

TCES 203
Programming Practicum
Assignment 5 – Multiple Classes
15 Points

This assignment tests your understanding of concepts covered in the course dealing with classes. For this assignment, you have to create two classes with the corresponding header and implementation files as well as a test file that contains the main function.

The **GroceryItemOrder** class should store an item's name, quantity and price per unit. A GroceryItemOrder object should have the following functions and any necessary accessors:

- A constructor that creates an item order to purchase the item with the given name, in the given quantity, which costs the given price per unit. The name must be a cstring.
- A function that returns the total cost of this item in its given quantity. For example, 4 boxes of cookies that are 2.30 per unit have a cost of 9.20.
- A function that sets this grocery item's quantity to be the given value.
- A destructor that deallocates any dynamic memory.

Test the GroceryItemOrder class first before moving onto the next class.

The **GroceryList** represents a person's list of items to buy from the market, and another class named GroceryItemOrder that represents a request to purchase a particular item in a given quantity (example: 4 boxes of cookies).

The **GroceryList** class should use an array field to store the grocery items, as well as keeping track of its size (number of items in the list so far). Assume that a grocery list will have no more than 10 items. A GroceryList object should have the following functions:

- A constructor that creates a new empty grocery list.
- A function to add the given item order to this list, if the list is not full (has fewer than 10 items). This function must take an object of type GroceryItemOrder as a parameter by reference.
- A function that returns the total sum cost of all grocery item orders in this list. No parameters can be passed to this function.
- A destructor that deallocates any dynamic memory.

Test the functions of the GroceryList class.

Submission and Grading:

Submit GroceryItemOrder.h, GroceryItemOrder.cpp, GroceryList.h, GroceryList.cpp and main.cpp under the Assignments section of the course web page as one zip file with the name Grocery<LastName>.zip (Both last names in the case of pairs, exclude the < >).

There will be points taken off for not following the conventions listed in this document regarding submissions, outputs and naming conventions.

You are required to properly indent your code and will lose points if you make significant indentation mistakes. See the textbook for an explanation and examples of proper indentation.

Give meaningful names to functions and variables in your code. Localize variables whenever possible -- that is, declare them in the smallest scope in which they are needed.

Include a comment at the beginning of your program with basic information and a description of the program **and include a comment at the start of each function**. Your comments should be written in your own words and not taken directly from this document. Write comments within functions to explain the flow or any obscure code. Provide comments for the functions as well as for the class definition. Make sure that every file has a header comment including the .h and the main.cpp files.

You should include a comment at the beginning of your program with some basic information and a description of the program, as in:

```
// Menaka Abraham
// 3/30/15
// 203
// Assignment #1
//
// This program will...
```