# Assignment: 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.

## 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 | **4/10** | **We had each part woking on their own but were not able to get it all working together** |
| **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 | **2/5** | **Sorting algorithms all are there and work with some comparable** |
| **Testing** | * Comparable/ Comparator not implemented correctly | * Comparable/ Comparator partially implemented | * Comparable/ Comparator implemented correctly | **2/5** | **Had the framework all ready but ran out of time** |
|  | * Didn’t use correct OOP modelling of objects | * Used somewhat correct OOP modelling of objects | * Used correct OOP modelling of objects | **5/5** | **We followed good oop standards** |
|  | * 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** | **Everything was named correctly** |
|  | * Bubble sort not implemented in utility class | * Bubble sort implemented partially in utility class | * Bubble sort implemented in utility class | **3/5** |  |
|  | * Selection sort not implemented in utility class | * Selection sort implemented partially in utility class | * Selection sort implemented in utility class | **3/5** |  |
|  | * Insertion sort not implemented in utility class | * Insertion sort implemented partially in utility class | * Insertion sort implemented in utility class | **3/5** |  |
|  | * Merge sort not implemented in utility class | * Merge sort implemented partially in utility class | * Merge sort implemented in utility class | **6/10** |  |
|  | * Quick sort not implemented in utility class | * Quick sort implemented partially in utility class | * Quick sort implemented in utility class | **6/10** |  |
|  | * 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 | **2/5** |  |
|  | * Description and complexity analysis are unclear | * Description and complexity analysis are partially clear | * Description and complexity analysis are clear | **2/5** |  |
|  | * Sort Output is dissimilar to expected | * Sort Output is somewhat similar to expected | * Sort Output matches what is expected | **4/5** | **We know the sorts worked but were not fully able to test** |
|  | * Incorrect timing of sorts shown | * Somewhat correct timing of sorts shown | * Correct timing of sorts shown | **5/5** | **Test class worked fine** |
|  | * No Readmes/Instructions or javadoc documentation submitted | * Some Readmes/Instructions and javadoc documentation submitted | * All Readmes/Instructions and javadoc documentation submitted correctly | **8/10** | **We made clear readme and comments on our work** |
| **Total** | | | | **/95** |