Rob Navarro
CS162

Assignment 2

Design:

This program will require both a Item and List class to be implemented. After reviewing the instructions, I have determined that the Item class will need several private variables. These will include an item name, unit, number to buy, and price per unit. In additional to these variables I will need to also include setter/getter methods for the class. Also, I will need a function to overload the = operator so that different item objects can be compared.

The list class will contain most of the functions that do the work. I think a private vector that contains Item objects will be the best bet for storing items on the list. Within the vector I will need to manipulate the data. To do this I will need an add item function. This will ask the user for all of the info needed to declare an Item object. Inside this function I will also need to check and make sure that I am not adding a duplicate value. If a duplicate is found I should ask the user if the want to update the to buy. I can update this using the setter methods with the Item class.

In addition to adding items I will also need a function that removes items. This function should ask the user what item they want to delete and then find the item name within the vector with all of the items. A for loop should be able to handle this fairly easily. Finally, I will need a function that displays the list. While displaying this list I should have a variable that keeps track of the total price of the list. Also, I will need to calculate the total price for each individual item and display it on the list. For simplicity purposes I will likely separate each parameter with a comma.

With both classes set up I should be able to just call the functions from within the main function. I think a do/while loop is a good option for allowing the user to select what they'd like to do. I can include an option that will allow them to exit the program whenever they are finished with the list.

Testing:

For the testing of this program I decided to test each class as I built them. Starting with the item class, I created several different objects and then tested them within the main program. I made sure that all of the getter/setter methods were working as expected. I also made a simple test to determine if the overloaded = operator was working correctly. I found that this part was fairly simple and no changes were needed during the testing.

The list class took a bit more testing due to several additional functions. I went ahead and tested each function as I created them. Adding items to the vector was fairly straight forward. After a few initial tests I found that items were being stored correctly. I confirmed this by calling my getter methods in the item class and displaying my results. I actually ended up stealing part

of this test and applying into to the displaying list function. Once I was happy with items being added to the list I decided to work on displaying the list.

After stealing some of the ideas from my adding function I was able to create my display function fairly easily. To test this function, I would add several items to the list and then display the list on the screen. After going through this process several times I went ahead and tested the delete function. I would test the delete function by adding items and then removing them from the list. In between this I would display the list in order to be sure that the items were being removed correctly.

Finally, after testing all of the functions inside the list class, I proceeded to test the main program. I had created a main menu that allows the user to select what they would like to do. I went through this part several times confirming that main menu was calling the functions correctly. I made sure to create lists that were several different sizes and then I tested removing each item from the list. Some minor changes were required along the way but for the most part things worked as expected.

Reflection:

Planning out this program helped me stay fairly close to the original design of the program. I think the only changes that I really made was how to display the user interface. Implementing the = overloader for the item class also required me to move things around in my program but I didn't have to make any wholesale changes.

Many of the challenges that I faced with this assignment was figure out how to format numbers onto the screen. I spent a good amount of time reading on how numbers could be formatted so that they appeared as a currency should. I also made a few changes on how items would be added. Originally I was thinking about storing info in the main program and then passing an item class to the functions. However, after I started working I found it easier to build the items with the functions instead of passing items into them. This allowed me to build a much cleaner main program that only focused on the main menu.

For the most part I felt that this assignment built on several previous topics that we had been practicing. This allowed me to build an initial program that did not require that many changes.