CSC110AA and CIS163AA
Ch 10 Program 1 – StatsArray with Exceptions

Please adhere to the *Standards for Programming Assignments* and the *Java Coding Guidelines.*
Submit the following :

- List of source code with comments to document
- **Test output listed as comments at the end of your program**

1.Statistics Array with Exceptions Class. (10 points) Using the StatsArray class provided (or your own), modify the class based on the UML below. You will be adding 3 new methods to the StatsArray class.

StatsArray
-size : int //the size of the array -stats : int[ ] //an int array named stats
+fillArray() : void //fill the int array with random data in the range 0 – 100 **+fillArrayFromUser( ) : void //**New** prompt the user to input values.** **// See additional information below UML.** **+checkIfNegative(someValue : int) : void //**New** throws IllegalArgumentException** **// if someValue is negative.** **// See additional information below UML.** +display(Graphics g) : void //display the contents of the array on a GUI **+displayOut( ) : void //**New** display the contents of the array using System.out.** **// See additional information below UML.** +getMax() : int //find and return largest value - max +getMin() : int //find and return smallest value - min +getSum() : int //find and return sum of all values in the array +getAverage() : double //find and return the average of all values in the array +countValues(int lowRange, int highRange) : int //count how many numbers are >= lowRange and <= highRange +isValueFound(int someNumber): boolean // if the array contains someNumber, return true otherwise return false +sortArray() : void //sort the array

Additional Information about StatsArray implementation:

- **fillArrayFromUser** method.
  - Using a do..while loop, a try block and two catch blocks, do the following:
    - Prompt the user for a value. Your prompt should look like the example output. It should indicate to the user which element they are inputting a value into.
    - Using Scanner, **read each value input by the user as a String.**
    - Convert the value to an integer using Integer.parseInt( ).
    - Invoke the checkIfNegative method to see if the number is negative.
    - With a catch statement, handle NumberFormatException (an int was not entered) if it is thrown.
    - With another catch statement, handle IllegalArgumentException if it is thrown.
    - Assign the good value to the next element in the array

- **checkIfNegative** method. Throw an IllegalArgumentException if the value passed in is negative. Throw statement might look like this:

```
throw new IllegalArgumentException( );
```

- **displayOut** method. Using a for loop, display all of the elements of the array using System.out.println. It should look like the output in the example output below.

2.StatsArrayTester.java (10 points) Implement a tester class called StatsArrayTester to test your modified StatsArray class. Specifically, StatsArrayTester should do the following in this order. See the output for further clarification:

- Create a StatsArray object called **values**.
- The **values** object will fill its array with values from the user by invoking the fillArrayFromUser method.
- The **values** object will display the current contents of its array by invoking the displayOut method.
- Calculate and display the sum of all of the elements in **values'** array using the getSum method.
- Calculate and display the average of all of the elements in **values'** array using the getAverage method.
- Calculate and display the maximum of all of the elements in **values'** array using getMax method.
- Calculate and display the minimum of all the elements in the **values'** array using the getMin method.
- The **values** object will sort its array by invoking the sortArray method.
- The **values** object will display the current contents of its array by invoking the displayOut method.

Example output: (please show your code handling exceptions)

```
Welcome to our StatsArray

Enter value [0] : 10
Enter value [1] : 11.1
Invalid value.  Enter an int.
Enter value [1] : 11
Enter value [2] : 20
Enter value [3] : hello
Invalid value.  Enter an int.
Enter value [3] : 30
Enter value [4] : 44
Enter value [5] : 5
Enter value [6] : -66
Negative value.  Not allowed.
Enter value [6] : 6
Enter value [7] : 777
Enter value [8] : 81
Enter value [9] : 19
```

```
Current Array values
----------------------------

    Stats [0] = 10
    Stats [1] = 11
    Stats [2] = 20
    Stats [3] = 30
    Stats [4] = 44
    Stats [5] = 5
    Stats [6] = 6
    Stats [7] = 777
    Stats [8] = 81
    Stats [9] = 19

Sum :        1003
Average :    100.3
Max value:  777
Min value:  5

Sorted Now
Current Array values
----------------------------

    Stats [0] = 5
    Stats [1] = 6
    Stats [2] = 10
    Stats [3] = 11
    Stats [4] = 19
    Stats [5] = 20
    Stats [6] = 30
    Stats [7] = 44
    Stats [8] = 81
    Stats [9] = 777

Goodbye
```