# Assignment: Creating ADTs and Using an XML Parser - Marking Criteria: Submission 2

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.

### Group 2: Duc Hoang, Sean Lam, Edward Magtoto, Matt Yackel

## Marking Criteria

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Needs Improvement  (0–50%)** | **Good  (51–75%)** | **Excellent  (76–100%)** | **Group-Assigned Mark** | **Notes** |
| **Parsing Tags** | Incorrect logic used for parsing tags | Partially correct logic used for parsing tags | Correct logic used for parsing tags | **7/7** | Not applicable |
| **Output** | Program produces the wrong output | Program produces a partially correct output | Program produces the correct output | **5/5** | Not applicable |
| **Stack  Implementation** | Stack incorrectly implemented using MyArrayList data structure | Stack implemented somewhat correctly using MyArrayList data structure | Stack correctly implemented using MyArrayList data structure | **9/9** | Did not use java.util as required, all functions are properly implemented using @overide |
| **MyArrayList Implementation** | Incorrect implementation of MyArrayList | Partially correct implementation of MyArrayList | Correct implementation of MyArrayList | **9/9** | Did not use java.util as required, all functions are properly implemented using @overide |
| **Queue Implementation** | Queue incorrectly implemented using MyDLL data structure | Queue implemented somewhat correctly using MyDLL data structure | Queue implemented correctly using MyDLL data structure | **9/9** | Did not use java.util as required, all functions are properly implemented using @overide |
| **MyDLL Implementation** | Incorrect implementation of MyDLL | Partially correct implementation of MyDLL | Correct implementation of MyDLL | **9/9** | Did not use java.util as required, all functions are properly implemented using @overide |
| **JUnit tests for ArrayList** | Incomplete sets of JUnit tests for ArrayList | Partially complete sets of JUnit tests for ArrayList | Complete sets of JUnit tests for ArrayList | **9/9** | All functions are tested, including tests to fail if needed |
| **JUnit tests for DLL** | Incomplete sets of JUnit tests for DLL | Partially complete sets of JUnit tests for DLL | Complete sets of JUnit tests for DLL | **6/6** | All functions are tested, including tests to fail if needed |
| **JUnit tests for Stack** | Incomplete sets of JUnit tests for Stack | Partially complete sets of JUnit tests for Stack | Complete sets of JUnit tests for Stack | **6/6** | All functions are tested, including tests to fail if needed |
| **JUnit tests for Queue** | Incomplete sets of JUnit tests for Queue | Partially complete sets of JUnit tests for Queue | Complete sets of JUnit tests for Queue | **6/6** | All functions are tested, including tests to fail if needed |
| **Documentation** | Submitted no readmes/instructions or javadoc documentation | Submitted some readmes/instructions and javadoc documentation | Submitted correct readmes/instructions and javadoc documentation | **6/6** | Submitted with projectStatus at root folder, properly zipped, Javadoc is completed as required |
|  | | | | **81/81** |