Jeffrey Schachtsick

CS 162: Assignment 2

Understanding, Testing, and Design Doc

April 19, 2015

Understanding

In this week's assignment, I will be creating a shopping list program. This program will ask the user to either display a list of items on their shopping list, to add items to the list, or remove items from the list. Each item will be entered in the list by the user with an item name, a unit of measure, number the user will buy, and the unit price. Once the item is created, it will be added to the list.

The user can also remove items from the list. The program will ask the user which item to remove. The program will then remove the item based on an identifier of the item such as item name.

In this program, the user will be able to enter a shopping club membership. The user can enter in new items to the list and be asked if the item has a club discount. If it does have a club discount, 10% will be taking off the total price, otherwise its regular price. A learning point for this week's assignment is Inheritance. The club item class will have inherited data from the Items class.

Testing

Test Name	Purpose	Input	Expected Output
Add a regular item to list or Non Club Member	Item should be added to list by user. Item displayed to user when display list is called.	1. User selects to Add an Item 2. Enter Item data: Item name is Cereal; unit is Box; Number to buy is 1; Unit price is 4.50	Item: Cereal Unit type: Box Units: 1 Price per unit $4.50 Total Price List $4.50
Add a regular item to list with more than one quantity	Item should be added to list by user. User adds more than 1 unit. Item displayed to user when display list is called and total price reflects correct multiplication.	1. User selects to Add an Item 2. Enter Item data: Item name is Cereal; unit is Box; Number to buy is 3; Unit price is 4.50	Item: Cereal Unit type: Box Units: 3 Price per unit $4.50 Total Price List $13.50
More than one item added to list	Multiple items should be added to list by user. List is displayed with all items entered into the list and total price reflects the list. Round fractional	1. User selects to Add an Item 2. Enter Item data: Item name is Cereal; unit is Box; Number to buy is 3; Unit price is 4.50	Item: Cereal Unit type: Box Units: 3 Price per unit $4.50 Item: Watermelon Unit type: Pounds Units: 4.5

	pennies to the nearest penny.	3. User selects to Add an Item 4. Enter Item data: Item name is Watermelon; unit is Pounds; Number of units is 4.5; Unit price is .25 5. User selects to Add an Item 6. Enter Item data: Item name is Beer; unit is Case; Number to buy is 1; Unit price is 14.99	Price per Unit: $.0.25 Item: Beer Unit type: Case Units: 1 Price per Unit: 14.99 Total Price $ 29.62
Remove item from list	Multiple items should be added to list by user. List is displayed with all items entered into the list and total price reflects the list. Remove an item from list. Verify item has been removed with displaying list. Ensure total price reflects what's on list. Round fractional pennies to the nearest penny.	1. User selects to Add an Item 2. Enter Item data: Item name is Cereal; unit is Box; Number to buy is 3; Unit price is 4.50 3. User selects to Add an Item 4. Enter Item data: Item name is Watermelon; unit is Pounds; Number of units is 4.5; Unit price is .25 5. User selects to Add an Item 6. Enter Item data: Item name is Beer; unit is Case; Number to buy is 1; Unit price is 14.99 7. Remove Item from list 8. User enters Cereal 9. Display list	Item: Cereal Unit type: Box Units: 3 Price per unit $4.50 Item: Watermelon Unit type: Pounds Units: 4.5 Price per Unit: $.0.25 Item: Beer Unit type: Case Units: 1 Price per Unit: 14.99 Total Price $ 29.62 Item: Watermelon Unit type: Pounds Units: 4.5 Price per Unit: $.0.25 Item: Beer Unit type: Case Units: 1 Price per Unit: 14.99 Total Price $ 16.12
Remove Item from list does not exist	Multiple items should be added to list by user. List is displayed with all items entered into the list and total price reflects the list. Remove an item from list that does not exist.	1. User selects to Add an Item 2. Enter Item data: Item name is Cereal; unit is Box; Number to buy is 3; Unit price is 4.50 3. User selects to Add an Item	Item: Cereal Unit type: Box Units: 3 Price per unit $4.50 Item: Watermelon Unit type: Pounds Units: 4.5 Price per Unit: $.0.25

	Verify NO item has been removed with displaying list. Ensure total price reflects what's on list. Round fractional pennies to the nearest penny.	4. Enter Item data: Item name is Watermelon; unit is Pounds; Number of units is 4.5; Unit price is .25 5. User selects to Add an Item 6. Enter Item data: Item name is Beer; unit is Case; Number to buy is 1; Unit price is 14.99 7. Remove Item from list 8. User enters Milk 9. Display list	Item: Beer Unit type: Case Units: 1 Price per Unit: 14.99 Total Price $ 29.62 The item 'Milk' does not exist! Please try again Item: Cereal Unit type: Box Units: 3 Price per unit $4.50 Item: Watermelon Unit type: Pounds Units: 4.5 Price per Unit: $.0.25 Item: Beer Unit type: Case Units: 1 Price per Unit: 14.99 Total Price $ 29.62
User enters Club membership and enters a Club item	User is asked if they are a Club member. They provide Member ID. User selects to add item to list. User is prompted if it's a club discount and says yes. Item should be added to list by user. Item displayed to user when display list is called and has 10% discount.	1. User enters a club membership 2. User selects to Add an Item 3. User selects this new item is on club discount 4. Enter Item data: Item name is Cereal; unit is Box; Number to buy is 1; Unit price is 4.50	Item: Cereal Unit type: Box Units: 1 Price per unit $4.50 Club Savings $0.45 Total Price List $4.05

Design

<u>List Class</u>

Data:

- vector <item> list
- vector <clubItem> list

Methods:

- Default Constructor()
  - vector <item> list[0]
- Constructor
- void displayList()
  - Go through each item or ClubItem in the list and display each and their data
  - Gather total of Shopping list
- void addItem(bool)
  - If isMember
    - Ask if item to be added is a club item (isClubItem)
  - Ask for Item name (string ItemName)
  - Ask for unit type (enum unitType)
  - Ask for number of units (double numUnits)
  - Ask for unit price (double unitPrice)
  - If isClubItem
    - Create Club Item to list with arguments itemName, unitType, numUnits, unitPrice
  - Else
    - Create Item to list with arguments itemName, unitType, numUnits, unitPrice
- void removeItem()
  - Ask user for Item Name
  - Go through each item compare the name to names on list
  - If name match
    - delete vector item from list
  - Else
    - Display name does not match

<u>Item Class</u>

Data:

- string item name
- enum unitType {Cans, Cases, Pounds, Ounces, Box, Dozen, Unit}
- double numberBought
- double unitPrice

Methods:

- Default Constructor()
- Constructor(name, unitType, numberBought, unitPrice)
- string getItemName()
- void setItemName(string)
- enum getUnitType()
- void setUnitType(enum)
- double getNumBought()
- void setNumBought(double)
- double getUnitPrice()
- void setUnitPrice(double)

ClubItem Class

Data:

- string item name
- enum unitType {Cans, Cases, Pounds, Ounces, Box, Dozen, Unit}
- double numberBought
- double unitPrice

Methods:

- Default Constructor()
- Constructor(name, unitType, numberBought, unitPrice)
- string getItemName()
- void setItemName(string)
- enum getUnitType()
- void setUnitType(enum)
- double getNumBought()
- void setNumBought(double)
- double getUnitPrice()
- void setUnitPrice(double)

Main
Program Start
Welcome message starting the program
Ask user if they are a Club Member

- Set boolean variable to flag Club Member (isMember)

Create List object
Do Loop
{
    Display Menu for user. 4 options to either display list, add item, remove item, or quit.
    User makes selection
    Switch
    Case: display list
        list.displayList
    Case: Add Item
        list.addItem(isMember)
    Case: Remove Item
        list.removeItem()
    Case: Quit
        set variable to Exiting Program

}While (!Exiting Program)
Exit the program