

CSIS-1175 – Assignment# 2

Due: Mon, Dec 5, 11:59 PM - Total Points: 50

Submission guidelines: Please perform the following tasks. **Add comments** to your code explaining the variables and logic used. Not adding comments to each program and file you are submitting explaining the logic will result in loss of points for each question.

Create the C# project with your FirstNameLastInitial_Assign2 - **There should be NO SPACES in your PROJECT NAME. You must use upper and lower cases for the project name. (e.g., If your name is **John Smith**, then your project must be called **JohnS_Assign2**).**

At the beginning of your source code, you must again enter your section #, your name and student id as a comment.

Zip up your entire program from the top folder and submit it via Blackboard before the due date. **Do not rename the folder. You will be responsible for accurately compressing the entire project folder (along with solution files) and submitting it on blackboard.** Late submission will not be accepted. The name of your zip file must be the same as your project name. If you do not follow the above instructions, including the file name and subject line, you will get marks deducted.

Note: Sharing your code, and looking at another student's code will constitute to cheating. In case of cheating both the student that copies and the student that shared or allowed the copying would get Zero and will reported to administration.

Task: In this assignment, you are required to implement a grocery management application in C#. Grocery Data (see given csv file, superstore.csv) contains ItemCode, ItemName, UnitPrice, QtyMinForRestock (minimum quantity needed for restocking the item), InitialQty (initial quantity count for each item), QtySold, QtyRestocked.

Class Definition: You must use a Grocery class to maintain a grocery item with Properties:

- ItemCode, ItemName, QtyMinForRestock, InitialQty, QtySold, QtyRestocked.
- You will use two additional computed read-only properties: i. AvailableQty, which is computed as the Initial quantity – Quantity Sold + Restocked Quantity, and ii. Total Sales for each item based on QtySold multiplied by UnitPrice. These two additional properties of TotalSales and AvailableQty will be read-only properties, and cannot be directly set at any point. They will be updated to reflect the correct amount whenever you try to get these properties.
- You will need to define constructors and ToString() methods as needed

You will use a list of Grocery Items class objects to maintain all grocery items in the list (input file). Feel free to create additional fields or class methods as needed.

Form Design: Design the App with the following design features (see figure for sample). Your form should contain:

1. A grocery listbox that will display ItemCode, ItemName, InitialQty, QtyMinForRestock, QtySold, QtyRestocked, UnitPrice, AvailableQty and Sales. Set the listbox font to Consolas (or any other monospaced font).
2. Below the list box, you will have a set of controls in three group boxes for loading data, updating data, saving data. On the left, you have the Load Grocery Button in the Load Data GroupBox. The Update Data group box contains one set (of label, TextBox and Button) for incrementing Qty Sold, and another set (of label, TextBox and Button) for incrementing Qty Restocked. Then, there is a Delete Selected Item Button and Sort Items By Sales Button. The Save Data group box contains three buttons one each to Save Grocery Data, Save Sales Report and Save Restock Needs Report.
3. Below all these controls, add a status label that will display status information as various user events are handled. You can set the AutoSize property of the label to false (so that the label can be resized). You can set the BackColor property (background color) to any color at the Web tab. You can set the TextAlign Property to Center. You can also set the border style to Fixed 3D (to give it an embossed look).
4. Any other form layout or arrangement is fine as long as all this information is there in a clear format.

Event Handlers:

1. **Load Grocery Data:** When the Load Grocery button is clicked, grocery records will be read from a text file (superstore.csv) from the default folder, and the lines read from the file is tokenized into field values, parsed (as needed) and then loaded onto the listbox and the list of class objects. In addition to all the field values in the input file, the listbox should, also display columns for AvailableQty and Sales. Do some background research to see how String.Format() or PadRight() method can be used to format the listbox

contents such that all values in a column are aligned. Make sure your listbox has **monospaced font**. Finally, update the status label to indicate how many grocery items were loaded into the list.

Note that each item in listbox should always correspond to each item in the List of grocery items except for the extra header line at the top in the listbox but not in the List of Grocery Items. Every i th item in the listbox will thus correspond to $(i-1)$ th item in the list of Grocery objects.

Feel free to use methods as you see fit.

ItemId	ItemName	UnitPrice	QtyInForRestock	InitialQty	QtySold	QtyRestocked	AvailableQty	Sales
11-550-3026	Shampoo	\$8.83	13	31	0	0	31	\$0.00
36-389-7424	Conditioner	\$8.72	10	60	0	0	60	\$0.00
64-098-4668	Dandruff Shampoo	\$8.09	13	66	0	0	66	\$0.00
29-054-1549	Body wash	\$9.01	6	34	0	0	34	\$0.00
76-866-0130	Bar Soap	\$8.76	14	78	0	0	78	\$0.00
12-575-1269	Hand Soap	\$6.43	12	61	0	0	61	\$0.00
51-854-4250	Razor Blades	\$3.96	14	97	0	0	97	\$0.00
38-416-2598	Shaving Cream	\$4.04	10	41	0	0	41	\$0.00
00-499-3472	Feminine Products	\$3.64	5	95	0	0	95	\$0.00
90-222-4482	women's Deodorant	\$3.56	20	19	0	0	19	\$0.00
71-288-8443	Men's Deodorant	\$6.32	14	12	0	0	12	\$0.00
36-559-4086	Toothpaste	\$3.04	7	52	0	0	52	\$0.00
13-363-4031	Toothbrushes	\$2.37	6	85	0	0	85	\$0.00
14-459-7070	Floss	\$3.67	11	50	0	0	50	\$0.00
88-225-9449	Lotion	\$2.64	19	85	0	0	85	\$0.00

2. **Update Sold Qty For Selected Item:** When this button is clicked, and if no item or if the headerline is selected from the listbox, you should display an error message in the status label indicating that *“Please selected a grocery item to increment sold qty”*. If a grocery item is selected from the listbox, then TryParse the sold Qty from the QtySold TextBox. If the TryParse fails or if the Qty Sold from TextBox input is less than or equal to 0 or if QtySold from TextBox is greater than Available Qty, then output relevant error messages in the status label. If the parse succeeds, and if the Qty Sold from the TextBox is greater than 0 and if Qty Sold from TextBox is lesser than Available Qty for that item, then increment the

QtySold field by that amount given for the corresponding grocery item from the list. Available Qty should also reflect this change. Display a message in the status label saying *“Sold Qty has been increased for the item with item code ABC”*. See sample below.

Form1

Super Store Grocery Management System

ItemId	ItemName	UnitPrice	QtyMinForRestock	InitialQty	QtySold	QtyRestocked	AvailableQty	Sales
11-550-3026	Shampoo	\$8.83	13	31	0	0	31	\$0.00
36-389-7424	Conditioner	\$8.72	10	60	0	0	60	\$0.00
64-090-4668	Dandruff Shampoo	\$8.09	13	66	0	0	66	\$0.00
29-054-1549	Body Wash	\$9.01	6	34	0	0	34	\$0.00
76-866-0130	Hand Soap	\$6.43	12	61	0	0	61	\$0.00
12-575-1269	Hand Soap	\$6.43	12	61	0	0	61	\$0.00
51-854-4250	Razor Blades	\$3.96	14	97	0	0	97	\$0.00
38-416-2990	Shaving Cream	\$4.04	10	41	20	0	21	\$80.00
00-499-9472	Feminine Products	\$3.64	5	95	0	0	95	\$0.00
98-222-4482	Women's Deodorant	\$3.56	20	19	0	0	19	\$0.00
71-280-8443	Men's Deodorant	\$6.32	14	12	0	30	42	\$0.00
36-559-4086	Toothpaste	\$3.04	7	52	0	0	52	\$0.00
23-363-4831	Toothbrushes	\$2.37	6	85	0	0	85	\$0.00
14-459-7070	Floss	\$3.67	11	50	0	0	50	\$0.00
88-225-9449	Lotion	\$2.64	19	85	0	0	85	\$0.00

Load Data

Load Grocery Data

Update Data

Quantity Sold:

Quantity Restocked:

Update Sold Qty For Selected Item

Update Restocked Qty For Selected Item

Delete Selected Grocery Item

Sort Items Based on Sales

Save Data

Save Grocery Data

Save Sales Report

Save Restock Needs Report

Incremented Restocked Qty for item with Item Code 71-280-8443

Form1

Super Store Grocery Management System

ItemId	ItemName	UnitPrice	QtyMinForRestock	InitialQty	QtySold	QtyRestocked	AvailableQty	Sales
11-550-3026	Shampoo	\$8.83	13	31	0	0	31	\$0.00
36-389-7424	Conditioner	\$8.72	10	60	0	0	60	\$0.00
64-090-4668	Dandruff Shampoo	\$8.09	13	66	0	0	66	\$0.00
29-054-1549	Body Wash	\$9.01	6	34	0	0	34	\$0.00
12-575-1269	Hand Soap	\$6.43	12	61	0	0	61	\$0.00
51-854-4250	Razor Blades	\$3.96	14	97	0	0	97	\$0.00
38-416-2990	Shaving Cream	\$4.04	10	41	20	0	21	\$80.00
00-499-9472	Feminine Products	\$3.64	5	95	0	0	95	\$0.00
98-222-4482	Women's Deodorant	\$3.56	20	19	0	0	19	\$0.00
71-280-8443	Men's Deodorant	\$6.32	14	12	0	30	42	\$0.00
36-559-4086	Toothpaste	\$3.04	7	52	0	0	52	\$0.00
23-363-4831	Toothbrushes	\$2.37	6	85	0	0	85	\$0.00
14-459-7070	Floss	\$3.67	11	50	0	0	50	\$0.00
88-225-9449	Lotion	\$2.64	19	85	0	0	85	\$0.00
24-443-3243	Excema Cream	\$3.14	6	104	0	0	104	\$0.00

Load Data

Load Grocery Data

Update Data

Quantity Sold:

Quantity Restocked:

Update Sold Qty For Selected Item

Update Restocked Qty For Selected Item

Delete Selected Grocery Item

Sort Items Based on Sales

Save Data

Save Grocery Data

Save Sales Report

Save Restock Needs Report

Deleted record for item with item code 76-866-0130

5. **Sort Items Based On Sales:** When this button is clicked, sort the items in the list and listbox based on sales (see pic given below).

Form1

Super Store Grocery Management System

ItemId	ItemName	UnitPrice	QtyMinForRestock	InitialQty	QtySold	QtyRestocked	AvailableQty	Sales
38-416-2990	Shaving Cream	\$4.04	10	41	20	0	21	\$80.00
11-550-3026	Shampoo	\$8.83	13	31	0	0	31	\$0.00
36-389-7424	Conditioner	\$8.72	10	60	0	0	60	\$0.00
64-090-4668	Dandruff Shampoo	\$8.09	13	66	0	0	66	\$0.00
29-054-1549	Body Wash	\$9.01	6	34	0	0	34	\$0.00
12-575-1269	Hand Soap	\$6.43	12	61	0	0	61	\$0.00
51-854-4250	Razor Blades	\$3.96	14	97	0	0	97	\$0.00
00-499-9472	Feminine Products	\$3.64	5	95	0	0	95	\$0.00
98-222-4482	Women's Deodorant	\$3.56	20	19	0	0	19	\$0.00
71-280-8443	Men's Deodorant	\$6.32	14	12	0	30	42	\$0.00
36-559-4086	Toothpaste	\$3.04	7	52	0	0	52	\$0.00
23-363-4831	Toothbrushes	\$2.37	6	85	0	0	85	\$0.00
14-459-7070	Floss	\$3.67	11	50	0	0	50	\$0.00
88-225-9449	Lotion	\$2.64	19	85	0	0	85	\$0.00
24-443-3243	Excema Cream	\$3.14	6	104	0	0	104	\$0.00

Load Data

Load Grocery Data

Update Data

Quantity Sold:

Quantity Restocked:

Update Sold Qty For Selected Item

Update Restocked Qty For Selected Item

Delete Selected Grocery Item

Sort Items Based on Sales

Save Data

Save Grocery Data

Save Sales Report

Save Restock Needs Report

Sorted All Items in Descending order of Sales

6. **Save Grocery Data (updatedgrocery.csv):** When the user clicks on this button, **for those items that are still in the list box (and the list of grocery objects)**, write the ItemCode, ItemName, InitialQty, QtyMinForRestock, QtySold, QtyRestocked, UnitPrice onto a comma-separated text file in the default folder location using file name updatedgrocery.csv. The comma separated text file will have each column separated by a comma. The file should have a header row at the top that gives the name of each field. Finally, the status label should display log messages indicating how many grocery items were written to the file (see sample).

The screenshot shows the 'Super Store Grocery Management System' application window. It features a table with 10 columns: ItemId, ItemName, UnitPrice, QtyMinForRestock, InitialQty, QtySold, QtyRestocked, AvailableQty, and Sales. Below the table are three main functional areas: 'Load Data' with a 'Load Grocery Data' button; 'Update Data' with input fields for 'Quantity Sold' and 'Quantity Restocked', and buttons for 'Update Sold Qty For Selected Item', 'Update Restocked Qty For Selected Item', 'Delete Selected Grocery Item', and 'Sort Items Based on Sales'; and 'Save Data' with buttons for 'Save Grocery Data', 'Save Sales Report', and 'Save Restock Needs Report'. A status bar at the bottom indicates 'Saved 48 records into the output inventory file'.

Below the application window, a Notepad window titled 'updatedgrocery - Notepad' displays the saved data in a comma-separated format. The data includes a header row and multiple rows of item details.

ItemCode	ItemName	UnitPrice	QtyMinForRestock	InitialQty	QtySold	QtyRestocked
38-416-2990	Shaving Cream	4.04	41	10	20	0
11-550-3026	Shampoo	8.83	31	13	0	0
36-389-7424	Conditioner	8.72	60	10	0	0
64-090-4668	Dandruff Shampoo	8.09	66	13	0	0
29-054-1549	Body Wash	9.01	34	6	0	0
12-575-1269	Hand Soap	6.43	61	12	0	0
51-854-4250	Razor Blades	3.96	97	14	0	0
00-499-9472	Feminine Products	3.64	95	5	0	0
90-222-4482	Women's Deodorant	3.56	19	20	0	0
71-280-8443	Men's Deodorant	6.32	12	14	0	30
36-559-4086	Toothpaste	3.04	52	7	0	0
23-363-4831	Toothbrushes	2.37	85	6	0	0
14-459-7070	Floss	3.67	50	11	0	0
88-225-9449	Lotion	2.64	85	19	0	0
24-443-3243	Eczema Cream	3.14	104	6	0	0

7. **Save Sales Report (grocerysales.csv):** When the user clicks on this button, **use a LINQ query to find those items have QtySold greater than 0 from the list of grocery items.** Then, **for each item in the resulting list from the LINQ query**, write the ItemCode, ItemName QtySold, UnitPrice and Sales onto a comma-separated text file in the default folder location using the name grocerysales.csv. The comma separated text file will have each column separated by a comma. The file should have a header row at the top that gives the name of each field. Finally, the status label should display log messages indicating how many sale items were written to the sales report file (see sample).

Form1

Super Store Grocery Management System

ItemId	ItemName	UnitPrice	QtyMinForRestock	InitialQty	QtySold	QtyRestocked	AvailableQty	Sales
38-416-2990	Shaving Cream	\$4.04	10	41	20	0	21	\$80.80
11-550-3026	Shampoo	\$8.83	13	31	0	0	31	\$0.00
36-389-7424	Conditioner	\$8.72	10	60	0	0	60	\$0.00
64-090-4668	Dandruff Shampoo	\$8.09	13	66	0	0	66	\$0.00
29-054-1549	Body Wash	\$9.01	6	34	0	0	34	\$0.00
12-575-1269	Hand Soap	\$6.43	12	61	0	0	61	\$0.00
51-054-4250	Razor Blades	\$3.96	14	97	0	0	97	\$0.00
00-499-9472	Feminine Products	\$3.64	5	95	0	0	95	\$0.00
90-222-4482	Women's Deodorant	\$3.56	20	19	0	0	19	\$0.00
71-280-8443	Men's Deodorant	\$6.32	14	12	0	30	42	\$0.00
36-559-4086	Toothpaste	\$3.04	7	52	0	0	52	\$0.00
23-363-4831	Toothbrushes	\$2.37	6	85	0	0	85	\$0.00
14-459-7070	Floss	\$3.67	11	50	0	0	50	\$0.00
88-225-9449	Lotion	\$2.64	19	85	0	0	85	\$0.00
24-443-3243	Eczema Cream	\$3.14	6	104	0	0	104	\$0.00

Load Data

Load Grocery Data

Update Data

Quantity Sold:

Update Sold Qty For Selected Item

Quantity Restocked:

Update Restocked Qty For Selected Item

Delete Selected Grocery Item

Sort Items Based on Sales

Save Data

Save Grocery Data

Save Sales Report

Save Restock Needs Report

Saved 1 records into the output sales file

grocerysales - Notepad

File Edit Format View Help

ItemCode,ItemName,UnitPrice,QtySold,Sales
38-416-2990,Shaving Cream,\$4.04,20,\$80.80

Ln 1, Col 1100%Windows (CRLF)UTF-8

8. **Save Restock Needs Report (groceryrestocks.csv):** When the user clicks on this button, use a LINQ query to find those items that have AvailableQty lesser than QtyMinForRestock from the list. Then, for each item in the resulting list from the LINQ query, write the ItemCode, ItemName, QtyMinForRestock, AvailableQty onto a comma-separated text file in the default folder location using the file name groceryrestocks.csv. The comma separated text file will have each column separated by a comma. The file should have a header row at the top that gives the name of each field. Finally, the status label should display log messages indicating how many restock needed items were written to the restock needs report file (see sample).

Form1

Super Store Grocery Management System

ItemId	ItemName	unitPrice	QtyMinForRestock	InitialQty	QtySold	QtyRestocked	AvailableQty	Sales
38-416-2990	Shaving Cream	\$4.04	10	41	20	0	21	\$80.80
11-550-3026	Shampoo	\$8.83	13	31	0	0	31	\$0.00
36-389-7424	Conditioner	\$8.72	10	60	0	0	60	\$0.00
64-090-4668	Dandruff Shampoo	\$8.09	13	66	0	0	66	\$0.00
29-054-1549	Body Wash	\$9.01	6	34	0	0	34	\$0.00
12-575-1269	Hand Soap	\$6.43	12	61	0	0	61	\$0.00
51-854-4250	Razor Blades	\$3.96	14	97	0	0	97	\$0.00
00-499-9472	Feminine Products	\$3.64	5	95	0	0	95	\$0.00
90-222-4482	Women's Deodorant	\$3.56	20	19	0	0	19	\$0.00
71-280-8443	Men's Deodorant	\$6.32	14	12	0	30	42	\$0.00
36-559-4086	Toothpaste	\$3.04	7	52	0	0	52	\$0.00
23-363-4831	Toothbrushes	\$2.37	6	85	0	0	85	\$0.00
14-459-7070	Floss	\$3.67	11	50	0	0	50	\$0.00
88-225-9449	Lotion	\$2.64	19	85	0	0	85	\$0.00
24-443-3243	Eczema Cream	\$3.14	6	104	0	0	104	\$0.00

Load Data

Load Grocery Data

Update Data

Quantity Sold:

Quantity Restocked:

Update Sold Qty For Selected Item

Update Restocked Qty For Selected Item

Delete Selected Grocery Item

Sort Items Based on Sales

Save Data

Save Grocery Data

Save Sales Report

Save Restock Needs Report

Saved 3 records into the restocks needed output file

groceryrestocks - Notepad

File Edit Format View Help

ItemCode,ItemName,AvailableQty,QtyMinForRestock
90-222-4482,Women's Deodorant,19,20
32-998-9786,Chapsticks,3,19
60-641-7970,Radish,11,16

Ln 1, Col 1

100%

Windows (CRLF)

UTF-8

Note: Use additional methods as you see fit to complete the task.