

31927 32998: Application Development with .NET

Week-7 Lecture

Windows Forms in C#: Part - 1



Outline

- Windows Forms
 - Event Handling basics
 - Mouse and Keyboard events
- Controls
 - Dialog / Message box
 - Buttons, Labels, and Textbox
 - Checkboxes and RadioButtons
 - ListBoxes and ComboBoxes
 - GroupBoxes and Panels
 - Menus

Windows Forms

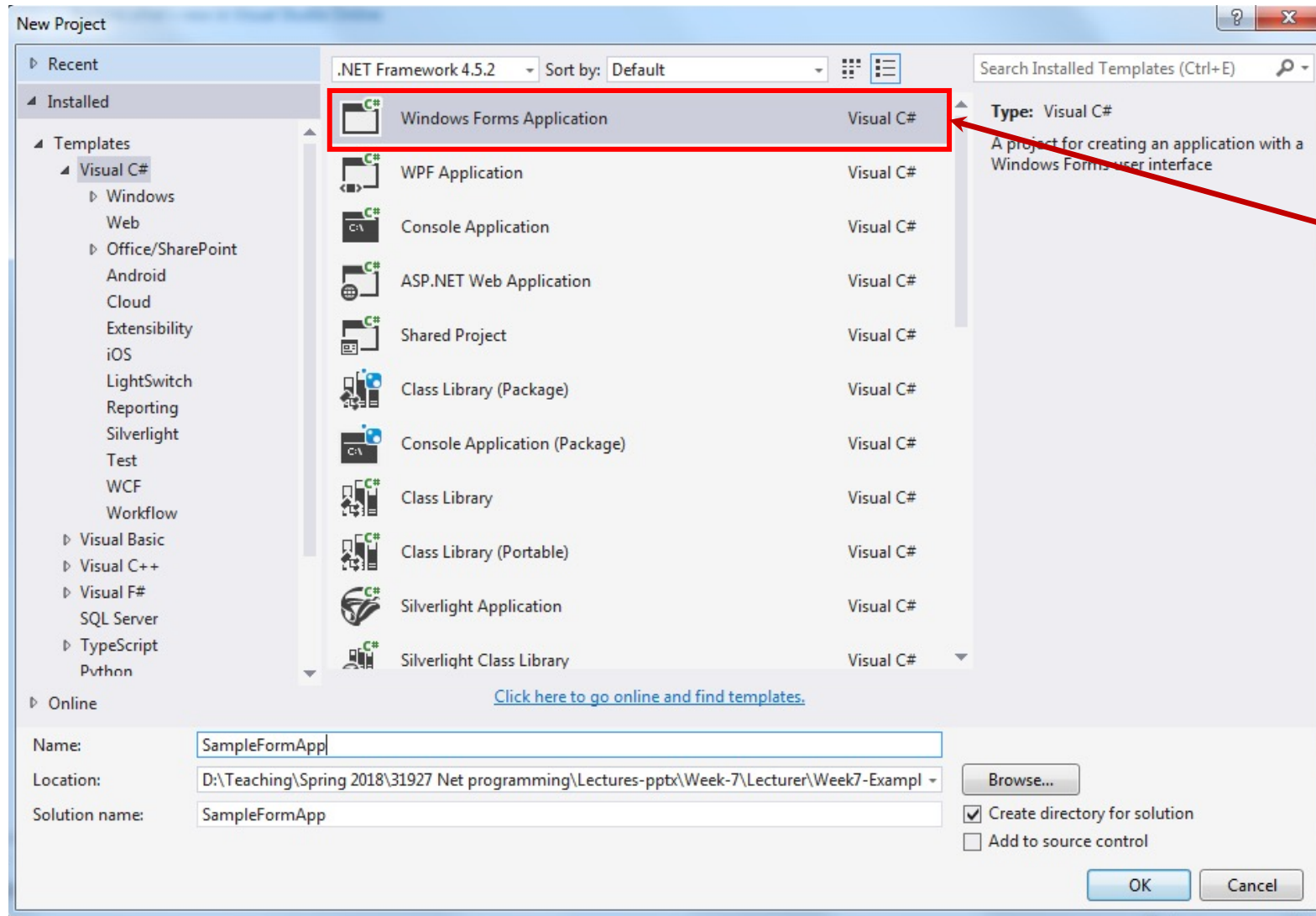
- Instead of making console project, make windows form application project.
- A form is a graphical element that appears on your computer's desktop
- Is a container for components and controls
- Components implement `IComponent` Interface
- A control is a component that has a graphical interface (such as a button) and event handlers.

Event Handling

- Graphical User Interface's (GUIs) are event driven
- All controls have events associated with them.
- Examples of events would include mouse and keyboard handling. Controls being moved and resized. Controls being created and closed.
- Events are user actions such as key press, clicks, mouse movements, etc., or some occurrence such as system generated notifications.
- Applications need to respond to events when they occur. For example, interrupts. Events are used for inter-process communication.
- Click in the Properties Window of a Control to create the required event handler. The generated handler will be in the form

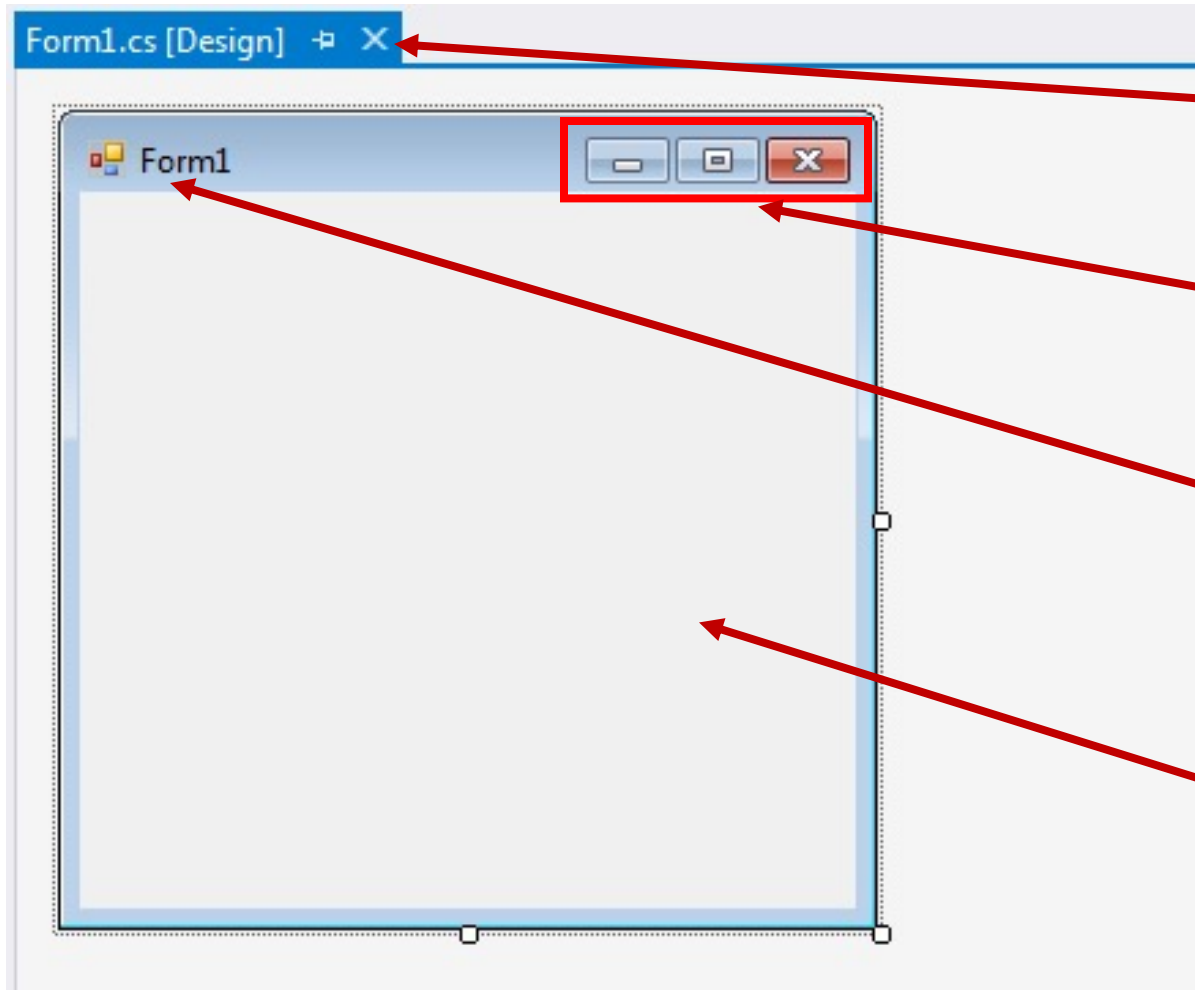
```
private void button1_Click(object sender, EventArgs e)
```

Creating a Windows Form



Select Windows Forms Application

Creating a Windows Form



Designer view:

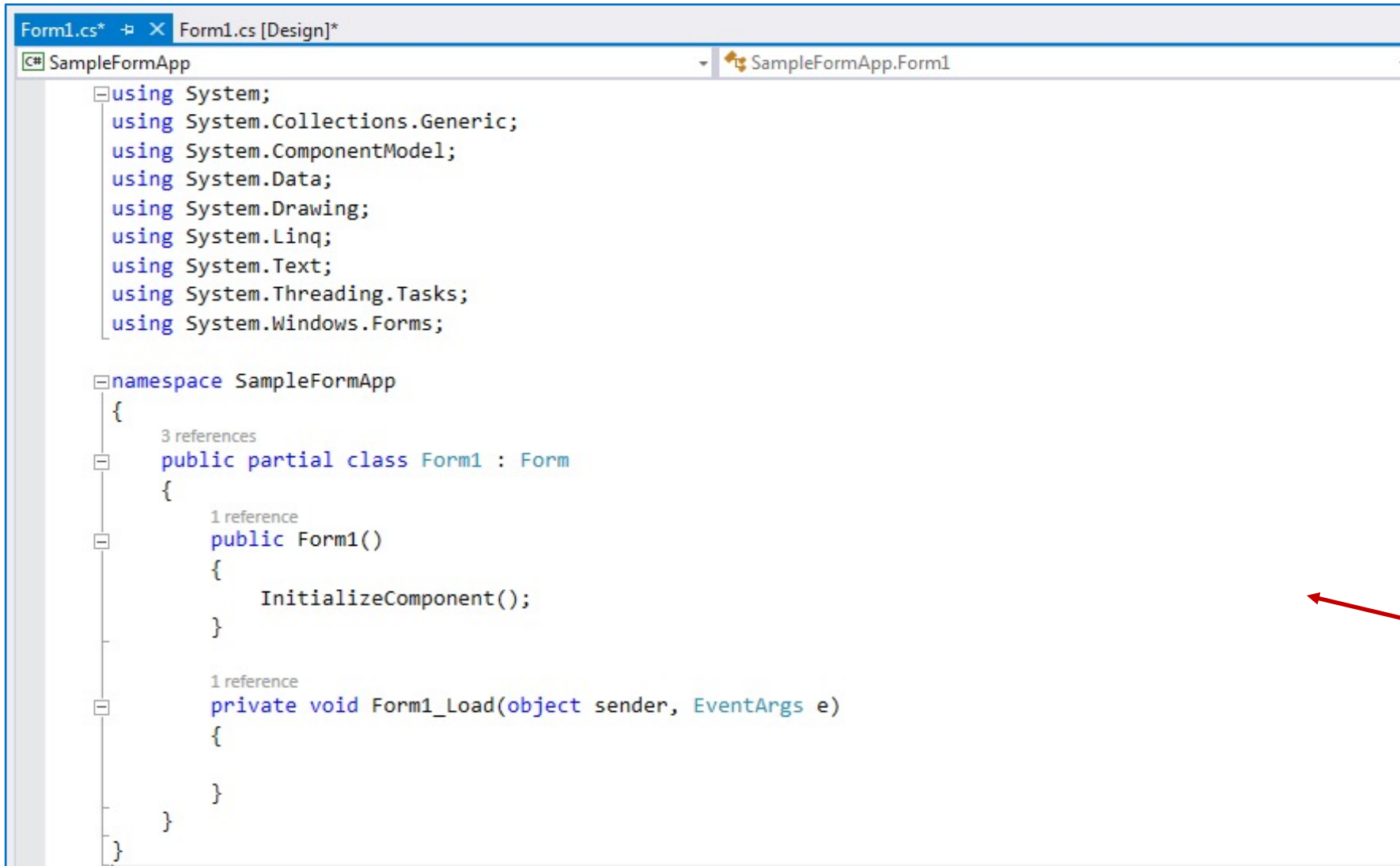
Visual representation of the window that will open when a window application is opened.

Title bar has minimize, maximize, and close buttons

“Form1” is the default name of the form, can be changed

- Default blank form
- Similar to a canvas where control can be added
- Controls can be dragged and dropped
- The size and position of the form can be adjusted
- The size and position of the controls can be adjusted

Creating a Windows Form



```
Form1.cs*  Form1.cs [Design]*
C# SampleFormApp SampleFormApp.Form1

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

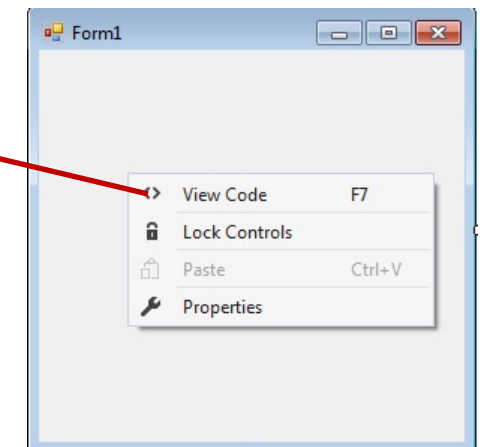
namespace SampleFormApp
{
    3 references
    public partial class Form1 : Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();
        }

        1 reference
        private void Form1_Load(object sender, EventArgs e)
        {

        }
    }
}
```

Code view/Window:

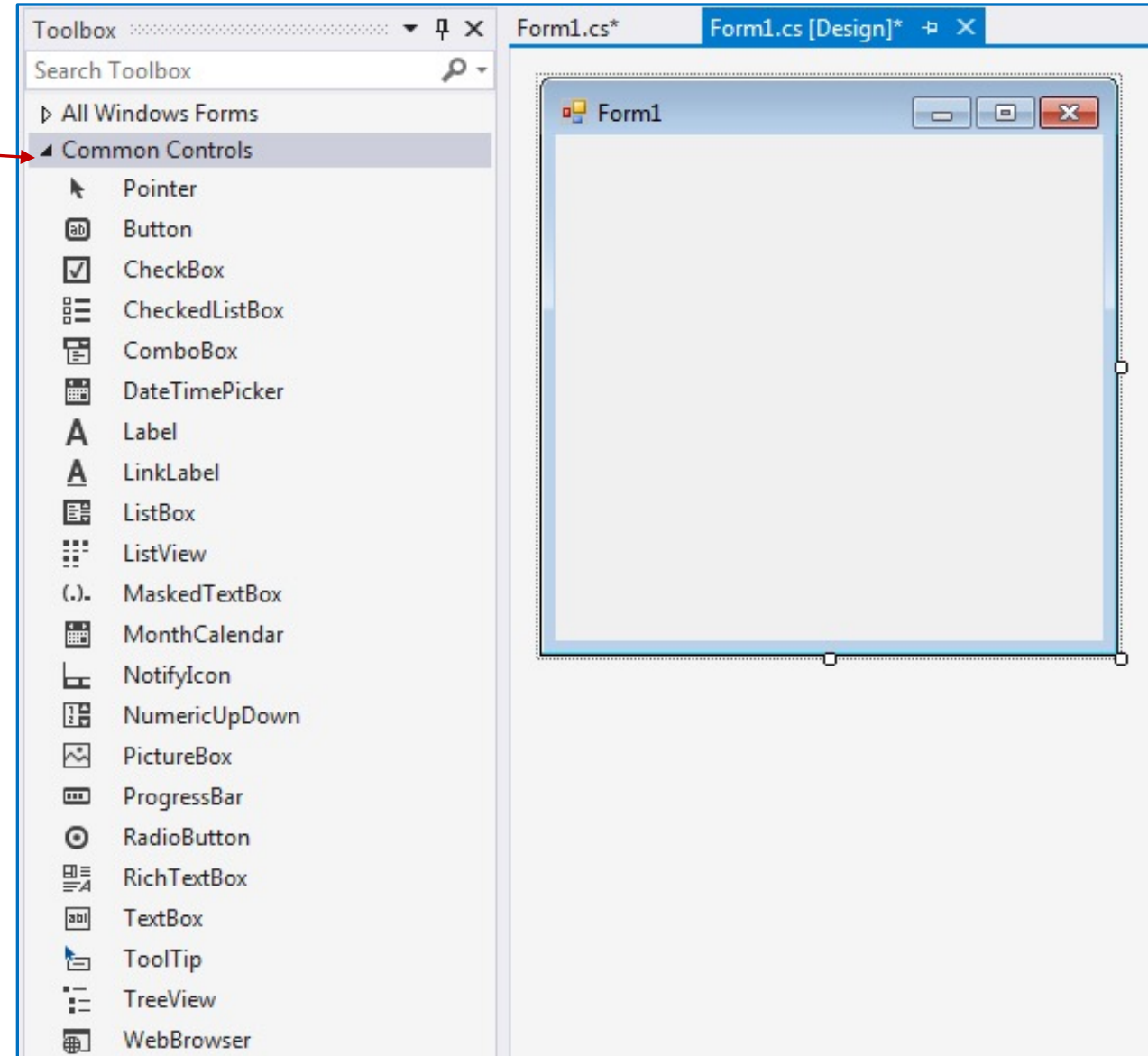
- Double click the Form to switch to code window
- Or Right click on the Form and select View Code



Creating a Windows Form

Toolbox View:

Shows the available Controls for GUI developments



Controls : MessageBox

- `MessageBox.Show` (Over 20 overloaded options)
- Will display message on the screen in a dialog box. User clicks OK to close box. Useful for displaying debugging messages, etc.

Simple example:

`MessageBox.Show(message, title);`

`MessageBox.Show("Hello World from MessageBox!", "My Message");`



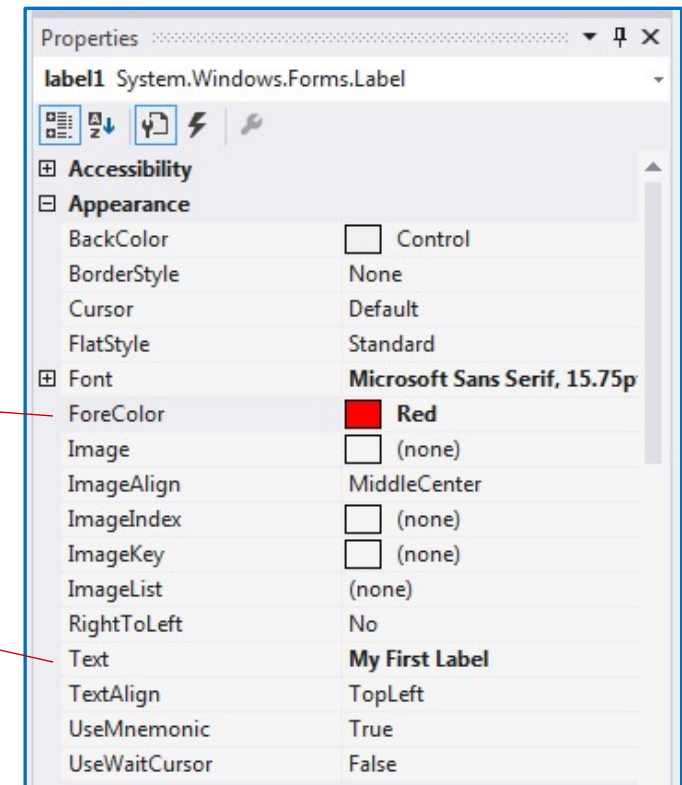
Controls : Label

Displays read only text on the application. Can be changed programmatically.

A Label

: Drag and Drop the label control from toolbox to the Form

Example:



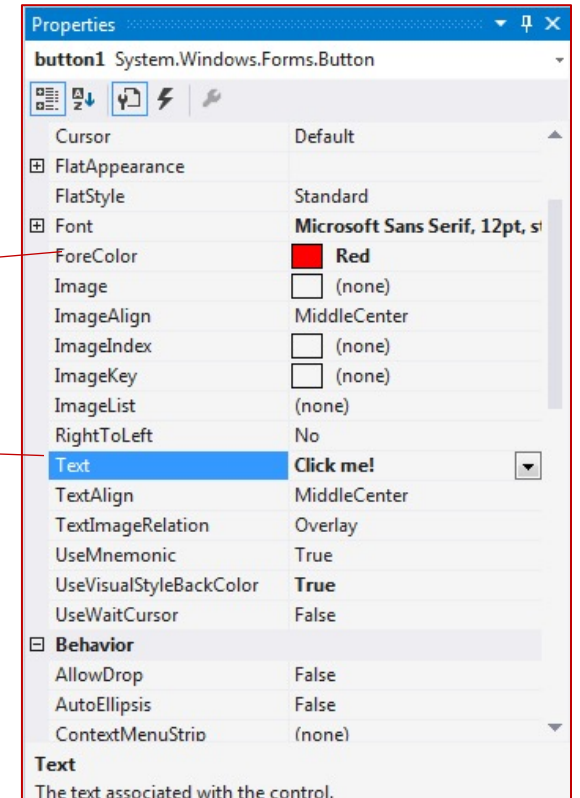
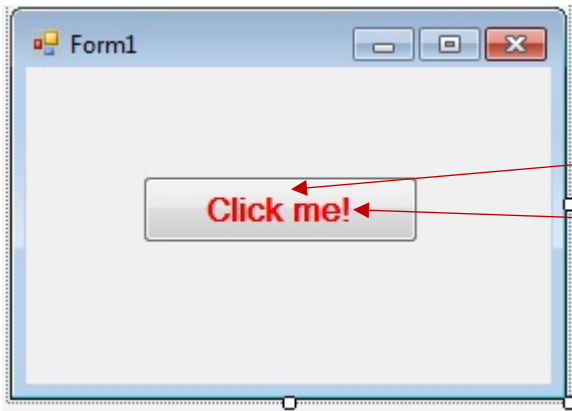
Controls : Button

A control that the user clicks to trigger a specific action.

Creates a button_Click event.

 **Button** : Drag and Drop the Button control from toolbox to the Form

Example:

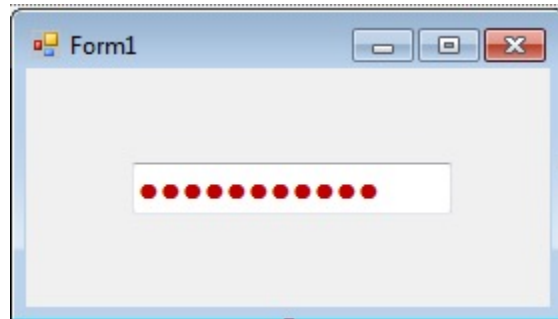
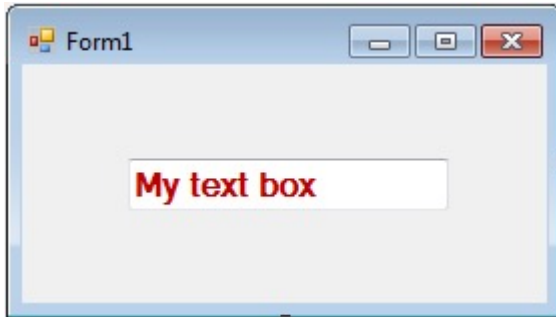


Controls : TextBox

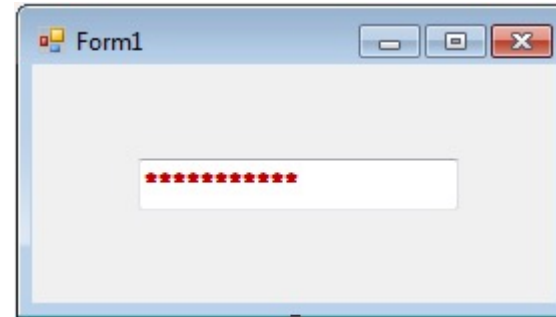
Used to display text or can be changed by user. Can be made into a password box by setting the `UseSystemPasswordChar` property to true.

 : Drag and Drop the TextBox control from toolbox to the Form

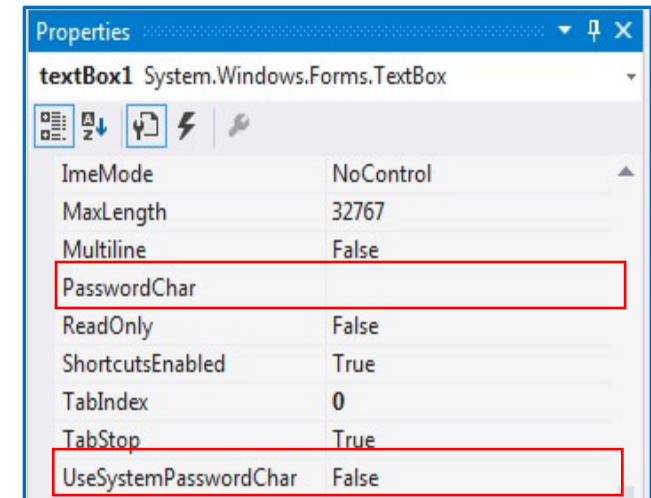
Example:



`UseSystemPasswordChar = true`



`PasswordChar = *`



Controls : CheckBoxes

CheckBoxes allow the user to make multiple selections from a number of options. They have a state that is either checked or unchecked. CheckBox comes with a caption, which you can set in the Text property.

 : Drag and Drop the CheckBox control from toolbox to the Form

Example:

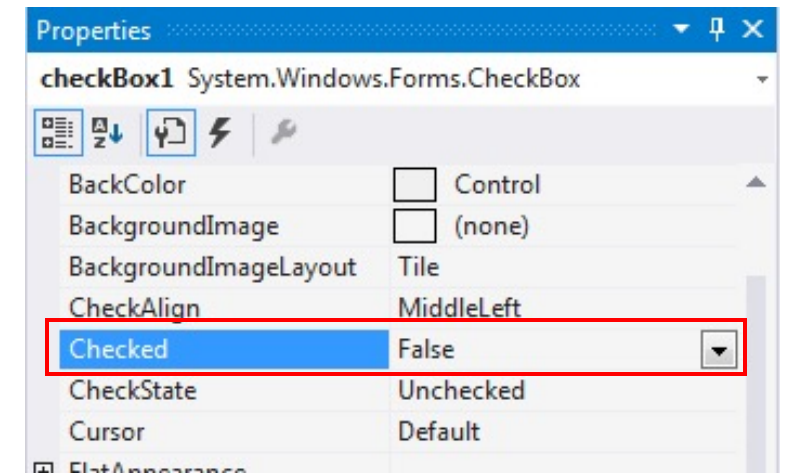


Dinner Menu

- ☐ Salad
- ☒ Pizza
- ☐ Burger
- ☒ Juice

Order

```
if (checkBox1.Checked == true){}
```

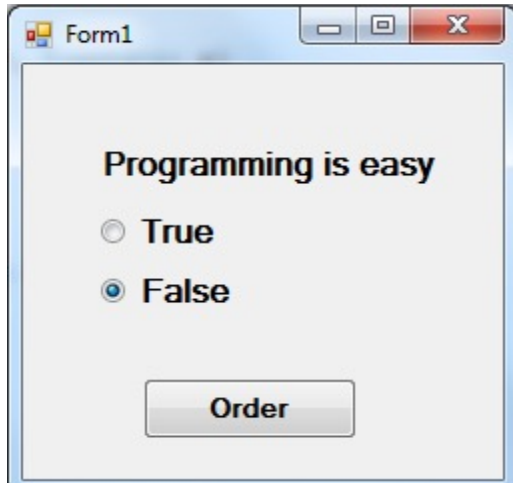


Controls : RadioButtons

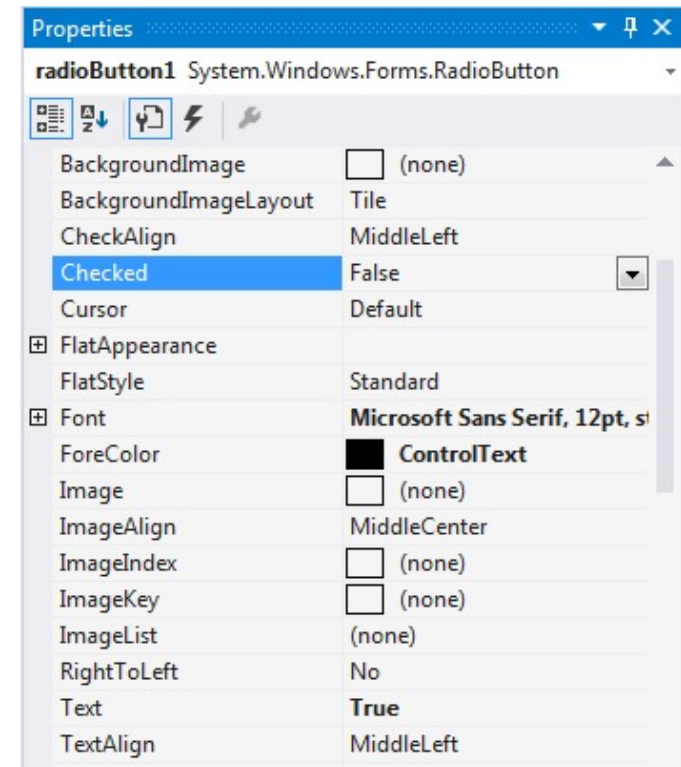
Similar to CheckBoxes with one important exception. Within any container only one radio button can be checked.

 **RadioButton** : Drag and Drop the RadioButton control from toolbox to the Form

Example:



```
if (radioButton1.Checked == true){}
```



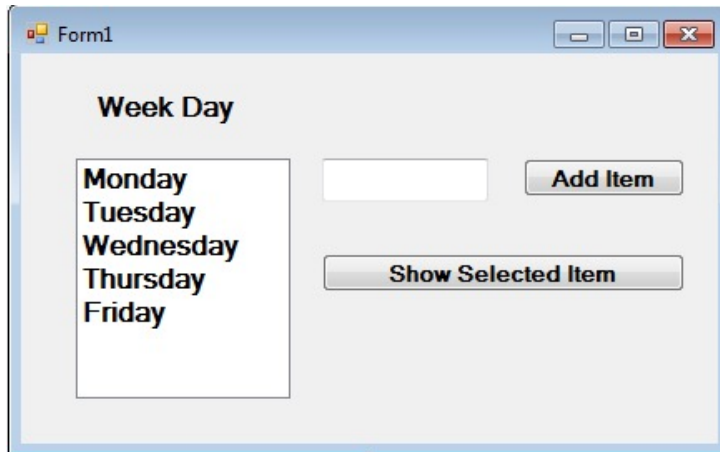
Controls : ListBox

The ListBox control enables you to display a list of items to the user that the user can select by clicking.



: Drag and Drop the ListBox control from toolbox to the Form

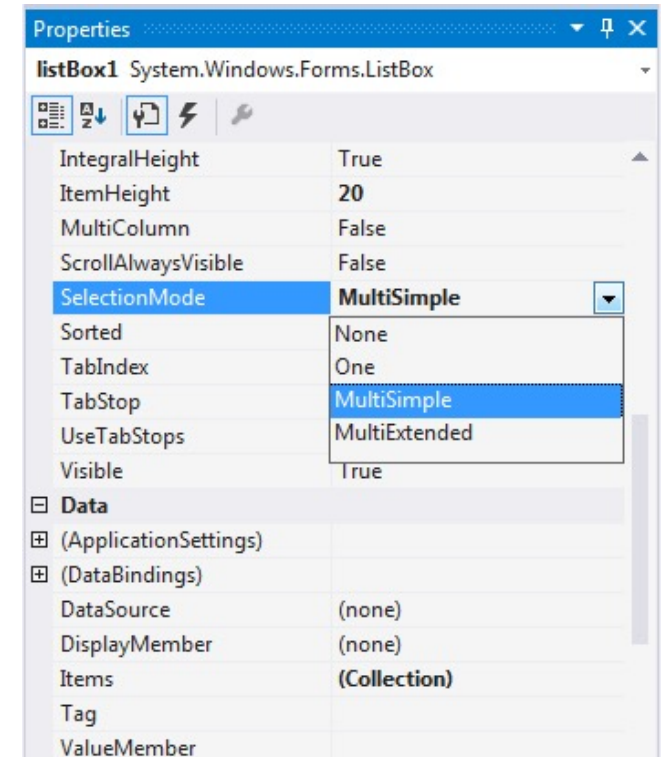
Example:

The screenshot shows a Windows Form titled 'Form1'. It contains a label 'Week Day' above a list box. The list box contains the items 'Monday', 'Tuesday', 'Wednesday', 'Thursday', and 'Friday'. To the right of the list box is a text input field and an 'Add Item' button. Below the list box is a 'Show Selected Item' button.

```
// For Adding items
listBox1.Items.Add("Hello");

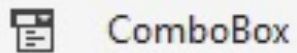
// For Single item Selection
MessageBox.Show(listBox1.SelectedItem.ToString());

// For multiple item Selections
foreach (Object obj in listBox1.SelectedItems)
{
    MessageBox.Show(obj.ToString());
}
```

The screenshot shows the Visual Studio Properties window for a 'listBox1' control of type 'System.Windows.Forms.ListBox'. The 'SelectionMode' property is set to 'MultiSimple', which is highlighted in the dropdown menu. Other visible properties include 'IntegralHeight' (True), 'ItemHeight' (20), 'MultiColumn' (False), 'ScrollAlwaysVisible' (False), 'Sorted' (None), 'TabIndex' (One), 'TabStop' (MultiSimple), 'UseTabStops' (MultiExtended), and 'Visible' (True). The 'Data' section shows 'DataSource' as '(none)', 'DisplayMember' as '(none)', 'Items' as '(Collection)', 'Tag' as empty, and 'ValueMember' as empty.

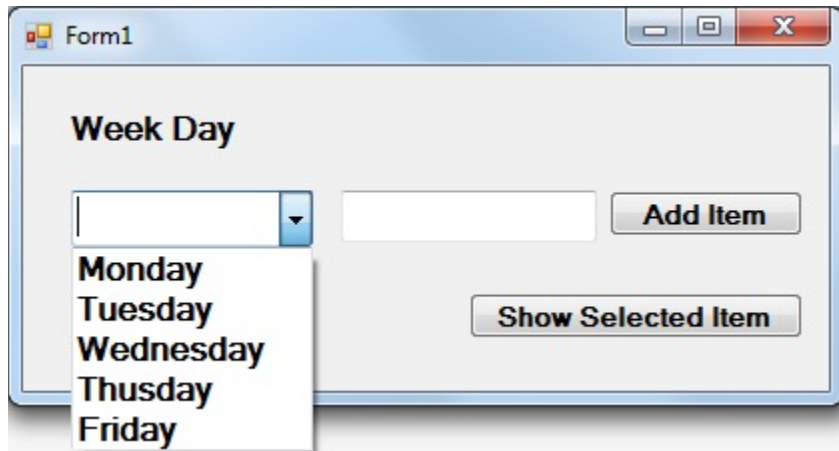
Controls : ComboBox

A ComboBox displays a text box combined with a ListBox, which enables the user to select items from the list or enter a new value . Similar to `ListBox` except we have a drop down list of the items



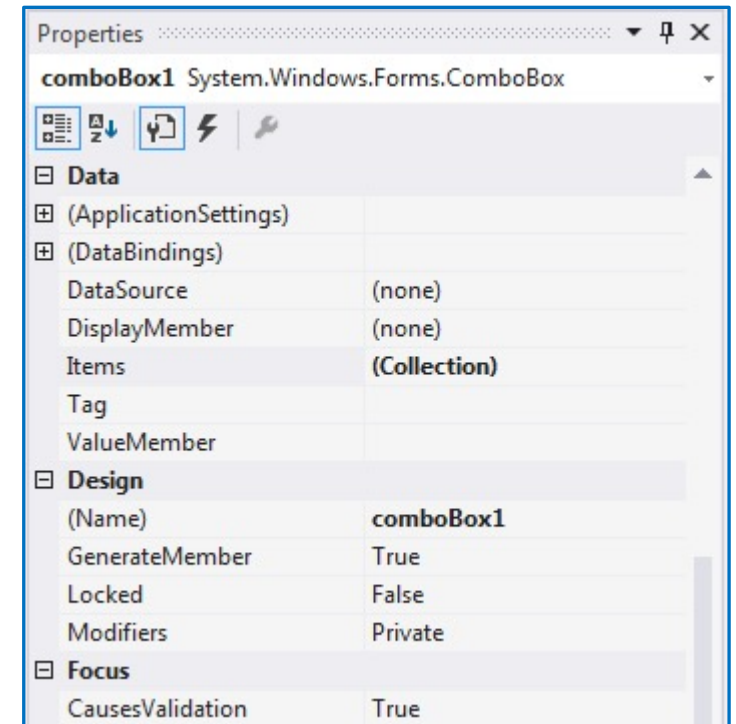
: Drag and Drop the ComboBox control from toolbox to the Form

Example:



```
// For Adding items
comboBox1.Items.Add("Hello");

// Show selected item
MessageBox.Show(comboBox1.Text);
```

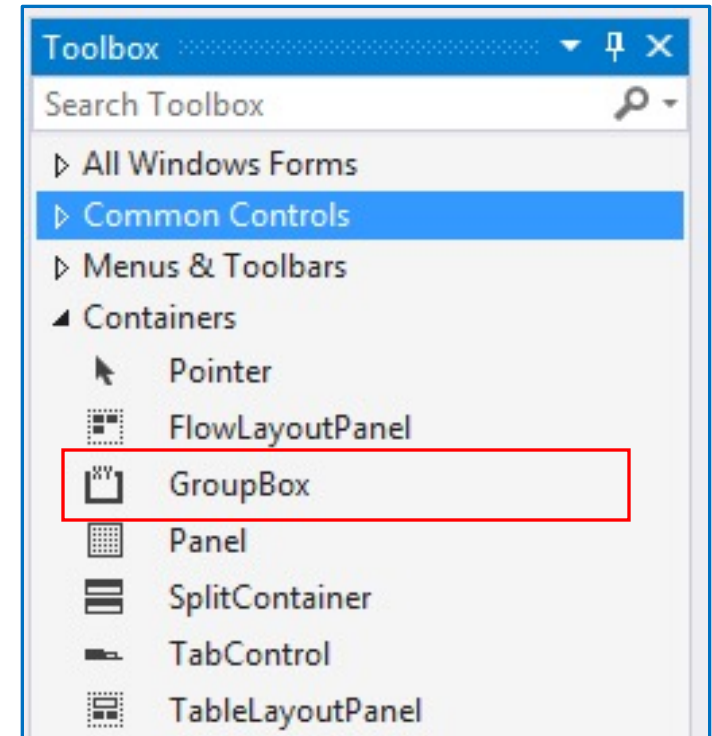


Controls : Groupboxes

1. **GroupBoxes** can contain multiple controls.
2. Allows the controls to be moved and hidden all at the same time.
3. **GroupBoxes** can display a caption and do not have scroll bars.
4. **GroupBox** control displays a caption.

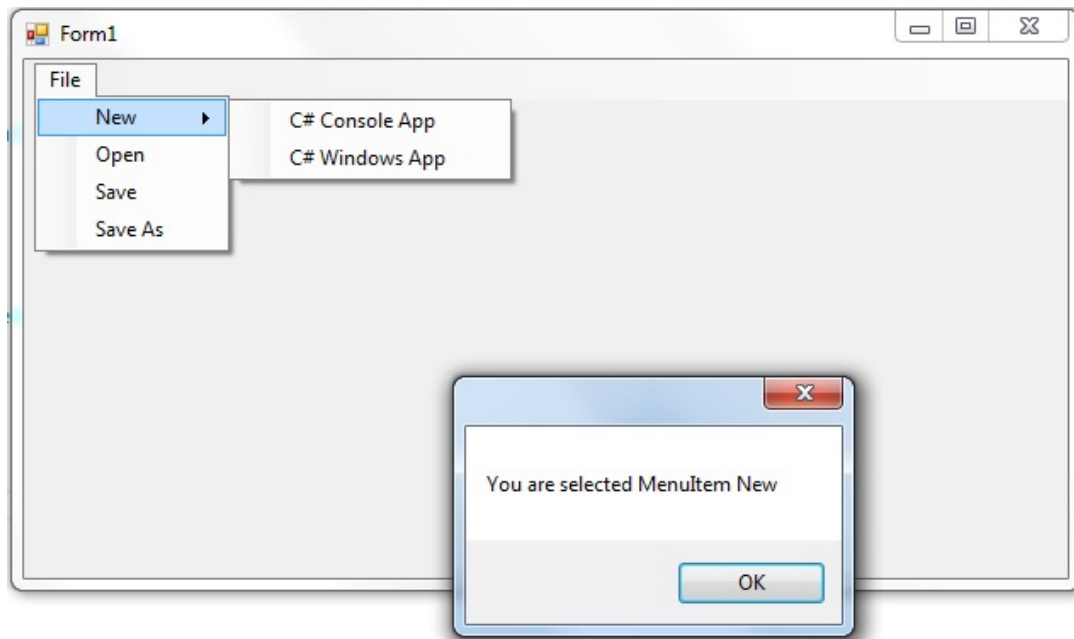
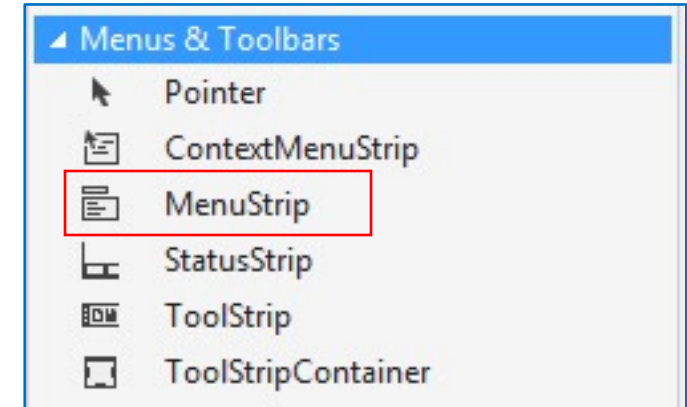


The screenshot shows a Windows Form titled "Form1". It contains two GroupBoxes. The first GroupBox is titled "Dinner Menu" and contains a list of items: Salad, Pizza, Burger, and Juice, followed by an "Order" button. The second GroupBox is titled "Lunch Menu" and contains a list of items: Salad, Sushi, Steam Rice, and Juice, followed by an "Order" button.



Controls : Menus

1. A Menu on a Windows Form is created with a MainMenu object, which is a collection of MenuItem objects.
2. Menus provide groups of related commands for Windows Applications



```
private void newToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("You are selected MenuItem New");
}

private void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("You are selected MenuItem Open");
}
```