Thanhbinh Truong
933233558
Truontha@oregonstate.edu
CS 162

For project 2, the assignment was to design, implement and test a grocery shopping list program. The program was required to include a few classes such as an item class and a list class. To be able to start designing the program, I had to first try to write out what needed to be included in the items and list class. Making the item class was simpler than the list, since most of what was needed was already described in the requirements. The challenging part was making the list class because one requirement was that its not to contain a vector, which would have made things a little easier, and not knowing what other member variables and functions to include within the class. The main problem was how would I use an array that would need to take a size that could expand at any moment. While trying to figure this problem out, I turned the piazza to see if anyone else has this problem, which luckily someone did. The solution that was brought up was to copy the existing array into a new array that was created to hold a larger amount of items.

Another issue I had while making the list class was how to create a member variable that would point to the items list. At first, I started by making it just a pointer that pointed to an item class. After creating the two classes, I began to create my main function, yet every time I ran trial runs I would always run into errors that dealt with my pointer groceryList. From these errors, I had to try other options that the textbook had such as pointer to pointer. Applying this method to the program made problem solving a lot easier.

Furthermore, some smaller problems occurred for me while trying to perfect my list class. I had to make a way to properly set the size of the original array be a maximum of 4 spaces, while being able to make a larger array when needed. This is when I thought to make an int originally set at 4 to help create the first array or pointer, depending on what was going to be used. From there, I would make a function that would check the number of items in the array with the maximum size that it should fit, and if equal, then recreate an array and move all the values over with a larger maximum size to hold more items.

Test Plan

① Add Item

	Item	qty	unit	price	exp. Result	Result
①	Strawberry	3	box	$3	Item added	Item added
②	egg	e	cart	$3	error on ① qty ② unit	error for qty, unit
③	orange	2	box	run	error on run	error for run.
④	Strawbery	3	box	$4	alert for copy, ask for update	prompt "already added," ask for update.

② Delete Item (from before list)

1. (Added strawberry)
   - Delete — "Prompt which to delete — strawbery

   exp. result: deletes strawbery
   result: deletes strawbery
2. (Added eggs
   - 2 Delete option - "Prompt which to delete - egg

   exp. result: No "egg" in cart
   result: "no egg in cart

Classes

1. Item
   - (private)
     1 - name, unit (strings)
     2 - Price (per unit), total (xprice), quantity (double)
         qty
   - (Public)
   - Item Constructor (default)
   - Item Destructor
   - Item Constructor w/ parameters
   - Set/get for each member variable
   - Print function to print info of every member function

List

- ★ arrayCheck / arraySize - to set and the size of the original array
- (private) groceryList (array of items? pointer to items?)
- totalCost (double) - to hold total cost of everything in the cart.

(public)

List constructor, destructor.

- List constructor w/ size parameter.
- Functions
  1- AddItem (add item to list)
  2- DeleteItem (delete item from list)
  3- DoubleSize (double size of array)
  4- get/set functions for private members
  5- getTotalCost - get List's total cost