

Chermaine Cheang

Assignment 2 Reflection

The first problem I encountered while implementing my program was that the program would not compile. The compiler kept throwing an error about me initializing the dynamically allocated array to null. Upon further investigation, I realized that in my original design, my array is an array of Item objects. Since Item objects are not pointers, they could not be initialized to NULL. Hence, I had to change my array to an array of pointers to Item objects. Since I now had an array of pointers, I could initialize them to NULL when the List object is created.

Another problem I faced was that my program kept terminating with a segmentation fault error every time I called my displayList() function. This was when I realized that I was asking the system to print out all content in my List including cell with null pointer. Because nothing is in the cell pointing to null, the system could not access any Item class member function. Hence, the compiler terminated with a segmentation fault. To solve this problem, I added an if statement to check whether this array cell is pointing to an Item object. If the cell is pointing to an Item, then I will call the item's getter function to access Item's information. Adding the if statement after the for loop and before calling getter functions were able to resolved this error.

The same problem also occurred in my other functions involving a for loop to loop through all element in the array. Every time I tried to loop through the array, I received compilation errors. Upon further investigation, I realized that again, I was having problem looping through Null pointer. To solve this issue, I added an if statement after my for statement to make sure that all actions in the for loop only apply to element of array that is pointing to an Item object.

After my program compiled, I had to overload the == operator and take into account the adding of the same item twice. Because of this, I had to change my userAdd() function to check

whether the item user wants to add is already in the List before calling addItem() to add the item to the List. I also changed my userRemove() function to take into account if the item the user wants to remove is in the List. By comparing user's entered item's name, item's unit of sale and item's unit price to all Item in List before calling addItem() and removeItem().

Testing Results

Test	Input values	Driver function	Expected outcome	Test Results
1	Display starting menu	startMenu()	Starting menu displayed on screen.	Start menu displayed on screen.
2	Display starting menu, user chooses 4 (quit)	startMenu() intValidate()	Starting menu displayed on screen. User's choice is valid. Program terminates.	Start menu displayed on screen. Screen print out "Program terminating. Goodbye!" Program terminated.
3	Display starting menu, user chooses 5	startMenu() intValidate()	Starting menu displayed on screen. Invalid input. Prompt user to re-enter choices.	Start menu displayed on screen. Invalid input. Enter 1, 2, 3, or 4 only. Start menu displayed again prompting users for their choice.
4	Display starting menu, user chooses 1	startMenu() intValidate() userAdd() addItem()	Starting menu displayed on screen. Valid input. Prompt user for item's details. Add item to list	Start menu displayed. Enter item's name: Bread Enter item's unit of sale: Loaf Enter item's unit price: 2.99 Enter number needed: 1 Item successfully added to List.

5	Display starting menu, user chooses 2	startMenu() intValidate() userRemove() removeItem()	Starting menu displayed on screen. Valid input. Prompt user for item's details. Remove item from list	Start menu displayed. Enter item to be removed: Bread Enter item's unit of sale: Loaf Enter item's unit price: 2.99 Enter number to remove: 1 Item successfully removed from List
6	Array is full. User wants to add item to list	checkArray() expandArray() userAdd() addItem()	Return true. Create a new array with bigger size. Prompt user for item's details Add item to list	Successfully prompt user to enter item's information and add item to List.
7	User wants to display list	displayList() totalPrice()	Print out content of List and total price of all item in the list.	Content of List is printed out. Total cost of all items in List is printed out and calculated correctly.
8	User added bread to the list. User try to add \$2.99/loaf of bread again.	Operator==()	Print out "Item already in list" Prompt user whether they want to update the number of loaf of bread needed	Item already in cart. Do you want to update number of LOAF of BREAD to buy? Enter Y for Yes or N for No.
9	User added bread to the list. User wants to add the same bread again	Operator==() setNumberNeeded()	Print out "Item already in list" Prompt user whether they want to update the number needed Update number needed	Item already in cart. Do you want to update number of LOAF of BREAD to buy? Enter Y for Yes or N for No. Y

				Number of bread updated successfully and correctly.
10	User added bread to the list. User do not want to add the same bread again.	Operator==()	Print out "Item already in cart" Ask user whether they want to update number of bread needed. Number needed stay the same.	Item already in cart. Do you want to update number of LOAF of BREAD to buy? Enter Y for Yes or N for No. N Number of bread did not update.
11	User tries to remove item not in list	userRemove()	Prompt user for information of item to be removed. Print out "Item not in cart"	Successfully prompt user for information of item's to be removed. Print out "Item not in cart" successfully and Display content currently in List.