Tatyana Vlaskin
Assignment #2

Design:

This assignment does not make much sense. If I were to make a shopping list, I would not know the price of the product. Also, I do not know the sales price. At the same time, there are at least 4 dozen different stores out there, and all stores have different prices and all stores have different sales. So, the whole assignment does not make such sense. I am sorry for venting.

So I am designing very simple program:

I'll make a class:

```
class Item{
private:
        string name, location;
        double regularPrice, salePrice;
        int quantity;
public:
        Item(string name, string location, double regularPrice, double salePrice, int quantity)
        {
                this->name= name;
                this->location = location;
                this->regularPrice = regularPrice;
                this->salePrice = salePrice;
                this->quantity = quantity;
        }
```

At the beginning of the program the user will be asked to provide a name of the file, which they want to open. This file will contain the list of product that was saved from the previous use of the program

If the user entered incorrect name/ file that does not exist, the user will be asked to reenter the file name.

As the file opens, this will be the current list of the products.

After user will be presented with the menu:

```
                cout << "--Please select option --" << endl;
                cout << "1. Add Item to list" << endl;
                cout << "2. Delete Item from list" << endl;
                cout << "3. Display the list" << endl;
                cout << "4. Checkout" << endl;
```

Depending on the option that the user chooses,they will be taken to the switch loop:

```
    case 1:
            addProduct(groceryList);
            break;
        case 2:
            deleteProduct(groceryList);
            break;
        case 3:
            displayList(groceryList);
            break;
        case 4:
            checkout(groceryList);
            break;
        default:
            cout << "Invalid Option selected" << endl;
            break;
        }
```

Add function will look like this:

void addProduct(vector<Item>& groceryList)

```
{
    User will be asked for the name of the product
    User will be asked for the location of the product
    At this point the user will see a list of possible locations that can be chosen
    Please note that the user will need to enter location in capital letters
    The user will be in te do/while loop until correct location is enterd
    List of possible locations will be stored in the array:
    string phrase_array[14]={"MEAT","DAIRY","CONDIMENTS","SWEETS",
         "PRODUCE","GRAINS","HEALTH","BEVERAGES","HYGIENE",
         "FOZEN","BAREKY","BABY", "HOUSEHOLD",
         " MISALENEOUS"};
    Once the user enters all information, item will be added to the vector:
    groceryList.push_back(Item(name, location, regularPrice, salePrice,
quantity));
}
```

deleteProduct(groceryList);

If the user choses this option, they will be presented with the list of good that are in their basket
and which they can delete
The user needs to type a name of the product that they want to delete
The name entered by the user will be compared to all names in the vector and if the name matches, the product will be deleted. If the name does not exist, the user will get a message that product does not exist and will be taken to the main menu. Something like this:

```
        for(vector<Item>::iterator it = groceryList.begin(); it != groceryList.end(); ++it)
        {
                if(it->getName().compare(pItem) == 0)
      {
         groceryList.erase(it);
         cout << "Item deleted successfully" << endl;
         return;
      }
```

displayList(groceryList);

So we need to display list by location. At this point I have no idea how to do that. I need to sort vector someone, Ill do some additional reading during the implementation process and I hope that Ill be able to come up with a way to sort a vector by location. If not, then, Ill simply display all products that were added to the list, using something like this:

```
        for(vector<Item>::iterator it = groceryList.begin(); it != groceryList.end(); ++it)
        {
                cout << "Product Location: " << it->getLocation() << endl;
                cout << "Product Name: " << it->getName() << endl;
                //cout << "Product Location: " << it->getLocation() << endl;
                cout << "Product Quantity: " << it->getQuantity() << endl;
        }
```

Finally if the user choses:

void checkout(vector<Item>& groceryList)

A user will be presented with the list of items that they added on the list along with the prices, total balance of the purchase and total savings.

At the same time, if the user choses this option, all the items from the list, will be exported to the text document that will be saved in the same location as the .cpp file. This text document will be used in case the user decides to edit the list in the future.

Testing:

What are we testing	**How we are testing**	**What is expected**	**What is the output**	**PASS/FAIL**
Program complies	Compile the program	No error	NA	Pass
If the user enters number HIGHER THAN 4 while selecting an option at the menu, an invalid entry will be displayed	Chose number 5	Invalid Option Selected message	Invalid Option Selected message	Pass
If the user enters invalid type of data, an invalid type message is displayed	Enter mkflkdsjflkf	That is NOT an integer message…	That is not an integer message	Pass
User choses # 1 on the menu	1	User is asked to enter product name	User is asked to enter product name	Pass
The user is able to enter product name	No restriction – enter whatever you want, but	Entry is accepted and User is asked to	Entry is accepted and User is asked to enter location of the	Pass

	remember if you decide to remove this product form the list, you’ll need to type the same thing, so do not use words that are too long. Enter MILK for example	enter location of the product	product	
Enter the location in the store. Only locations provided on the list are accepted	User enters location provided on the list. Remember, you need to enter location in capital letters. Enter MEAT	User is prompted to enter price	User is prompted to enter price	Pass
Test and make sure that all locations on the list work	One by one test that all locations work.	User is asked to enter price	User is asked to enter price	Pass
If the user enters location that is not on the list, the program waits till the correct location is entered	Type ice, or Whatever	Nothing happens, program waits will correct location is entered	Nothing happens, program waits will correct location is entered	Pass
User enters string for price, invalid message is	Fkdjfslk	Invalid entry message	Invalid entry message	Pass

displayed				
User enters a float/int for price	5 or 0.9 whatever	User is prompted to enter sales price	User is prompted to enter sales price	Pass
User enters string for sales price and invalid message is displayed	Dfskdf;l	Invalid entry message	Invalid entry message	Pass
User enters double or int for sales price, price is accepted	10	Item is added to the list. User is taken back to the menu, to make another selection	Item is added to the list. User is taken back to the menu, to make another selection	Pass
User choses option # 3 to display the items on the list	3	All items that were added are displayed. Please note, at this point I items will not be displayed by location, Ill work on this later	All items that were added are displayed	pass
User choses an option #2 to delete an item from the list	2	User is presented with the list of items	User is presented with the list of items that can be	Pass

		that can be deleted	entered	
The user types an item that is not on the list to be deleted	Type an item that is not on the list	Message that invalid item was entered and user is taken back to the main menu	Message that invalid item was entered and user is taken back to the main menu	Pass
The user types an item that they want to delete and that is on the list	Type the item that is on the list	Message saying that item was removed successfully and the user is taken back to the menu	Message saying that item was removed successfully and the user is taken back to the menu	Pass
User choses option 4 from the list to checkout	4	The list is displayed along with the total and how much money was saved	The list is displayed along with the total and how much money was saved	pass

Reflections: Overall program was hard, I do not like designing because it does not stop, you modify one thing and another one comes up and the process never stops. I do understand that during the real job, this is necessary, but we are just learning all these concepts, I feel that its much better to give us a set of requirements and check that we can implement them.
So I am still trying to figure out how to sort my class of vectors by location.
After reading information on the following websites;
http://stackoverflow.com/questions/1380463/sorting-a-vector-of-custom-objects

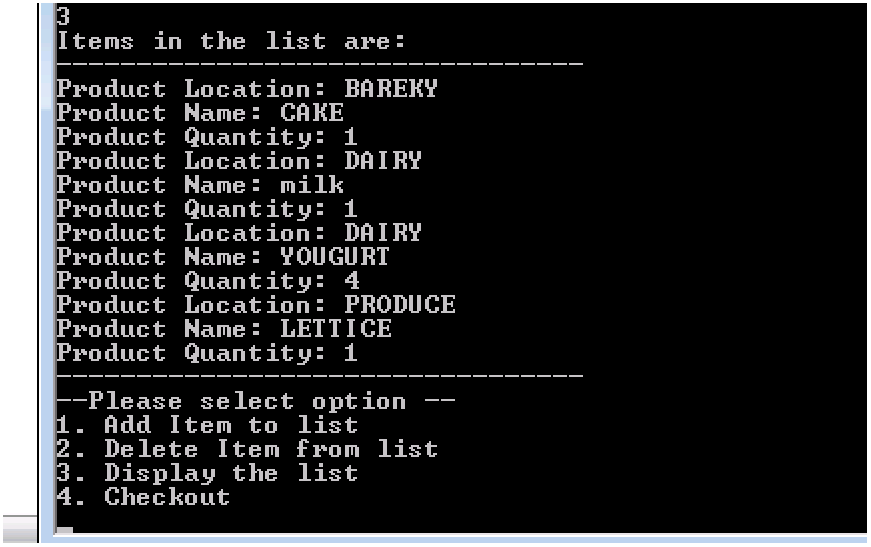
http://stackoverflow.com/questions/5635909/c-how-to-sort-vectorclass-with-operator
http://forums.codeguru.com/showthread.php?366064-STL-Sorting-How-to-sort-a-std-vector-containing-classes-structures
http://www.daniweb.com/software-development/cpp/threads/256019/sorting-objects-by-class-member-variables
I came up with the sort function:
bool myfunction (Item i, Item j) { return (i.getLocation() < j.getLocation());}
So when the user wants to display the list, I call the sort function first and only than display the list:

```
        case 3:
         sort(groceryList.begin(),groceryList.end(), myfunction);
                         displayList(groceryList);
                         break;
```

At the same time, when the user is ready to checkout, I call the same function as well:

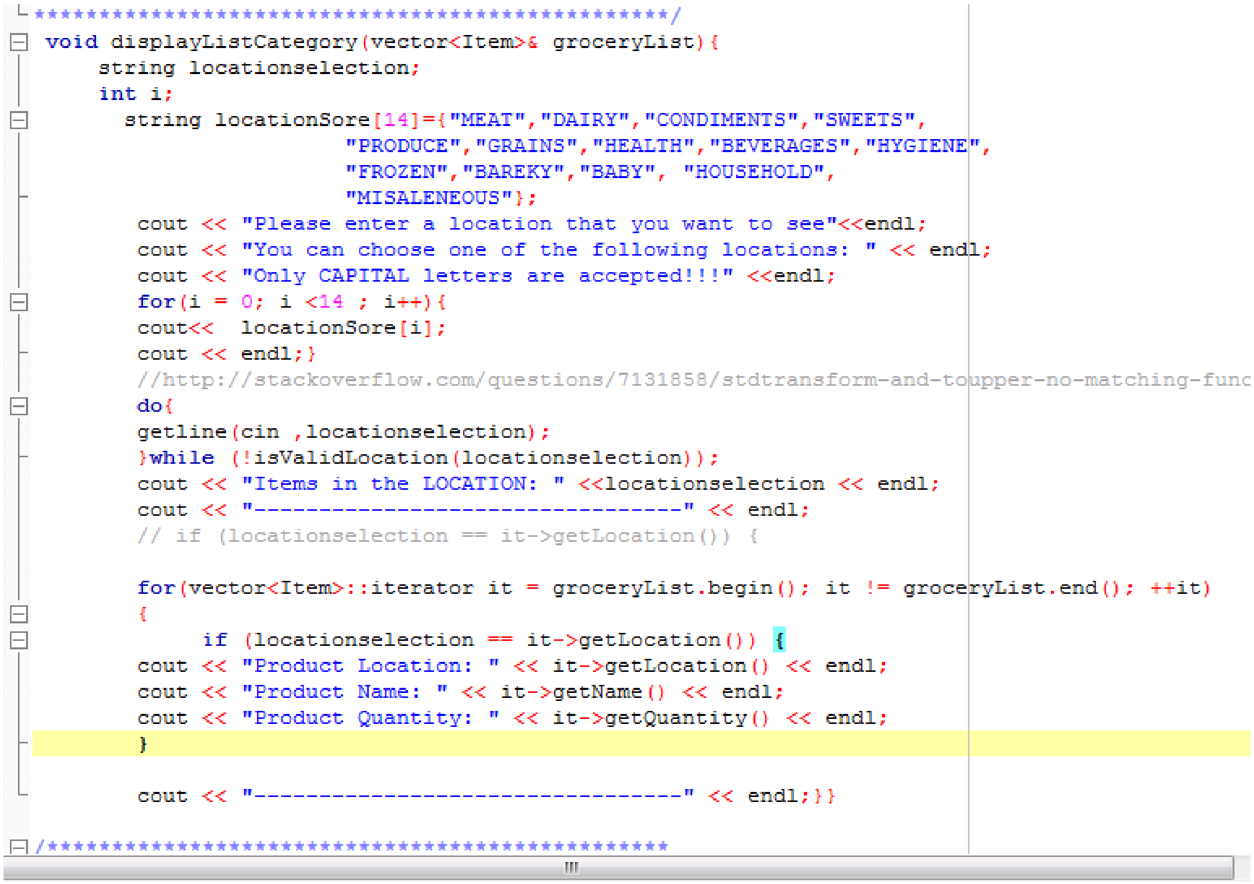
```
        case 4:
         sort(groceryList.begin(),groceryList.end(),myfunction);
                         checkout(groceryList);
                         break;
```

It works, the only limitation is that, the location is displayed multiple times, but all items in the case category are grouped together:



```
3
Items in the list are:
---------------------------------
Product Location: BAREKY
Product Name: CAKE
Product Quantity: 1
Product Location: DAIRY
Product Name: milk
Product Quantity: 1
Product Location: DAIRY
Product Name: YOUGURT
Product Quantity: 4
Product Location: PRODUCE
Product Name: LETTICE
Product Quantity: 1
---------------------------------
--Please select option --
1. Add Item to list
2. Delete Item from list
3. Display the list
4. Checkout
```

As you can see products are sorted alphabetically, but location is displayed more than once. I am running low on time, so this will NOT be fixed. I even do not understand if that’s what we had to do. That’s how I interpreted it. However, after thinking I decided maybe that’s not what we need to do, so I added one more function to my program. This will be part of the menu. If the user what to see items in some specific location, the user will be asked to enter location and all the items in that location will be displayed. The function provided below. Maybe that was what we need to do? I do not know. At this point I think that I really deserve those last 5 points of the grade, for displaying output by location. I wish I knew what it means.



```
****************************************************/
void displayListCategory(vector<Item>& groceryList){
    string locationselection;
    int i;
      string locationSore[14]={"MEAT","DAIRY","CONDIMENTS","SWEETS",
                         "PRODUCE","GRAINS","HEALTH","BEVERAGES","HYGIENE",
                         "FROZEN","BAREKY","BABY", "HOUSEHOLD",
                         "MISALENEOUS"};
       cout << "Please enter a location that you want to see"<<endl;
       cout << "You can choose one of the following locations: " << endl;
       cout << "Only CAPITAL letters are accepted!!!" <<endl;
       for(i = 0; i <14 ; i++){
       cout<<  locationSore[i];
       cout << endl;}
       //http://stackoverflow.com/questions/7131858/stdtransform-and-toupper-no-matching-func
       do{
       getline(cin ,locationselection);
       }while (!isValidLocation(locationselection));
       cout << "Items in the LOCATION: " <<locationselection << endl;
       cout << "----------------------------------" << endl;
       // if (locationselection == it->getLocation()) {

       for(vector<Item>::iterator it = groceryList.begin(); it != groceryList.end(); ++it)
       {
            if (locationselection == it->getLocation()) {
       cout << "Product Location: " << it->getLocation() << endl;
       cout << "Product Name: " << it->getName() << endl;
       cout << "Product Quantity: " << it->getQuantity() << endl;
       }

       cout << "----------------------------------" << endl;}}

/*************************************************
```

Another problem that I've noticed during testing is that the program accepts sales prices that are higher than the regular prices, this can be fixed by imposing a restriction that if the sale price is higher than the regular price, the user needs to reenter price, but as I pointed out I am running low on time, so this will not be fixed

Another problem that I have is when you chose an option to delete an item from the list and there are no items on the list, you are still prompted to make an entry. A fix to this can be a message no items in the cart, which can be accomplished by checking the size of the vector and is the size id 0, display the message. This will not be fixed due to lack of time.

Another limitation of the program is that the user needs to type location instead of choosing an option, which chose 1 for Meat section, 2 for Dairy Section, etc.

Finally, on the last assignment I lost 4 points because I had global variable. I decided not to use any global variable on this assignment. However, as a result of this, I declared the following array several times in my code:

```
 string locationSore[14]={"MEAT","DAIRY","CONDIMENTS","SWEETS",
                    "PRODUCE","GRAINS","HEALTH","BEVERAGES","HYGIENE",
                    "FROZEN","BAREKY","BABY", "HOUSEHOLD",
                    "MISALENEOUS"};
```

Where should I declare a variable like this, which I want to be visible everywhere? I need to do additional reading on this.

Other than that program works ok, and I think that all requirements were met.