|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Rubric for C++ Object Oriented Programming Assignment – Poker Hand Evaluator** | | | | | | | | |
| Criteria/Objective | **Performance Level** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **Mark** |
| Ability to implement application logic and to integrate with existing code | * All hand types are correctly sorted between the hand type and within the hand types. * Supplied code to read from file works as supplied. * Expected results for all test files.   (22 – 25 marks) | * Hand types are correctly sorted between hand types and some are correctly sorted within the same hand type. * Expected results for supplied test files.   (19 – 21 marks) | * All hand types are correctly sorted between hand types. * Expected results for most supplied test files.   (16 – 18 marks) | * Some hand types are correctly sorted between hand types. * Expected results for some supplied test files   (10 – 15 marks) | | * Hand types are not correctly sorted. * Supplied code to read from file does not work. * Supplied test files produce incorrect results   ( 0 – 9 marks) | | / 25 |
| Ability to create definitions and implementations of abstract data types | * All ADTs defined and implemented according to the specification.   (9 – 10 marks) | * Most ADTs defined and implemented according to the specification.   (8 marks) | * Some ADTs defined and implemented according to the specification.   (7 marks) | * ADTs have been defined and implemented to achieve the aims of the assignment.   (4 – 6 marks) | | * ADTs do not achieve the aims of the assignment.   (0 – 3 marks) | | /10 |
| Ability to write code that is efficient with respect to memory use and data structures. | * Pointers to objects are used in all collections / arrays AND all object instances have been dynamically created.   (9 – 10 marks) | * Pointers to objects are used in all collections / arrays OR all object instances have been dynamically created.   (8 marks) | * Pointers to objects are used in one collection / array. Uses code and data structures which result in a few minor inefficiencies.   (7 marks) | * Objects are used but are not dynamically created. Uses code and data structures which result in some inefficiencies.   (4 – 6 marks) | | * Uses inefficient code and data structures with regard to memory use and execution speed.   (0 – 3 marks) | | /10 |
| Ability to write code to a professional quality. Consideration given to comments, variable names, layout etc. | * Code is fully commented AND clearly follows a professional coding standard. Code is readable.   (5 marks) | * Code is fully commented OR clearly follows a professional coding standard.   (4 marks) | * Code is mostly well commented AND mostly follows a professional coding standard.   (3 marks) | * Code is mostly well commented OR mostly follows a professional coding standard. * Code is mostly readable.   (2 marks) | | * Code is not well commented and does not follow a professional coding standard.   (0 – 1 marks) | | /5 |
| / 50 | | | | | | | | |

**Additional Comments:**