Lab 12

Merge Sorter

Solutions limited in scope to:		
•	•	•

Submission Rules:

1. Submissions must be zipped into a **handin.zip** file. Each problem must be implemented in its own class file. Use the name of the problem as the class name.

2. You must use standard input and standard output for ALL your problems. It means that if input is needed the input should be entered from the keyboard while the output will be displayed on the screen.

3. Your source code files should include a comment at the beginning including your name and that problem number/name.

4. The output of your solutions must be formatted exactly as the sample output to receive full credit for that submission.

5. Compile & test your solutions before submitting.

6. Each problem is worth 40 points total.

7. This lab is worth a max total of: 40 points.

8. Submission:

- You have unlimited submission attempts until the deadline passes
- You'll receive your lab grade immediately after submitting

Problem 1: Merge Sorter (40 points)

(Sorting) Merge sort is a divide-and-conquer algorithm based on the idea of breaking down a list into several sub-lists until each sublist consists of a single element and merging those sublists in a manner that results into a sorted list.

UML Class Diagram:

MergeSorter
-
• `mergeSort(): int[]` • `sortArray(): int[]`

***MergeSorter* Method API:**

Modifier and Type	Method and Description
`public int[]`	`mergeSort(int[] data)` Returns a sorted integer array.
`public int[]`	`sortArray(int[] data, int low, int high)` Returns a sorted integer array based on specified low and high indexes.
`public void`	`merge(int[] data, int left, int middle1, int middle2, int right)` `Merges two subarrays based on specified indexes.`

Tester Files:

Use the TestMergeSorter*.java* file to test your implementation. You can use a command like: "java org.junit.runner.JUnitCore TestMergeSorter" after compiling the files.