Assignment: Complexity and Sorting - Marking Criteria Group3 evaluation.

As a group, evaluate your work against each criterion and assign yourselves a mark for each row. If you would like, provide a short explanation for the self-assigned mark in the

Group 3 decided on full marks as we implemented everything correctly. Worked out extensive comments explaining the applications functions. Implemented all the algorithms succsefully. Hard great collaboration on our github repo.

Marking Criteria

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Needs Improvement (0–50%)** | **Good (51–75%)** | **Excellent (76–100%)** | **Group-Assigned Mark** | **Notes** |
| **Working Code** | * Command line does not work properly and does not generate the objects using reflection | * Command line works and generates the objects partially using reflection | * Command line works properly and generates the objects using reflection | **10/10** |  |
| **Style** | * Sorting algorithms don’t work on list of Comparables | * Sorting algorithms partially work on a list of Comparables | * Sorting algorithms work on a list of Comparables | **5/5** |  |
| **Testing** | * Comparable/ Comparator not implemented correctly | * Comparable/ Comparator partially implemented | * Comparable/ Comparator implemented correctly | **5/5** |  |
|  | * Didn’t use correct OOP modelling of objects | * Used somewhat correct OOP modelling of objects | * Used correct OOP modelling of objects | **5/5** |  |
|  | * Incorrect naming and packaging of classes according to Java standards | * Partially correct naming and packaging of classes according to Java standards | * Correct naming and packaging of classes according to Java standards | **/5** |  |
|  | * Bubble sort not implemented in utility class | * Bubble sort implemented partially in utility class | * Bubble sort implemented in utility class | **/5** |  |
|  | * Selection sort not implemented in utility class | * Selection sort implemented partially in utility class | * Selection sort implemented in utility class | **/5** |  |
|  | * Insertion sort not implemented in utility class | * Insertion sort implemented partially in utility class | * Insertion sort implemented in utility class | **5/5** |  |
|  | * Merge sort not implemented in utility class | * Merge sort implemented partially in utility class | * Merge sort implemented in utility class | **10/10** |  |
|  | * Quick sort not implemented in utility class | * Quick sort implemented partially in utility class | * Quick sort implemented in utility class | **10/10** |  |
|  | * Student’s choice of sorting algorithm not implemented in utility class | * Student’s choice of sorting algorithm partially implemented in utility class | * Student’s choice of sorting algorithm correctly implemented in utility class | **5/5** |  |
|  | * Description and complexity analysis are unclear | * Description and complexity analysis are partially clear | * Description and complexity analysis are clear | **5/5** |  |
|  | * Sort Output is dissimilar to expected | * Sort Output is somewhat similar to expected | * Sort Output matches what is expected | **5/5** |  |
|  | * Incorrect timing of sorts shown | * Somewhat correct timing of sorts shown | * Correct timing of sorts shown | **5/5** |  |
|  | * No Readmes/Instructions or javadoc documentation submitted | * Some Readmes/Instructions and javadoc documentation submitted | * All Readmes/Instructions and javadoc documentation submitted correctly | **10/10** |  |
| **Total** | | | | **/95** |