enter value #9: 9
Please enter value #10: 10

The numbers being averaged are ...
1 2 3 4
5 6 7 8
9 10

Average is: 5.5

This program will find the average of any set of numbers.

Please choose number of values to average: 0

No numbers to average. Exiting program.

This program will find the average of any set of numbers.

Please choose number of values to average: 8
Please choose column width for output formatting: 5

Please enter value #1: 324
Please enter value #2: 23
Please enter value #3: 34
Please enter value #4: 45
Please enter value #5: 65
Please enter value #6: 53
Please enter value #7: 24
Please enter value #8: 63

The numbers being averaged are ...
324 23 34 45 65
53 24 63

Average is: 78.875

Part 3: (Assessment) Logic Check

Create a Word document or text file named **Part3** that contains answers to the following:

1. Consider the declaration in **SumAllArr.java**: `int[] arr1 = new int[max + 1];`
 - a. How many array elements does this statement create?
 - b. Why do we have `max + 1`?
 - c. Can we just use `max` and have the program still work correctly?
2. Before the while-loop in **SumAllArr.java** we have `i = 1`.
 - a. Why is it there?
 - b. Can we use another variable instead?
3. The do-while loop in **SumAllArr.java** is implemented using `(++i <= max)`.
 - a. Provide an alternative expression that implements the same logic.
 - b. What will `(i++ <= max)` result in?

What to hand in

When you are done with this lab assignment, submit all your work through CatCourses.

Before you submit, make sure you have done the following:

- Attached the file named **Part3** containing answers to the assessment questions.
- Attached the created **SumSquareArr.java** and **AnyAverageArr.java** files.
- Filled in your collaborator's name (if any) in the "Comments..." text-box at the submission page.

Also, remember to demonstrate your code to the TA or instructor before the end of the grace period.