Java: Classes beyond “HelloWorld” :

Christian Gonzaga

NOTE package up programming assignment Classes under: **edu.cuny.csi.csc330.lab2**

1. Read Textbook **Chapter 3** - “OO Using Java Classes (Roughly pp 94 – 156). Complete the following exercises (24 pts):

Short answers / multiple-choice questions – p160:

* + - 1. Import

2 True.

3.True

4. new

6. class Method

11. False,

1. Programming assignment (76 pts):

The Lab 1 development assignment was largely an exercise in completing an already started implementation. The Lab 2 development assignment will call on you to implement a program from scratch. It’s an exercise in learning more about Java basics, naming conventions and Class/ method-level **modularity**.

Implement a runnable Class called “NumericAnalyzer”. Here’s the functional behavior that must be implemented.

* NumericAnalyzer will accept a list of 1 or more numbers as command line arguments. The list does not need to be sorted.
* Error checking: if the user fails to pass in parameters, the program will display an error message (of your choice) and exit early.
* The main() method String [] args argument values must be assigned to a numeric array.
* Your program will display the following information about the numbers provided by the user:
  + **This list of numbers provided by the user sorted ascendingly.**
  + **The average or mean value.**
  + **The median - the middle value of the set of numbers sorted. (Algorithm: index = the length of the array divided by 2; if index > 0 and is even, decrement it by 1. array[index] is your median).**
  + **The min value.**
  + **The max value.**
  + **The sum**
  + **The range: the difference between the max and min**

Development guidelines:

* The output should be neat, formatted and well organized – shouldn’t hurt your eyes!
* Your main() method’s actions should be limited to:
  + gathering command line arguments
  + displaying error messages
  + creating an instance of NumericAnalyzer and invoking its top level method (e.g., “calculate()” )
* Your code should adhere to naming conventions as discussed in class.
* Your implementation should embrace modularity:
  + Each mathematical calculation should be implemented by a separate method.
  + Yet another method should be responsible for displaying a sorted list of the numbers provided, and displaying all derived values above.

So your main() method will include a sequence of instructions similar to this:

|  |
| --- |
| NumericAnalyzer analyzer = new NumericAnalyzer(args);  ….  analyzer.calculate();  analyzer.display();  …. |

You test cases should include testing for 1 value and multiple values – at least 6+ … .

**SAMPLE OUTPUT (1 &2)**

|  |
| --- |
| 22 64 128 256 818 877 8763 102400  Min: 22  Max: 102400  Range: 102378  Sum: 113328  Mean: 14166  Median: 256  256  Min: 256  Max: 256  Range: 0  Sum: 256  Mean: 256  Median: 256 |

PASTE YOUR OUTPUT HERE

ANSWER: Please scroll down

|  |
| --- |
| Hi!, please enter your number >  33 10 5 100 60 65 50  Your numbers are 33 10 5 100 60 65 50  After selection sort, the array is  5 10 33 50 60 65 100  Total: 323  Mean: 46  Median: 50  Range: 95  Maximum: 100  Minimum: 5  This Arrary only consists of one element:10  Total: 10  Mean: 10  Median: 10  Range: 0  Maximum: 10  Minimum: 10 |