Assignment 2

Mike Peters

petermi3@oregonstate.edu

10/25/2015

Design

You will use a class for the **Items**. The class should have data elements for the following information: **item name, unit** (i.e. can, box, pounds, or ounces), **number to buy**, and **unit price**. Do you need any functions other than the constructor(s)? How do you calculate the **extended price** for the item (number to by times unit price? How do you **print it to the screen(display)**?

You will also need a **List** class. You will store Item objects in your List object. As each item is entered an Item object must created and added to the List object. One List object may have many Item objects. Do you need a print function in this class?

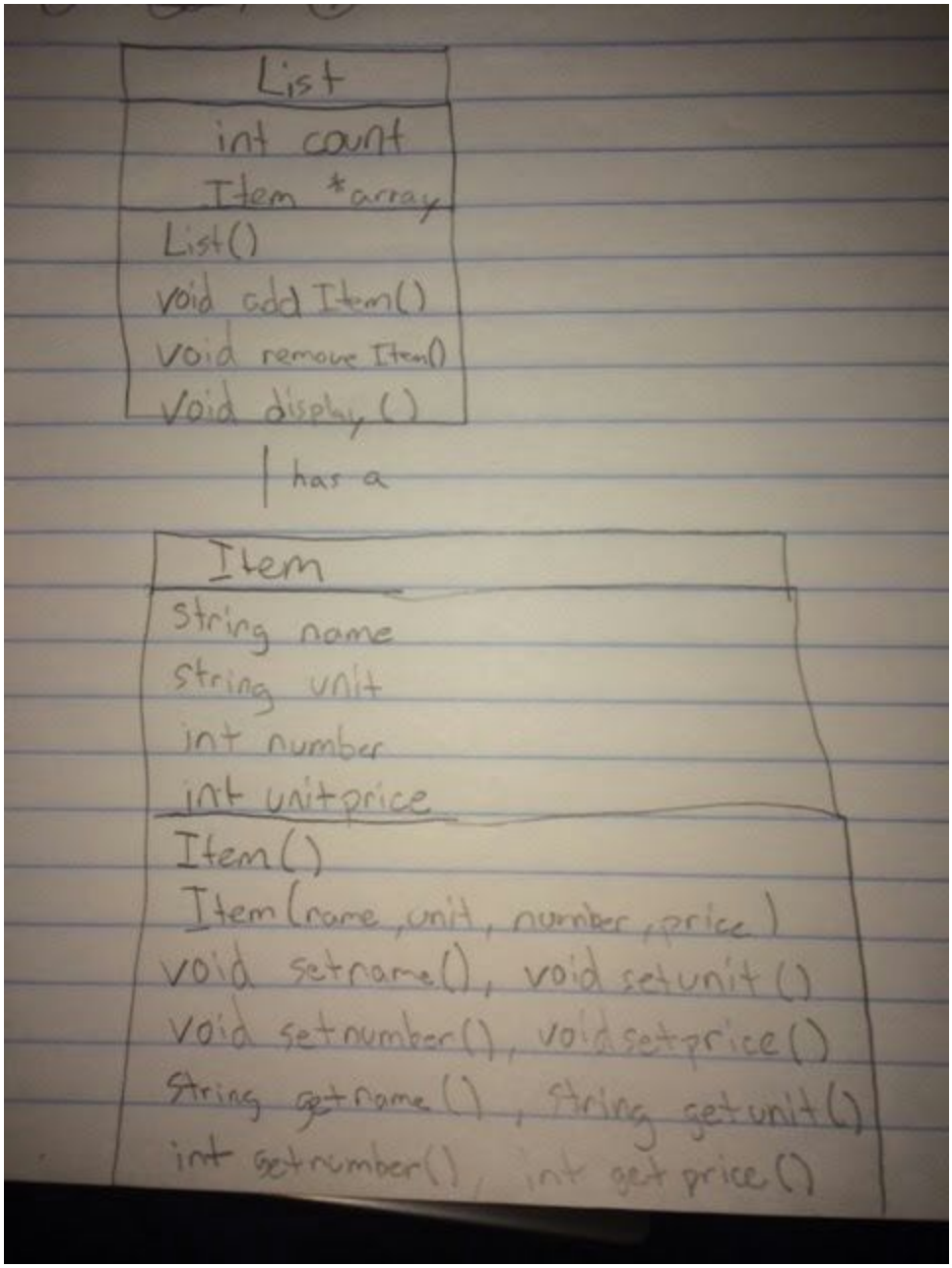
Your program must perform the following activities: **create a list, add items, remove items, and display the shopping list**. To add an item you should prompt the user to enter the **name, unit of sale**, the **number needed**, and the **unit price**. The **display** should show: each **item** in the list, the **number of items**, the **unit of sale**, the **unit price**, the **extended price** for each item, and the **total price** for all items.

Once you have the **List** and **Item** classes working correctly, test if an item is already in your List before adding it. **Overload the == operator** to perform the test. There is a simple example to overload this operator in the book. Keep it simple. How will you compare items? You can assume that the user will type the information in correctly.

Use the list class to create Item objects stored in a list object array. Constructors/get/set functions are all that should be needed for item class. List class will have add, remove, and display functions. A loop will be used to print the contents of the list array. Size of array will need to be “counted” to know how many items there are to print.

Test Plan

Test Case	Input Values	Expected Outcomes
Item name with spaces	black olives	Item = black olives
Item with no spaces	tomatoes	Item = tomatoes
Add multiple items	black olives and tomatoes	Items = black olives and tomatoes
Remove an Item	remove black olives from above list	Items = tomatoesl
Decimal unit price	3.5	Unit price = 3.5
Unit price and quantity	3.5 and 3	extended price = 10.5
Total price	3 black olives @ 3.5 4 tomatoes @ 2	total price = 18.5
Total price after removing an item	same as above then remove black olives	total price = 8


List
int count
Item *array
List()
void addItem()
void removeItem()
void display()
has a
Item
String name
String unit
int number
int unitprice
Item()
Item(name, unit, number, price)
void setname(), void setunit()
void setnumber(), void setprice()
String getname(), String getunit()
int getnumber(), int getprice()

Debug and design changes

First problem I encountered was simply using cin was not going to work for item names with spaces. This was solved by using getline(cin, name). However the menu would not pause to allow user input at proper time. I had to go back and add cin.ignore() to the input and output of the string name. I added a few variables that I didn't think of when planning my program. I also forgot to include the parameters of my set functions in planning. Other than minor spelling and syntax errors the test plan worked as expected. Previously I had always used if loops for my menu selection, however after some reading I noticed a lot of comments saying a switch is better practice. I tried that this time and it seems to work just fine. The code definitely looks better and seems easier to follow.