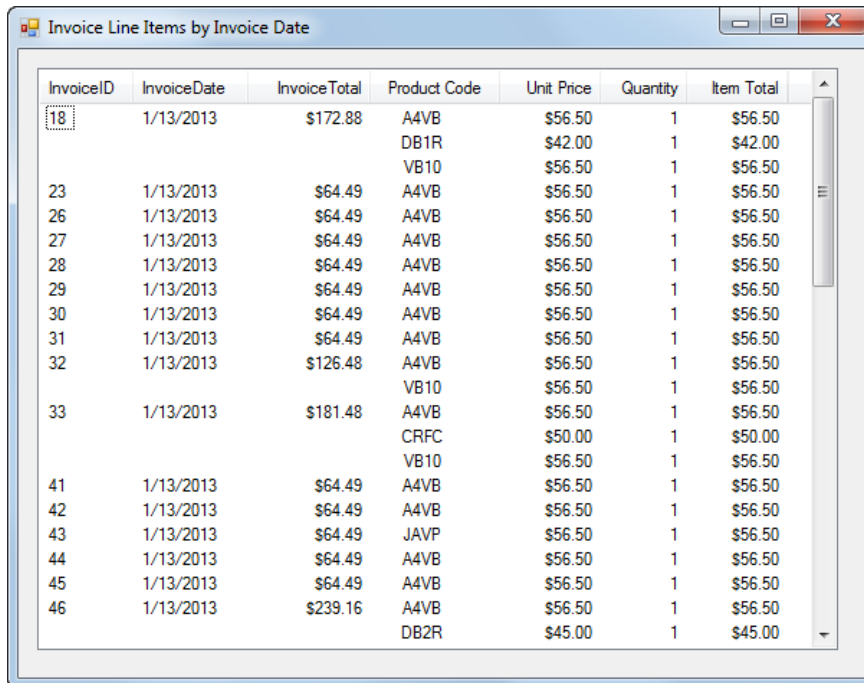


## Week 11: Assignment

### Use LINQ to create an Invoice Line Items application

---

In this exercise, you'll use LINQ to join the data in two List<> objects and then display that data in a ListView control.



InvoiceID	InvoiceDate	InvoiceTotal	Product Code	Unit Price	Quantity	Item Total
18	1/13/2013	\$172.88	A4VB	\$56.50	1	\$56.50
			DB1R	\$42.00	1	\$42.00
			VB10	\$56.50	1	\$56.50
23	1/13/2013	\$64.49	A4VB	\$56.50	1	\$56.50
26	1/13/2013	\$64.49	A4VB	\$56.50	1	\$56.50
27	1/13/2013	\$64.49	A4VB	\$56.50	1	\$56.50
28	1/13/2013	\$64.49	A4VB	\$56.50	1	\$56.50
29	1/13/2013	\$64.49	A4VB	\$56.50	1	\$56.50
30	1/13/2013	\$64.49	A4VB	\$56.50	1	\$56.50
31	1/13/2013	\$64.49	A4VB	\$56.50	1	\$56.50
32	1/13/2013	\$126.48	A4VB	\$56.50	1	\$56.50
			VB10	\$56.50	1	\$56.50
33	1/13/2013	\$181.48	A4VB	\$56.50	1	\$56.50
			CRFC	\$50.00	1	\$50.00
			VB10	\$56.50	1	\$56.50
41	1/13/2013	\$64.49	A4VB	\$56.50	1	\$56.50
42	1/13/2013	\$64.49	A4VB	\$56.50	1	\$56.50
43	1/13/2013	\$64.49	JAVP	\$56.50	1	\$56.50
44	1/13/2013	\$64.49	A4VB	\$56.50	1	\$56.50
45	1/13/2013	\$64.49	A4VB	\$56.50	1	\$56.50
46	1/13/2013	\$239.16	A4VB	\$56.50	1	\$56.50
			DB2R	\$45.00	1	\$45.00

#### Design the form

1. Open the attached InvoiceLineItems project. This project contains the Invoice Line Items form, along with the business and database classes and database files needed by the application.
2. Add a ListView control to the form, and set the View property of this control to Details.
3. Use the smart tag menu for the ListView control to display the ColumnHeader Collection Editor. Then, define the column headers for this control so they appear like the first and the last four shown above.

#### Add code to display the line item data

4. Add an event handler for the Load event of the form. Then, use the GetLineItems method in the LineItemDB class to get a List<LineItem> object, and store this list in a variable.
5. Define a query expression that returns all the line items from the line item list. The select clause for this query expression should select entire line items.
6. Use a foreach statement to execute the query and load the results into the ListView control.
7. Test the application to be sure it displays the line items correctly.

#### Enhance the application to include invoice data

8. Add two more columns to the ListView control for displaying the invoice date and invoice total. (To get these columns to display before the last four columns, you'll need to set their DisplayIndex properties to 1 and 2.) Then, add a statement to the Load event

## Week 11: Assignment

handler of the form that uses the GetInvoices method of the InvoiceDB class to get a List<Invoice> object, and store the list in a variable.

9. Modify the query expression so it joins the data in the invoice list with the data in the line item list, so it sorts the results by invoice date, and so only the fields that are needed by the form are returned by the query.
10. Modify the foreach statement so it adds the invoice ID, invoice date, and invoice total to the ListView control.
11. Test the application to be sure it displays the invoice data correctly.

### **Enhance the application so it doesn't repeat invoice information**

12. Declare a variable outside the foreach statement to hold an invoice ID, and initialize this variable to 0. Then, add code within the foreach statement that checks if the invoice ID for the current item is equal to the invoice ID in the variable you just declared. If they aren't equal, the invoice ID, invoice date, and invoice total should be added to the ListView control, and the invoice ID variable should be set to the value of the invoice ID for the current item. Otherwise, spaces should be added to the ListView control for these fields.
13. Test this change to be sure it works correctly.