Lab 2 Array Exercise

CST8132 Fall 2017

Due:

Check for the due date on Blackboard

Marks:

out of 10 marks: 8 marks for demonstration, 2 marks for readability.

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/

Purpose:

To practice deciphering and following instructions, inheritance, 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.

Exercise 1 (demonstration required, 3 marks):

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

1. The ExerciseOne class shall have an instance variable named "myArray" which must be an array of (primitive) integers.
2. The constructor for the ExerciseOne class takes one integer argument (i.e. ExerciseOne(int d)), instantiates an array of ten integers, assigns it to myArray, and initializes the elements of the array with the values 0, d, 2d, 3d... to 9d (i.e. 10 elements), 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 index variable, for example, i.)
3. The class has a method called "printArrayStatement()" that uses a for-loop and System.out.print() and/or System.out.println() statements to print out the Java code statement that would declare, create, and initialize the elements of the array all at once, naming the array "outputArray" (see the first line of Sample Output below)
4. The class has a method called "displayArrayTotal()" that uses an ENHANCED for-loop to calculate the total sum of elements in the array, then print out an English sentence stating the total sum.
5. The main method should create an ExerciseOne object, and then call its "printArrayStatement()" and "displayArrayTotal()" methods. I.e:

```
ExerciseOne e1 = new ExerciseOne(1);
e1.printArrayStatement();
```

```
    e1.displayArrayTotal();
```

At this point your program will print out the first two lines of the sample output below.

Required Output with d==1;

```
 int[] outputArray = {0,1,2,3,4,5,6,7,8,9};
 The sum total of all elements of myArray is 45
```

Exercise 2 (demonstration required, 5 marks):

This Exercise is more difficult, so be prepared to spend some time working it out. This one is not supposed to be easy, but trying it yourself for a while and then asking for help if you get stuck is a good way to increase your skill at this kind of programming.

Write a program called ExerciseTwo.java that inherits ExerciseOne that also does with two dimensions instead of one, according to the following points:

1. ExerciseTwo class shall have an array of 8x10 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. Use nested for-loops to initialize this array of arrays, such that the first row has the values 0, d, 2d, 3d,…9d and the last row has values 70d,…79d. (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 index variables – the ones that are perhaps named i and j.)
3. Similarly to ExerciseOne, create a new printArrayStatement(int i) (Note different parameter signature for method overloading – the actual parameter int i will not be used) use a nested for-loop to print out the Java code statement that would declare, create, and initialize this array all in one step.
4. Create a new displayTotal(int i) (Again, overload a method with different parameter signature for method overloading – the actual parameter int i will not be used) using 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.
5. The main method should call an ExerciseTwo object and call with following methods (note, first two methods should be using superclass ExerciseOne's methods):

```
ExerciseTwo e2 = new ExerciseTwo(1);
e2.printArrayStatement();
e2.displayArrayTotal();
e2.printArrayStatement(2);
e2.displayArrayTotal(2);
```

Required Output with d==1;

```
 int[] outputArray = {0,1,2,3,4,5,6,7,8,9};
 The sum total of all elements of myArray is 45

int[][] outputArray = {
    {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 total sum of the elements in myArray1[][] is 3160
```

6. You MUST use ExerciseOne’s methods (i.e. printArrayStatement(), and displayArrayTotal()) – not create (nor copy) new methods in ExerciseTwo. This is key learning point of inheritance.