CMP2801M

Assessment 1 RESIT

1 Implemented functionality

Briefly discuss which classes are implemented and how much of your implementation differs from the brief. Specifically, please indicate whether the following were implemented (and if not, state what you did instead):

- All item types (Appetiser, MainCourse, Beverage) derive from the Item class

All derived from the Public Item class and implemented as specified.

- Both the Order and Menu classes derive from the ItemList class

Both are derived from the ItemList class and implemented as shown in the diagram.

- Virtual functions, including pure virtual

Used virtual “toString” functions in both base classes to be used in derived classes.

- What error checking you performed (bounds, values, limits)

If input values don’t match a valid index shown in the menu e.g. “add 2”, then it will display an error message and not compute anything. This goes for every command that we had to specify in takeaway.cpp (like add or remove).

Where unexpected inputs arise, an error checking method has been used to take you back to the main program and asks for the input again.

- Any additional functionality, e.g. operator overloading is used to use the << operator in place of toString for Item’s derived classes. Another option would be to overload comparison operators (i.e., <, >, =) to allow for items to be sorted by their prices.

None of this has been implemented, only the core functionality.

2 Test cases

Briefly describe how you tested your program. For each function (if applicable) and the program demonstrate some test cases in a table form and indicate whether the tests were passed or not.

Please note that this is for us to evaluate how you approach software testing. The test cases we evaluate in the evaluation stage will be different and you will receive a mark will be affected by the number of test cases your program passes from within the cases we define. Below is an example to get you started:

Test Case	**Input Values**	**Expected output**	**Passed?**
Use invalid command to add to order	Add 13	Displays error message saying invalid input / invalid index	YES
Use invalid command	hlep	Displays invalid input warning	NO
Removing invalid item from order	Add 1 add 2 remove 3	Throws error message	YES
Using invalid input when prompted for checkout	Any letter other than y or n	Asks for input again	YES
Add item with negative index	Add -1	Gives last item in array	NO Gives invalid index message
Checkout after negative index error	Add -1 checkout	Goes to checkout that's empty	NO Bug wont allow other commands
Testing double 2-4-1 checkout	Add 1 Add 2 Add 1 Add 2 checkout	Adds everything, gives discount and calculates properly	YES
Receipt Testing, replicate console	Add 1 Add 2 Add 1 Add 2 Checkout y	Replicates the console output from checkout to a receipt.txt file	YES