Design

Program being designed: Grocery List program

Files needed: List, Item, and Main

Classes needed: Item; List

Item class:

- item name
    - String
- unit (i.e. cans or ponds)
    - String
- number to buy
    - Integer
- price of the item
    - Double
- Will need getters and setters for all four above ^.

List Class: needs an array to store Item objects in your List. When an item is added an Item object must be created with the information and added to the list object.

- Add Items to array
    - Get name, unit, quantity, and price. Make that a new item then add that to the List array.
- Dynamically double the size of the array when the array gets filled
    - Check to see if the array count equals capacity and if it does make a new array twice the size and transfer the items over.
- Remove an Item
    - This can be done for checking for the name of the item that you want removed and removing an item with that name
- Display the List
    - Set the precision of the cost to two here instead of at input. That way if that want to see how the cost would be with tenths of a penny they could and the math will still work out to the value they would actually pay. They can also always remove the item if it was an error in their input.
    - Also will display the total cost for each item and their total purchase.

The main: Set up four options in the UI. Set up input validation to make sure that they cannot choose something other than the options given.

    - Add an Item
    - Remove an Item
    - Display List
    - Exit

Test Plan

Test Case	Input Value	Expected Outcome	Observed Outcome
Test 1: Use 1 item to make sure that the items are properly adding to the array	Item1: apple, lb, 5, 2	This Item will be added to the array and will be able to be accessed by the other functions	The item when displayed showed properly with the correct corresponding math
Test 2: Four Items to fill the array	Item1: apple, lb, 5, 2 Item2: pear, lb, 3, 2 Item3: peach, lb, 3, 5 Item4: grapes, bunch, 1, 3	The expected outcome is that the array when displayed will give back these items with the item total as well as the entire list total	The Items were displayed properly and the math came out as apple costing 10 pear costing 6 peach costing 15 grape costing 3 Total being 34
From test number 2 I was able to determine that the array works with four items and that our math is giving back the values as expected.			
Test 3: Five Items to test the dynamic array	Item1: apple, lb, 5, 2 Item2: pear, lb, 3, 2 Item3: peach, lb, 3, 5 Item4: grapes, bunch, 1, 3 Item5: mango, each, 3, 5	The expected outcome is that the array will double in size to allow the fifth item into the array without any issues	The array doubled in size allowing for the fifth item to be added properly
From test number 3 I was able to determine that the array works with more than four items showing that the dynamic array is set up properly.			
Test 4: Remove an Item from a list of only that Item	Item1: apple, lb, 5, 2 Is added to the list and then removed with the word apple	The expected outcome is that after the item is added to the list that the word apple should remove the item and therefore leave us with an empty list.	The list is empty after using the remove function with the word apple.
From test number 4 I was able to determine that the remove function works when there is only one item in the array. This shows that the function works on a base level.			
Test 5: Test the remove function with multiple items to verify that it can work with an array properly.	Item1: apple, lb, 5, 2 Item2: pear, lb, 3, 2 Item3: peach, lb, 3, 5 Item4: grapes, bunch, 1, 3 Remove the apple by entering the word apple when prompted	The expected outcome is that the apple should be removed from the list while the rest of the items remain on the list.	The function properly removed the apples without affecting any of the other items. The total cost of the cart also properly changes to account for the change in the List.

From test number 5 I was able to determine that the remove function is able to go through the array and find the proper Item and remove it without effecting the other items but properly changing the Cart total.

Reflections:

This assignment took me longer than I expected and caused me to reread content in the book and look at content from CS 161. I struggled at first with making an array of objects with multiple attributes. I went through various ways of formatting the input of Items into a List and I was unable to solve it for a while. I solved this by rereading chapter 8 on arrays and chapter 10 on pointers. Once I figured out how to make the array I was able to do the rest of the program relative ease. I made the decision to remove items based on the name and to remove the first item in the list and that way if someone added multiple Items of the same name they could remove them in the order of them being added. I found this assignment to be very beneficial to be the first assignment of the quarter. It allowed me to review old topics and remaster important concepts that we touched on last quarter.