

CST8132 Object Oriented Programming

Lab 2: Array Exercise

Due: Start of Lab Sessions in Week 3

Problem Statement:

To practice deciphering and following instructions and working with arrays in Java and to become more comfortable working with array indices and arrays of arrays, as well as working with classes and objects.

Recommended Reading:

- Chapter 7 of Deitel and Deitel, Java How to Program
- <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html>
- Advanced Reading: <http://introcs.cs.princeton.edu/java/14array/>

Exercise One:

Write a program called **ExerciseOne.java** according to the following points:

1. The **ExerciseOne** class of objects has an instance variable named "**myArray**" which is a reference to an array of primitive integers.
2. The default **constructor** for the **ExerciseOne** class instantiates an array of ten integers, assigns it to **myArray**, and initializes the elements of the array with the values 1 through 10, using a **for-loop** (You may NOT keep a running counter of values to put in the array. You must assign the values in terms of the for-loop control variable, for example, i.)
3. The class has a method called "**printArrayValues()**" that uses a **for-loop** and **System.out.print()** statement to print out the elements of the array, as shown in the sample output.
4. The class has a method called "**displayArrayProduct()**" that uses an **ENHANCED for-loop** to calculate the product of elements in the array, then print out an English sentence stating the product.
5. The **main** method should create an **ExerciseOne** object, and then call its "**printArrayValues()**" and "**displayArrayProduct()**" methods. At this point your program will print out the lines of the sample output below.

Exercise One: Sample Output

```
myArray = {1,2,3,4,5,6,7,8,9,10};  
The product of all elements of myArray is 3628800.
```

Exercise Two:

This Exercise is a bit more difficult, but if you spend some time on it, you can make it.

Write a program called **ExerciseTwo.java** to do the same thing as ExerciseOne, except this time with two dimensions instead of one, according to the following points:

1. Instead of an array of 10 elements, the array will be an 8 x 10 array of arrays of integers (that is, **a two-dimensional array of integers**). You can think of this as a two-dimensional array with 8 rows and 10 columns, or you can think of it as an array of 8 elements, where each of those 8 elements is itself an array of 10 integers.
2. In the default **constructor**, use **nested for-loops** to initialize this array of arrays with values from 0 to 79, such that the first row has the values 0,1,2,3,4,5,6,7,8,9 and the last row has values 70,71,72,73,74,75,76,77,78,79. (You may NOT keep a running counter of values to put in the array. You must **assign the values in terms of the for-loop control variables** – the ones that are perhaps named i and j.)
3. The class has a method called "**printArrayValues()**" that uses **nested for-loops** and **System.out.print()** statement to print out the elements of the array, as shown in the sample output.
4. Write a method called "**getArrayAverage(int[] a)**" which **takes an array of primitive integers as a parameter**, and **returns a double** that is the average value of all of the elements of the array. The method uses an **ENHANCED for-loop** to calculate the average.
6. The class has a method called "**displayArraySumOfAverages()**" which uses an **ENHANCED for-loop** to iterator through the "rows" of the 2-dimensional array, calling the method "**getArrayAverage**" for each inner array, and adding the returned average to the total sum of all of the averages. The method then prints out an English sentence stating the total sum.
5. Add a **nested set of ENHANCED for-loops** that calculates the total sum of all elements of the 2-dimensional array, so your program can print out an English sentence stating the total sum.
6. The **main** method should create an **ExerciseTwo** object, and then call its "**printArrayValues()**" and "**displayArraySumOfAverages()**" methods. At this point your program will print out the lines of the sample output below.

Exercise Two: Sample Output

```
myArray = {
    {0,1,2,3,4,5,6,7,8,9},
    {10,11,12,13,14,15,16,17,18,19},
    {20,21,22,23,24,25,26,27,28,29},
    {30,31,32,33,34,35,36,37,38,39},
    {40,41,42,43,44,45,46,47,48,49},
    {50,51,52,53,54,55,56,57,58,59},
    {60,61,62,63,64,65,66,67,68,69},
    {70,71,72,73,74,75,76,77,78,79}
};
The sum of the average value of each array in myArray is 316.0.
```

Submission Requirements

1. Create a new assignment folder and name it:
CST8132_<SectionNo>_<LastName>_<FirstName>_Lab2.
2. Place copies of your **ExerciseOne.java** and **ExerciseTwo.java** files in the assignment folder.
3. Create an **ZIP** file of the assignment, and name it:
CST8132_<SectionNo>_<LastName>_<FirstName>_Lab2.zip.
4. Upload the **ZIP** file to Blackboard.

Grading Scheme:

- 5 marks: Correctness
 - Array element values are correct
 - Array output format is correct
 - Exercise 1 sum is correct.
 - Exercise 2 sum is correct.
 - For loops are used where required.
 - Enhanced for loops are used where required.
 - Classes are implemented as described.
 - Methods are implemented as described.
- 2 marks: Code Conventions and Submission
 - Proper naming (use of Java naming conventions and following requirements)
 - Use of comments in code
 - Consistent and proper indentation and code alignment
 - Good use of both vertical and horizontal whitespace for readability
 - Submission correct
- -5 marks: Code does not compile

Tips:

- Use an “if” statement inside of your loop outputting the array element values to optionally print the final comma.
- Array element values **MUST** be assigned as a formula based on the for-loop control variables.
- Each exercise is a separate program, independent of one another, and each requiring its own main method.

Questions for Reflection

- Why were enhanced for loops not used to initialize the arrays? What happens if I try to initialize an array with an enhanced for loop?
- Why were classic for loops used in places?
- What are advantages / disadvantages to each for loop?