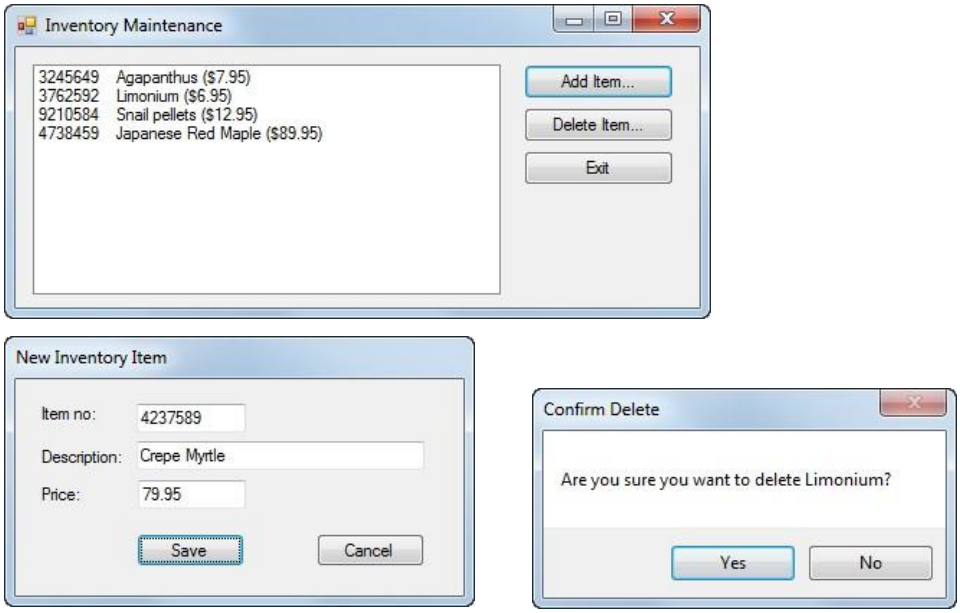
Create and use an Inventory Item class

In this exercise, you'll add a class to an Inventory Maintenance application and then add code to the two forms that use this class.



Open the project and add an InvItem class

1. Extract the provided start project (`W01 Exercise Start.zip`). Then, review the existing code for both of the forms so you get an idea of how this application should work.

2. Add a class named `InvItem` to this project, and add the properties, method, and constructors that are shown in the table below.

Property	Description
`ItemNo`	Gets or sets an int that contains the item's number.
`Description`	Gets or sets a string that contains the item's description.
`Price`	Gets or sets a decimal that contains the item's price.
Method	**Description**
`GetDisplayText()`	Returns a string that contains the item's number, description, and price formatted like this: `3245649 Agapanthus ($7.95)`. (The item number and description are separated by four spaces.)
Constructor	**Description**
`()`	Creates an InvItem object with default values.
`(itemNo, description, price)`	Creates an InvItem object with the specified values.

Add code to implement the New Item form

3. Display the code for the New Item form, and declare a class variable named `invItem` of type InvItem with an initial value of null.
4. Add a public method named `GetNewItem` that displays the form as a dialog box and returns an InvItem object.
5. Add code to the `btnSave_Click` event handler that creates a new InvItem object and closes the form if the data is valid.

Add code to implement the Inventory Maintenance form

6. Display the code for the **Inventory Maintenance** form, and declare a class variable named `invItems` of type `List<InvItem>` with an initial value of null.
7. Add a statement to the `frmInvMaint_Load` event handler that uses the `GetItems` method of the `InvItemDB` class to load the items list.
8. Add code to the `FillItemListBox` method that adds the items in the list to the `Items` list box. Use the `GetDisplayText` method of the `InvItem` class to format the item data.
9. Add code to the `btnAdd_Click` event handler that creates a new instance of the **New Item** form and executes the `GetNewItem` method of that form. If the InvItem object that's returned by this method is not null, this event handler should add the new item to the list, call the `SaveItems` method of the `InvItemDB` class to save the list, and then refresh the Items list box. Test the application to be sure this event handler works.
10. Add code to the `btnDelete_Click` event handler that removes the selected item from the list, calls the `SaveItems` method of the `InvItemDB` class to save the list, and refreshes the Items list box. Be sure to confirm the delete operation. Then, test the application to be sure this event handler works.