CS162 Project 1

Khoa Phan

4/30/17

Project Reflection

This project shopping list gave me a good insight of dealing with classes and using dynamic arrays with them. My original design was based off of menus and how they branched from there. In order to achieve this, I had to make sure I was pointing classes correctly through the menu. I used an outline to organize my thoughts throughout the process. This was my outline:

Main file, starts program

- Call menu

Menu class

- Add/remove item
    - Add item
        - Create new item, write to object, forward object to list array
        - Store object into list array
        - If objects in list > 4, increase array size by 1
    - Remove item
        - Select object from list, remove object from list array
        - Decrease array size if list is larger than 4
- Display Item
    - Display item
        - Display list of items, by name
        - Select item and display item information
    - Display list
        - Displays all item information from list
- Exit program

Using this outline, I started to fill in pseudo code for each step. In my process, I learned how to increase dynamically allocated arrays by copying the information and creating a new array. This meant reallocating information and deallocating old ones. Adding was particularly easier. When removing, I had tried to just remove the last item, but wanted to incorporate a way to select the item. This meant I needed to rework my array to remove a particular element and shifting the rest of the items down one element. This was done by copying all but the selected item into the new array. During the process, it was difficult for me to understand where I needed to deallocate the memory and how to go about writing the new array and passing it on. By deleting my old item list, I was able to reestablish it by pushing it out from a function.

The classes I mainly used were the item class, list class and menu class. The item stored all specific item data, while the list class enabled all the functions to write and display items. The menu class really gave an interface to access all of these functions.

Another important thing I learned was overloading functions. Although it was not used nearly enough in this project, I had come to a better understanding of how to approach a situation where it can be used. By overloading the operator, I was able to compare two item names and see if they were equal. This allowed me to check for duplicates.

I referenced an old project I had done back in CS162 with a similar concept. It used vectors instead but this project gave me an understanding on what goes behind vectors using the arrays. Reusing my menu and validation classes were of much help and saved me a ton of time. There is definitely a big learning experience with the project in particular because passing objects had worked along with Lab 4 as well.

As for my test plan and testing results, I had tried to come up with all the situations that may occur. To me these were the following:

Test Case	**Input Values**	**Driver Functions**	**Expected outcomes**	**Observed Outcomes**
Adding Item Outside of array size	Call set functions, user input information	increaseList new array + 1, object added increase listSize,	New Array size + 1 if > 4, Copy old, new object written to element.	New Array size + 1, Copy old, new object written to element.
Removing item from specific location	Prompts user to select item to be removed	removeItem new array -1, remove object decrease listSize	New array size – 1 if > 4, copy old array elements into new except selected	New array size – 1 if > 4, copy old array elements into new except selected
Adding a duplicate item	Prompts for user to update or cancel	dupItem check for duplicate as for update or cancel	If updated, item elements will be edited. If cancelled, the prompt will revert back to screen without any changes	If updated, item elements will be edited. If cancelled, the prompt will revert back to screen without any changes
Display selected item	Select from list, user input	selectItem select from list display information	User input-1 Finds that input in array and calls displayInfo with correct information	User input-1 Finds that input in array and calls displayInfo with correct information
Displaying list of items	n/a	displayList call get functions to display items	Loop through array, Display all items in array by calling getter functions	Loop through array, Display all items in array by calling getter functions

Outside of data validation, there weren't too many problems in my approach that I could see. I however, wished I had recorded my issues better throughout working on this project. This would enable me to reflect with better documentation and have a better understanding at the end of what I achieved.