# Assignment 1: Complexity and Sorting - Marking Criteria

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 Notes column.

Note: If your program (the jar file and/or eclipse project) does not run due to compilation, runtime or logic errors, you will receive an automatic zero grade on this assignment!

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **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 | * Command line works and partially generates the objects | * Command line works properly and generates the objects | **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** |  |
| **OOP Models** | * Didn’t use correct OOP modelling of objects | * Used somewhat correct OOP modelling of objects | * Used correct OOP modelling of objects | **5/5** |  |
| **Java Conventions** | * 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 | **5/5** |  |
| **Bubble** | * Bubble sort not implemented in utility class | * Bubble sort implemented partially in utility class | * Bubble sort implemented in utility class | **5/5** |  |
| **Selection** | * Selection sort not implemented in utility class | * Selection sort implemented partially in utility class | * Selection sort implemented in utility class | **5/5** |  |
| **Insertion** | * Insertion sort not implemented in utility class | * Insertion sort implemented partially in utility class | * Insertion sort implemented in utility class | **5/5** |  |
| **Merge** | * Merge sort not implemented in utility class | * Merge sort implemented partially in utility class | * Merge sort implemented in utility class | **10/10** |  |
| **Quick** | * Quick sort not implemented in utility class | * Quick sort implemented partially in utility class | * Quick sort implemented in utility class | **10/10** |  |
| **Choice** | * 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** |  |
| **Analysis** | * Description and complexity analysis are unclear | * Description and complexity analysis are partially clear | * Description and complexity analysis are clear | **5/5** |  |
| **Output** | * Sort Output is dissimilar to expected | * Sort Output is somewhat similar to expected | * Sort Output matches what is expected | **5/5** |  |
| **Benchmarking** | * Incorrect timing of sorts shown | * Somewhat correct timing of sorts shown | * Correct timing of sorts shown | **5/5** |  |
| **Documentations** | * No Readmes/Instructions or javadoc documentation submitted | * Some Readmes/Instructions and javadoc documentation submitted | * All Readmes/Instructions and javadoc documentation submitted correctly | **5/5** |  |
| **Group Evaluation** | * No group application evaluation submitted. | * Some parts of the group application evaluation submitted. | * All parts of the group application evaluation submitted. | **5/5** |  |
| **Total** | | | | **/95** |