CMP2801M - Advanced Programming - Assessment Item 1 RESIT

Learning Outcome	Criterion	Pass	2:2	2:1	1st	1st++
[LO1] Apply concepts of advanced software development and programming methods to computational problems (40%)	Implementation	The code compiles and executes. Fair program structure and some code comments. A basic menu is displayed, and users can complete orders, but one or more user options may be incomplete or missing. It may not be clear to the user how to perform tasks.	The code compiles and executes. Good program structure and appropriate comments. A complete menu is displayed, and users can complete orders. All user options are present but may have flaws in their implementation. It is clear to the user how the application operates.	The code compiles and executes. Program is very well structured and commented. There is good demonstration of C++ features, such as collections, pointers and memory management. All user options are present and work correctly. The interface is well-designed and clear.	The code compiles and executes. The program is very well structured and commented with careful consideration for efficiency and appropriate evaluation strategies. There is good demonstration of C++ features, and all user options are correctly and elegantly implemented. The interface is aesthetically pleasing, and easy to use.	The implementation fulfils the 1st class criteria and goes beyond the brief. The application is a production-worthy solution that makes effective use of STL/contemporary C++ components. One or more of the stretch tasks have been attempted.
[LO2] Use advanced object oriented principles and programming techniques in software development (40%);	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 may not be correctly used. OOP features such as operator overloading may be evident in the code, but may have limited utility.	Class definitions are appropriate. Inheritance relationships are implemented with minor flaws. Access modifiers are correctly used in the classes. OOP features such as operator overloading are evident in the code and are well utilised.	Classes are defined to match the assignment brief perfectly. Access modifiers are correctly and effectively used. The inheritance relationships are fully and correctly implemented. There is effective usage of multiple OOP features, such as operator overloading, friend functions, etc.	The implementation demonstrates advanced knowledge of both OOP and functional paradigms, facilitating an elegant solution to one or more of the stretch tasks outlined in the brief.
[LO 3] Apply advanced logical and mathematical techniques in the development of software solutions (20%).	Report	Introduction is basic, it introduces the problem, but there is no more. Description of the code structure is incomplete. Results are limited in scope, and almost no attempt is made to evaluate the program.	The introduction provides an overview of the problem. Program structure and logic is described in a satisfactory way. Results are presented but the analysis of the program is lacking.	The introduction provides an explanation of the problem. Program structure and logic is described well. Results are presented and the analysis of the program is given.	The introduction provides a detailed explanation of the problem. Program structure and logic is described very well. Results are presented well, and the program is comprehensively evaluated.	A detailed examination of the stretch tasks is also given, and the program design is critically evaluated with close attention paid to the algorithmic choices and time complexity of the additional tasks.
Weighting		Weightings are indicated				