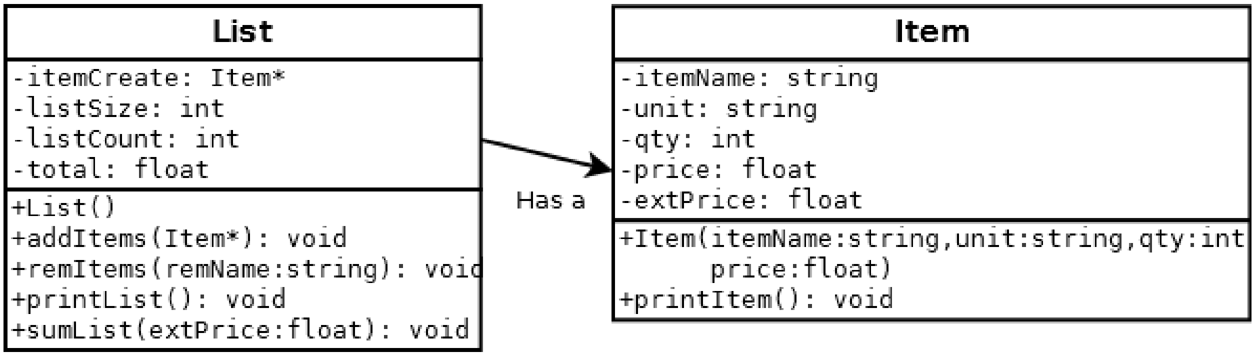
Justin Hammel
CS-162
Spring 2017
Project 2 Design and Reflection

1. Define what the program should do.
    a. Purpose: Design and implement a program to create and manage a grocery shopping list.
    b. Input:
        i. All the information regarding the items being added to the list; string itemName, string unit, int qty, float unitPrice
        ii. User input is expected for the menu options; Create list, add items, remove items, print list, and exit program
        iii. Remove item selection will prompt user to input the item name to search for
    c. Processing:
        i. Implement input validation for float numbers, already have this for integers. Could take the value as a c-string instead of a regular string in order to use atof(part of cstdlib) to convert the digit to a double. At the same time use atoi to convert the number to an integer and then subtract the integer from the double.
        ii. Search function to look through the characters in the name string to find a match. If that match is found return as a Boolean variable to prompt response from the user.
        iii. There will be a function to calculate the total price of the item based on unit price * qty, then this will be summed in a list total price.
        iv. There needs to be a method to dynamically update the size of the array as items are added to the list making it go over 4 items. As the add/remove item function is called this should alter a counter in the List class that will keep track of the size of the list.
    d. Output:
        i. Initial menu will output to the screen
        ii. Expect error outputs for violating different input validations as appropriate
        iii. Print list menu option will output the contents of the list array to the screen
        iv. Add/remove items will contain he appropriate responses necessary to allow the user to use those functions

2. Program design
    - Initial menu prints to the screen from the Menu class
        - Options are Create list, add items, remove items, print list, and exit program
        - Program will request that the user input an integer value, stringChoice, corresponding to the choices on the screen
        - Input validation for the menu will return false Boolean value if the input is not an integer, and does not correspond to one of the available choices
    - Create list selection will result in generating a new List object, currentList

    - List will be created as a dynamic area with a starting size of four, listSize will control the dynamic arrays size
- When add items is selected user is prompted to enter itemName, unit, qty, and unitPrice
    - Item object will be created and pointer to it will be stored in the list array
    - Total price for the qty of items will be calculated and added to the total list sum
    - Counter will be used to keep track of items currently in the list and as the index to add items to the array
    - If the counter exceeds the list size new space will be allocated to the array to match new larger size then the new item will be added to the end of the array
    - After each item is added search function will run to determine if the item already exists on the list
        - Search will be performed based on the item name
        - If item name found prompt if you wish to replace item with new one, item will be updated with new data, or user will return to main menu
        - If the item is to be updated, subtract the total price for this item from the sum, then add total price based on new quantity/price etc
- On selecting delete items search function will be called to find the item name
    - Program will first have to prompt the use to input the name of the item to be deleted and then pass this to the search function
    - If the item is found remove it from the list, move the items over to occupy the now blank space
- Print list function will print the contents of the list array by calling the item print function using the pointers stored in the array
- Exit will end the loop quitting the program

3. Class Diagram



4. Testing Plan

TEST	INPUT VALUES	DRIVER FUNCTIONS	EXPECTED RESULTS	OBSERVED OUTCOMES
Float validation	Input non-float prices for items	Testing float input validation	Error message to input valid numbers	I was unable to get float values working for this program. My input validation works to check for float input. I was unable to get the conversion from string to float working.
Items over 4	Input up to 6 items	Validate that dynamic memory allocation for array works to increase array size	Expect the new array to resize as needed for items over 4	Adding additional items over 4 works as expected. The array resizes as expected.
Search for items	Test with trying to remove item And with adding an already existing item	Search overload function	For remove should find and then remove item For the add item should prompt if the item needs to be updated with new data in list	Already existing items are correctly identified. Determined that the overload operator was not necessary for the remove item function. Decided to use just the inputted string and the retrieved string from the Item object to make the remove item search check.
Print list	None	Print function	Should move through the array printing each list item using a call to the print item function	All the items from the list are printed without the for loop moving out of bounds, or printing elements of the array with no Item's assigned to them.

5. Reflection

After getting some TA help concerning pointers both programing assignments from this week made much more since to me. The biggest downside to this was that my original plan for this project ended up not being very well thought out. Despite that, I am glad that I have finally started to get a better handle on using pointers and dynamic memory operations, even if I did have to rethink my whole program.

Originally when I started I did not have a good concept of how the actual List portion of the project would work. Without fully understanding how to manipulate pointers and realizing how classes could interact I never would have thought to make the list consist of an array of item pointers.

There are a few things I would still like to figure out with this project though. I was never able to figure out how to get floats to work for the program. I feel like I have a good grasp of how the conversion functions work, but they never worked for me. Even when I moved to different methods of conversion (stof, atof, strtof) my output had the decimal portion cut off. The funniest thing is that through all this I discovered that I really didn't even need to create my own input validation as some of the conversion methods actually do this for you.

I was feeling really overwhelmed the past few weeks, but now I think I am finally getting the hang of working with pointers. Now I just have to see if something new gets thrown my way that breaks my brain again.