**IT 1090C Computer Programming I   
Lab 9: Arrays (and more Looping)  
SPRING 2020  
20pts 5 Grad or EC pts**

Objective:

To thoroughly explore Java’s array and arrayList collections.   
Learn about random numbers in java.  
  
**Mini-lecture:**

We use java.util.Random() to generate random numbers. We have to include the import:

**import java.util.Random;**

Create a random object:

**Random rnd = new Random();**

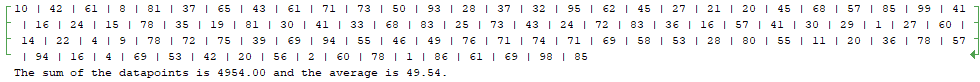
**int val = rnd.nextInt(100); // gives a value from 0 to 99**

**int val = rnd.nextInt(100) + 1; // gives a value from 1 to 100**

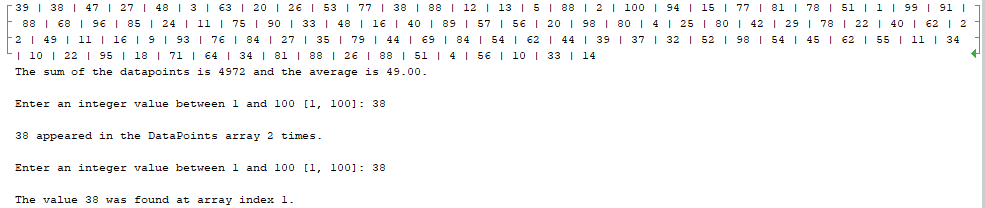
**int index = rnd.nextInt(myArray.length); gives a random valid index on myArray**

**Begin the lab:**  
  
Create a netbeans project called ArrayStuff with a single java main class also called ArrayStuff. All the code for the lab you will do today is in this single main class. Implement each problem in the main class and test it carefully as you go. In particular, watch out for off by one errors! Use your SafeInput library methods for input.

1. (10pts)
   1. Declare an array of type int named *dataPoints*.   
      *dataPoints* should have a length of 100. (i.e. it should hold 100 int values.)
   2. Now, code a regular for loop that iterates through the *dataPoints* array and initializes each element in it to a **random** value between 1 and 100.   
      (See notes above for using java.util.Random. Please do not use Math.Random!)
   3. Code a second loop that displays the *dataPoints* values like this (values are all on the same line and separated by “ | “ e.g. space, vertical bar, space):  
      val1 | val2 | val3 | HINT: use printf or System.out.print()
   4. Now code a loop that calculates the sum and the average of the values in *dataPoints.* Add code to output/display the sum and the calculated average. (Be sure to include some sort of description in the output so it is clear (i.e. use complete sentences) : ‘The average of the random array dataPoints is: ‘ or something similar. Don’t just print out the average and sum as a raw numbers. (Do this for all the output here… just like you should in every program you code!)  
        
      Paste the screen snip or copy of your output window here:   
      (do the rest the same way…)



1. Linear scan (search): (10pts)
   1. Add code to prompt and input an int value between 1 and 100 from the user. (Use your static getRangedInt(…) method from SafeInput from the last lab here! Just include your SafeInput.java Library. file in your Netbeans project and access your bullet-proofed input methods there. **SafeInput.getRangedInt(…)**
   2. Code a loop that iterates through the entire *dataPoints* array and counts how many times the user’s value is found within the array. Indicate with a clear output statement how many times the loop found the user’s value within the array.  
        
      Paste the screen shot of the output here:   
      
   3. Prompt the user again for a val between 1 and 100.   
      Code a loop that iterates through the *dataPoints* array checking each value to see if it matches the one the user input. This loop should short circuit (break) when it finds the value and display the first position of the value within the array. Again, make the output clear and explicit here. (e.g. “The value VALUE was found at array index POSITION” where VALUE is the user’s value you input and POSITION is the location (index) where the value is found in the array. If the value is not in the array, then the output statement should indicate that. Note you can use the **break** keyword to jump out of a loop before it completes. The problem here is that you have to keep track of whether or not the value was found at all.  
        
      Paste the output from running your code here:   
      

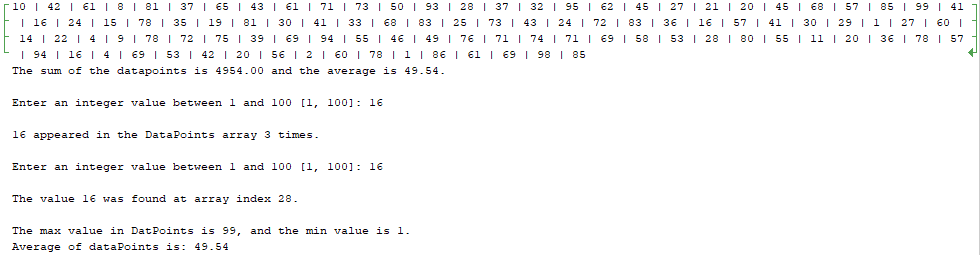


* 1. Write a loop to scan for the minimum and maximum values in the *dataPoints* array. Display these values to the user. (Note that these might not be unique, there might be several occurrences of either the min or the max value in the random array.) Hint: you can assume initially that the first element is the min and max and iterate through the rest of the array to determine if there is a lower or higher value which then becomes the min or max respectively. You can also write a single loop that determines both the min and the max at the same time.  
        
     Paste the output from running your code here:

****

* 1. Write this static method which takes an array of double values and returns the average and test it. Remember static methods go in the main java class file immediately after the main method.  
       
     **public static double getAverage(int values[])  
     {**

**}**

In your main code call it like this:  
 **System.out.printLn(“Average of dataPoints is: “ + getAverage(dataPoints));**  
  
Paste the output from running your code here: 

1. **Extra and Graduate Credit 5pts  
     
   Refactor you program and create the following static array methods. Recode your main method so you can compare the values returned by your inline code with the equivalent methods:  
     
   public static int min(int values[])** Returns the min value found **public static int max(int values[])** Returns the max value found

**public static int occuranceScan(int values[], int target)**   
 Returns the number of times target is found in the values array  
**public static int sum(int values[])** Returns the sum of the values array elements

**public boolean contains(int values[], int target)** Returns true if the values array contains target

**Submitting your work  
READ CAREFULLY:**

1. Save this word file as **Lastname\_Firstname Lab\_09.docx** Combine **this word file** and **your complete Netbeans project folder** into a **single** zip archive file and name it **Lastname\_Firstname\_Lab09.zip**.

Submit 2 things:

Submit your file archive and separately submit the docx file using the Canvas Assignment link.   
  
Just resubmit the entire assignment again exactly the same way using the extra credit link if you did the extra or graduate credit.