CMP2801M - Advanced Programming - Assessment Item 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Learning Outcome** | **Criterion** | **Pass** | **2:2** | **2:1** | **1st** | **1st++** |
| [LO2] Use advanced  object oriented  principles and  programming  techniques in  software  development (50%); | Implementation | Simple class structures are  used. Those classes may  incorporate variables and  data structures, but no  thought has been given to  their OOP principles. | Class definitions are appropriate. Inheritance relationships are implemented, although with maybe flawed or missing elements. Access modifiers are in place but may not be correctly used. | Class definitions are appropriate. Inheritance relationships are implemented with minor flaws. Access modifiers are correctly used in the classes. Abstract classes and virtual functions have been implemented as requested. | Classes are defined to match the assignment brief perfectly. The inheritance relationships are correctly implemented. Access modifiers are correctly and effectively used. Advanced techniques (e.g., operator overloading) are demonstrated and effectively utilised in the application. | The implementation demonstrates advanced knowledge of both OOP and functional paradigms (i.e. lambdas). Code repetition is kept to an absolute minimum and careful thought has gone into the application’s extensibility. |
| [LO3] Apply  advanced logical and  mathematical  techniques in the  development of  software solutions  (50%). | Implementation | The code compiles and executes. Fair program structure and some code comments. A working software with basic functionality is demonstrated, accomplishing the assignment tasks partially. | The code compiles and executes. Clear program structure and appropriate comments. A working application is demonstrated, accomplishing most of the tasks. | The code compiles and executes. The program code is well structured and commented. Good demonstration of C++ features, such as collections, pointers and memory management. The functionality of the code is almost complete. | A complete implementation featuring all the desired functionality. The program code uses appropriate evaluation strategies, is well-structured and commented. Good use of programming structures and C++ features is demonstrated. Some additional features have been attempted. | The application is a production-worthy solution that fulfils all elements of the brief and makes effective use of STL/contemporary C++ components. Some innovative additional have also been implemented. |
| **Weighting is 70% of the module** |