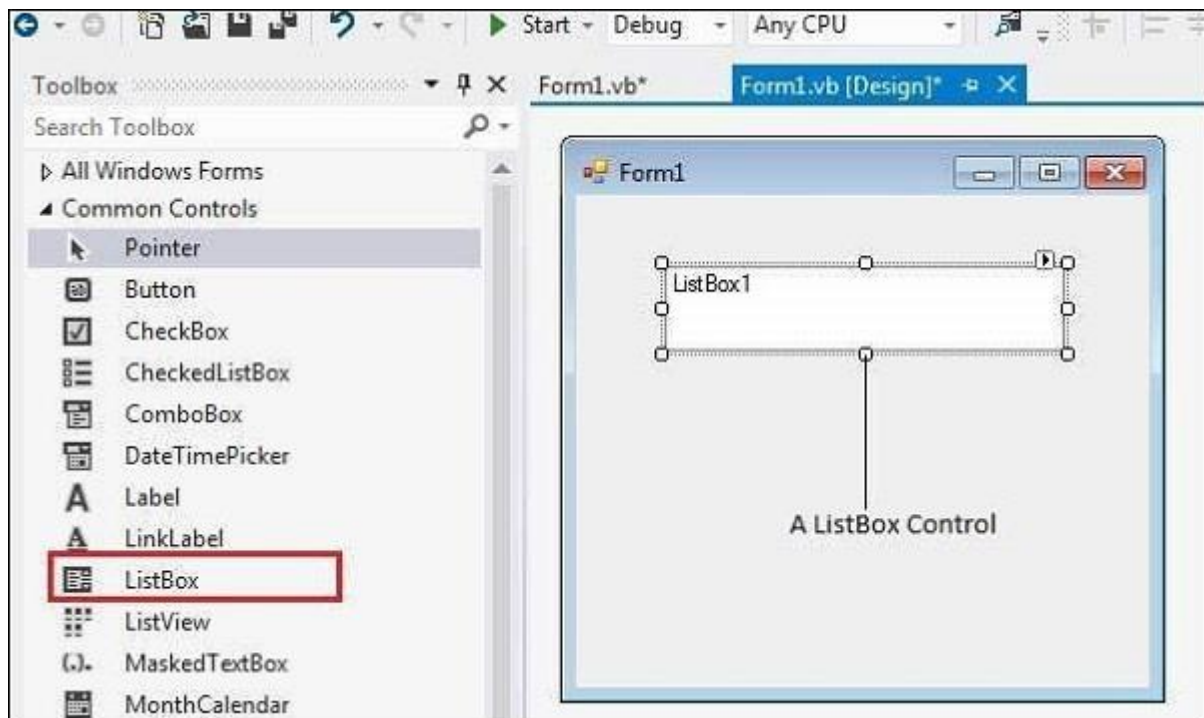The ListBox represents a Windows control to display a list of items to a user. A user can select an item from the list. It allows the programmer to add items at design time by using the properties window or at the runtime.

Let's create a list box by dragging a ListBox control from the Toolbox and dropping it on the form.

You can populate the list box items either from the properties window or at runtime. To add items to a ListBox, select the ListBox control and get to the properties window, for the properties of this control. Click the ellipses (...) button next to the Items property. This opens the String Collection Editor dialog box, where you can enter the values one at a line.

Events of the ListBox Control

The following are some of the commonly used events of the ListBox control −

Sr.No.	Event & Description
1	**Click** Occurs when a list box is selected.
2	**SelectedIndexChanged**

	Occurs when the SelectedIndex property of a list box is changed.

Consult Microsoft documentation for detailed list of properties, methods and events of the ListBox control.

Example 1

In the following example, let us add a list box at design time and add items on it at runtime.

Take the following steps −

- Drag and drop two labels, a button and a ListBox control on the form.
- Set the Text property of the first label to provide the caption "Choose your favourite destination for higher studies".
- Set the Text property of the second label to provide the caption "Destination". The text on this label will change at runtime when the user selects an item on the list.
- Click the listbox and the button controls to add the following codes in the code editor.

```vbnet
Public Class Form1
  Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    ' Set the caption bar text of the form.
    Me.Text = "Working with listbox"
    ListBox1.Items.Add("Canada")
    ListBox1.Items.Add("USA")
    ListBox1.Items.Add("UK")
    ListBox1.Items.Add("Japan")
    ListBox1.Items.Add("Russia")
    ListBox1.Items.Add("China")
    ListBox1.Items.Add("India")
  End Sub

  Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    MsgBox("You have selected " + ListBox1.SelectedItem.ToString())
  End Sub

  Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs)
    Handles ListBox1.SelectedIndexChanged
    Label2.Text = ListBox1.SelectedItem.ToString()
  End Sub
End Class
```

When the above code is executed and run using Start button available at the Microsoft Visual Studio tool bar, it will show the following window −

When the user chooses a destination, the text in the second label changes −

Clicking the Select button displays a message box with the user's choice −

Example 2

In this example, we will fill up a list box with items, retrieve the total number of items in the list box, sort the list box, remove some items and clear the entire list box.

Design the Form −

Add the following code in the code editor window −

```
Public Class Form1
  Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    ' Set the caption bar text of the form.
    Me.Text = "tutorialspont.com"
    ' creating multi-column and multiselect list box
    ListBox1.MultiColumn = True
    ListBox1.SelectionMode = SelectionMode.MultiExtended
  End Sub

  'populates the list
  Private Sub Button1_Click_1(sender As Object, e As EventArgs) _
    Handles Button1.Click
    ListBox1.Items.Add("Safety")
    ListBox1.Items.Add("Security")
    ListBox1.Items.Add("Governance")
    ListBox1.Items.Add("Good Music")
    ListBox1.Items.Add("Good Movies")
    ListBox1.Items.Add("Good Books")
    ListBox1.Items.Add("Education")
    ListBox1.Items.Add("Roads")
    ListBox1.Items.Add("Health")
    ListBox1.Items.Add("Food for all")
    ListBox1.Items.Add("Shelter for all")
    ListBox1.Items.Add("Industrialisation")
    ListBox1.Items.Add("Peace")
    ListBox1.Items.Add("Liberty")
    ListBox1.Items.Add("Freedom of Speech")
```

```vb
    End Sub
    'sorting the list
    Private Sub Button2_Click(sender As Object, e As EventArgs) _
       Handles Button2.Click
       ListBox1.Sorted = True
    End Sub

    'clears the list
    Private Sub Button3_Click(sender As Object, e As EventArgs) _
       Handles Button3.Click
       ListBox1.Items.Clear()
    End Sub

    'removing the selected item
    Private Sub Button4_Click(sender As Object, e As EventArgs) _
         Handles Button4.Click
       ListBox1.Items.Remove(ListBox1.SelectedItem.ToString)
    End Sub

    'counting the number of items
    Private Sub Button5_Click(sender As Object, e As EventArgs) _
       Handles Button5.Click
       Label1.Text = ListBox1.Items.Count
    End Sub

    'displaying the selected item on the third label
    Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs) _
       Handles ListBox1.SelectedIndexChanged
       Label3.Text = ListBox1.SelectedItem.ToString()
    End Sub
End Class
```
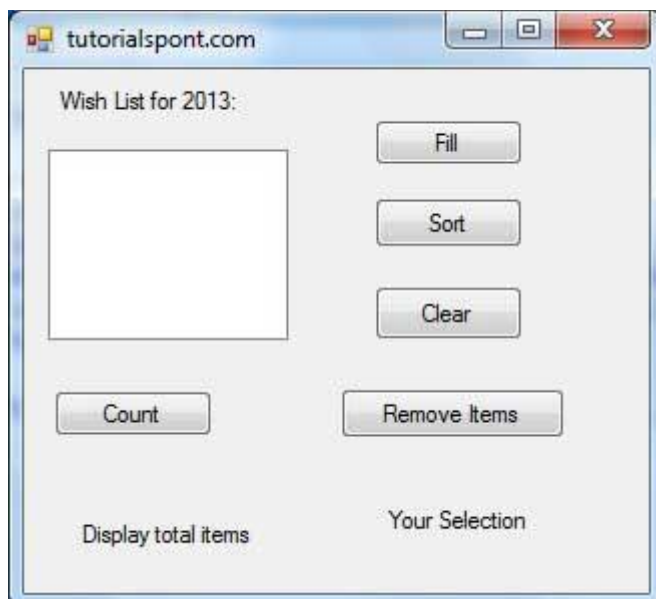
When the above code is executed and run using Start button available at the Microsoft Visual Studio tool bar, it will show the following window −

Fill the list and check workings of other buttons −

The ListView control is used to display a list of items. Along with the TreeView control, it allows you to create a Windows Explorer like interface.

Let's click on a ListView control from the Toolbox and place it on the form.

Example

In this example, let us create a list view at runtime. Let's double click on the Form and put the follow code in the opened window.

```vb
Public Class Form1
  Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    'create a new ListView
    Dim ListView1 As ListView
    ListView1 = New ListView()
    ListView1.Location = New Point(10, 10)
    ListView1.Size = New Size(150, 150)

    Me.Controls.Add(ListView1)
    'Creating the list items
    Dim ListItem1 As ListViewItem
    ListItem1 = ListView1.Items.Add("Item 1")

    Dim ListItem2 As ListViewItem
    ListItem2 = ListView1.Items.Add("Item 2")
```
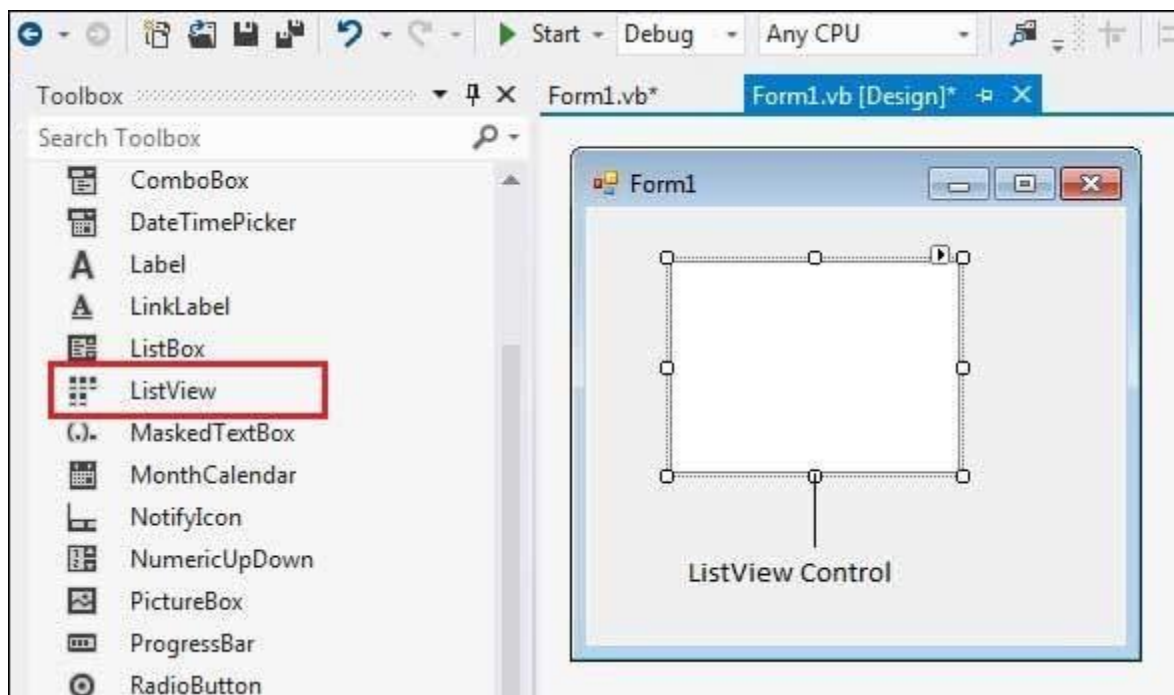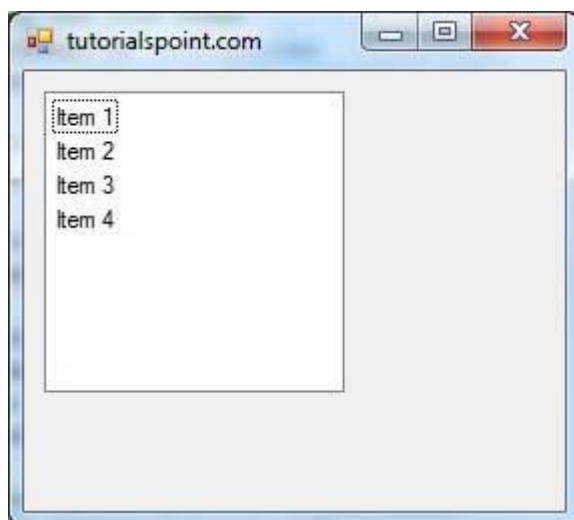
```
        Dim ListItem3 As ListViewItem
        ListItem3 = ListView1.Items.Add("Item 3")

        Dim ListItem4 As ListViewItem
        ListItem4 = ListView1.Items.Add("Item 4")
        'set the view property

        ListView1.View = View.SmallIcon
        ' Set the caption bar text of the form.
        Me.Text = "tutorialspoint.com"
    End Sub
End Class
```

When the above code is executed and run using **Start** button available at the Microsoft Visual Studio tool bar, it will show the following window −

The *ListView* control displays a list of items along with icons. The Item property of the ListView control allows you to add and remove items from it. The *SelectedItem* property contains a collection of the selected items. The *MultiSelect* property allows you to set select more than one item in the list view. The *CheckBoxes* property allows you to set check boxes next to the items.