

# Lab Report – Week 2 - Processing 1000 Random Integers

John Marshall  
CSCI112 Spring 2023

---

## Assignment Analysis and Design

*In your own words, describe the problem including input and output. Briefly describe how you developed your code. Briefly describe what your code does and how it works – including anything different or unique or special that you did in your software. If the software is long or complicated, describe how it is organized. Include a copy of any pseudocode or other design documents you used. If you worked with anyone else, asked anyone for help, or looked anything up, then mention it here. Include proper references to source material.*

The input and output were both generated by the application so need to obtain data from a user/file. It was simply an array of randomly generated numbers. The output was two fold, to log the array to the console in both sorted and unsorted forms and a text document with the sums and average. The design was to keep the code modular by creating a sort method and a calculate method outside of the main class. The biggest aspect I spent my time researching was bubbleSort and how to optimize code. When I read about bubblesort in this week's reading I instantly thought of how wasteful it is to iterate through numbers that have already bubbled to the top, so why waste time comparing them.

John Marshall Week 2 Lab: Processing 1000 Random Integers  
CS112

```
Main{
    Initialize Int[] = randomNums [1000]

    For (Length of randomArray){
        Array[i] = int(Math.Random*10) +1
    }

    Print randomNums

    BubbleSort randomNums

    Total = calculated Sum
    Aver = calculated Average

    Print sorted randomNums

    Calculate Sum
```

IO file x = John Marshall Week 2 Lab

```
Y = printWriter
Y print John Marshall Week 2 Lab: Processing 1000 Random Integers
Blank Line
Y Print Total
Blank Line
Y print Average
Close Stream
}
```

```
BubbleSort( array ){
    Is sorted? = false
    Y = number of times run to subtract from loop passes (end number is known to be the
    highest No need to Compare)
    Int i;

    while(isSorted == false){
        isSorted = true

        For (i, length of Array- y, i++)
            (if array[i -1] > array[i]){
                Swap array[i -1] & array[i]
            }

    }
}
```

```
Calculate sum(array){
    Int sum;

    For (i, length of Array- y, i++)
        Sum = sum + array[i]

    }
}
```

---

## Assignment Code

*Include the code for your assignment unless otherwise directed by the assignment or by your instructor, which will be a copy of your Python project submitted with the report. You can put the report and the Python project all in one submission. In the report, either tell the reader that it is attached file or include the code.*

/\* John Marshall Week 2 Lab: Processing 1000 Random Integers

CSCI112

Last Edited 1/30/2023

\*/

```
public class Main {
    public static void main(String[] args) throws Exception{
        int[] randomNums = new int[1000];
        int i;
        int total;
        double average;

        //Generate Array
        for(i = 0; i < randomNums.length; i++){
            randomNums[i] = 1 + (int)(Math.random()*10);
        }
        // calculate total & average. Narrow cast to double
        total = sum(randomNums);
        average = (double)total / randomNums.length;

        // print Array
        printArray(randomNums);

        //Sort Array
        bubbleSort(randomNums);

        // print Array after sorting
        printArray(randomNums);

        // create File with Name John_Marshall_Lab_week_2.txt
        java.io.File x = new java.io.File("John_Marshall_Lab_week_2.txt");
        java.io.PrintWriter y = new java.io.PrintWriter(x);
        y.print("John Marshall week 2 Lab Processing 1000 Random Integers" );
        y.println();
        // print Sum
        y.print("The sum of all 1000 Nunber: " + total);
        y.println();
        // print Average
        y.print("The average is: " + average);
        y.close();
    }
}
```

```
// end main
```

```
//Bubble Sort
```

```
private static void bubbleSort(int[] numList){  
    int y = 0; //Optimization Variable  
    int i;  
    int c; // catalyst var  
    boolean inOrder = false;
```

```
    while (!inOrder){  
        inOrder = true;
```

```
        //subtract the optimization var y so it doesn't iterate through already sorted  
        numbers
```

```
        for (i = 1; i < numList.length - y; i++){  
            if (numList[i-1] > numList[i]){
```

```
                //swap numbers if x-1 is bigger then x  
                c = numList[i];  
                numList[i] = numList[i-1];  
                numList[i-1] = c;  
                inOrder = false;
```

```
            }  
        }
```

```
        //add to optimization variable  
        y ++;  
    }
```

```
// end bubbleSort()
```

```
//simple for loop to calculate sum
```

```
private static int sum(int[] numList){  
    int sum =1;  
    int i;
```

```
    for(i = 0; i < numList.length; i++){  
        sum = sum + numList[i];  
    }
```

```
    return sum;
```

```
    } // end Sum()

    //simply prints array in rows of 10 numbers

    private static void printArray(int[] numList){
        int i;

        for (i = 1; i < numList.length; i++) {
            System.out.print(numList[i] + " ");
            if(i % 9 == 0 && i > 0){
                System.out.println();
            }
        }

    } // end print Array()

}
```

---

## Assignment Testing

*Describe how you tested this program to verify that it runs correctly. Assignment Evaluation*

*Briefly describe what you learned from this project. What things did you struggle with? What was easy? Give your opinions of the process, including what you liked about the project and any suggestions you have for improving the project.*

Testing the program was easy enough as was a simple log to the console and output file. One thing I did do that I hadn't planned for is formatting my console logged array. I originally tested my code with 10 numbers in the array as it was a more comprehensive number, but when testing 1000 I had to create a method that would print it into something that was manageable to scroll through.