CSC 1051 - Lab 13

Objectives:
This lab begins our study of arrays.

a) Simple array example

```java
class ArrayDemo {
    public static void main(String[] args) {
        int[] anArray; // declares an array of integers
        anArray = new int[10]; // allocates memory for 10 integers
        anArray[0] = 100; // initialize first element
        anArray[1] = 200; // initialize second element
        anArray[2] = 300; // etc.
        anArray[3] = 400;
        anArray[4] = 500;
        anArray[5] = 600;
        anArray[6] = 700;
        anArray[7] = 800;
        anArray[8] = 900;
        anArray[9] = 1000;
        System.out.println("Element at index 0: " + anArray[0]);
        System.out.println("Element at index 1: " + anArray[1]);
        System.out.println("Element at index 2: " + anArray[2]);
        System.out.println("Element at index 3: " + anArray[3]);
        System.out.println("Element at index 4: " + anArray[4]);
        System.out.println("Element at index 5: " + anArray[5]);
        System.out.println("Element at index 6: " + anArray[6]);
        System.out.println("Element at index 7: " + anArray[7]);
        System.out.println("Element at index 8: " + anArray[8]);
        System.out.println("Element at index 9: " + anArray[9]);
    }
}
```

• Type in this program, compile, and run.
• Rename the program Lab13a.java and modify it to set the array values using a loop. (Hint: Use two for loops, one for initializing the values and another for printing them out; use the loop counter to index into the array elements.)
• Modify the program to work with an array of 100 elements, instead of 10.

b) Arrays of different types

This example in part (a) is from the java online tutorials:
[http://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html](http://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html)
Read the tutorial on arrays. Note that this is one of the many tutorials available online on Java programming.

Create a new version of your program from part (a) and name it **Lab13b.java** that creates instead an array of 100 values of type `double`, set to random values in the range 0....1.

c) An array of booleans
Create a new version of your program **Lab13c.java** that creates instead an array of 100 values of type `boolean`. The values should be set to alternating true/false, i.e., `anArray[0] = true, anArray[1] = false, etc. (but be sure to use a loop here too).

d) Reading values into an array
Starting from Lab13a.java (array of int) create a new version of your program **Lab13d.java**, that uses Scanner to inputs integer values from the user. (Test it with smaller arrays, say with 5 entries, so you don't need to type so much.) Once you have it working, enlarge it again to 100 integers and modify it so that the inputs the values from a file. Use this file for input:

www.csc.villanova.edu/~map/1051/examples/oneHundredInts.inp

e) Using an array to keep a count
Create a new program **Lab13e.java** based on the previous program.

Instead of printing out all the elements in the array, this program should:

1. Create the array `anArray` as in Lab13d.java (100 random integers).
2. Create a second array, an array of `int` named `Counter` and allocate space for 10 values of type `int`.
3. Initialize all values in `Counter` to 0.
4. Loop through all the elements in `anArray[0].. anArray[99]` counting how many 0s, how many 1s, etc. For example, if the value of `anArray[i]` is 8, then `Counter[8]` should be incremented.
5. Print out the values stored in `Counter`, labeled appropriately.