Daniel Olivas

10/24/16

Assignment 2 reflection

For the design of assignment 2, I decided I would just use the item class and list class along with a main.cpp. Item class was nothing more than constructors and setters/getters for the item name, type, quantity, and price per unit. I also included a getTotal function that returns quantity multiplied by unit price.

The List class has 4 functions: addItem, removeItem, displayList, and menu. The menu function simply displays the 4 options the user has to choose from: add item, remove item, display list, and exit. Menu is called in main, and a do while loop is used with a switch menu to choose which function gets called. If 1 is selected addItem is called, 2 calls removeItem, 3 calls displayList, and 4 exits. Any other integer displays "Invalid Selection."

addItem asks user to enter item name, unit type, quantity, and unit price and adds variables to itemList array at the current position. itemList starts with a size of 4. A variable current keeps track of current array position and increases after every add. When current is no longer less than size(initiated at 4), then a temp dynamic array is used to copy itemList. Size is doubled (from 4 to 8, 8 to 16, etc.) every time current reaches size.

removeItem calls displayList. The user selects the number corresponding to the item they wish to select. If no items are present, "No items to remove" will be displayed. A temp dynamic array is used to copy items up to item being removed. Then the items after the removed item are copied. Current is then decremented.

displayList function uses a for loop to display the item variables. Included in the loop is a call from item class for getTotal to add up the total amount the items will cost.

I tested the code incrementally throughout the process of creating it. Upon completion of the addItem function being completed, I made the size of the array large enough that I could input items without having to "resize" the array. Once that was successful, I then called the display function to ensure that the items were being displayed properly. I then changed the size of the array to 1 so it would have to make a copy array to reallocate itemList with a larger size. The final function to test was the removeItem function.

Some problems I had when testing my program was when I entered strings with 2 words like ice cream, name would be ice and cream would then be entered into unit type. To overcome this, I used std:getline(std::cin, input), which enables you to input more than 1 word in a string. displayList initially was looped with size of array rather than current array position. When 2 items were entered with a size 4, then the display would show uninitialized array items.