UNIT 3

ADDITIONAL CONTROLS AND MENUS OF WINDOWS

Outline of the Chapter

1. WORKING WITH OTHER CONTROLS OF TOOL BOX:

1.1 List Box Control:

- The ListBox control enables you to represents or display a list of items to a user that the user can select by clicking.
- 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.
- A List or ListBox is an alternative option. It is similar to a drop-down list, except that the list is already open for you, and you no need to click any button to view the list.

Properties:

Property Name	Description
Items	It gets the items of the list box.
MultiColumn	It gets or sets a value indicating whether the listbox supports multiple columns.
ScrollAlwaysVisible	It gets or sets a value indicating whether the vertical scroll bar is shown at all times.
SelectedIndices	It gets a collection that contains the zero-based index of all currently selected item in a listbox.
SelectedItem	It gets or sets the currently selected item in the listbox.
SelectedValue	It gets a collection containing the currently selected items in the listbox.
SelectionMode	It gets or sets the method in which items are selected in the list box. This property has values: None, One, MultiSimple, and MultiExtended
Sorted	It gets or sets a value indicating whether the items in the list box are sorted alphabetically.
TopIndex	Gets or sets the index of the first visible item of a list box.

Check the common properties like, AutoSize, BackColor, Front, ForeColor, Enable and Visible.

Methods:

Method Name	Description
ClearSelected()	Unselects all items in the ListBox.
GetSelected()	Returns a value indicating whether the specified item is selected.
SetSelected()	Selects or clears the selection for the specified item in a ListBox.
FindString()	Finds the first item in the ListBox that starts with the string specified as an argument.

Events:

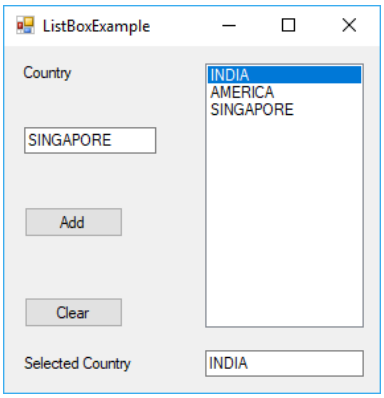
Event Name	Description
SelectedIndexChanged	Occurs when the SelectedIndex property of a list box is changed.

SelectedValueChanged	Occurs when the SelectedValue property of a list box is changed.

Example:

```
Public Class ListBoxExample
      Private Sub ListBoxExample_Load(ByVal sender As System.Object, ByVal e
      As System.EventArgs) Handles MyBase.Load
              ListBox1Country.Items.Add("INDIA")
                ListBox1Country.Items.Add("AMERICA")
      End Sub

      Private Sub Button2Add_Click(ByVal sender As System.Object, ByVal e As
      System.EventArgs) Handles Button2Add.Click
          If TextBox1Country.Text <> "" Then
          ListBox1Country.Items.Add(TextBox1Country.Text)
          End If
      End Sub
      Private Sub Button1Clear_Click(ByVal sender As System.Object, ByVal e
      As System.EventArgs) Handles Button1Clear.Click
           ListBox1Country.Items.Clear()
      End Sub

      Private Sub ListBox1Country_Click(ByVal sender As System.Object, ByVal
      e As System.EventArgs) Handles ListBox1Country.Click
           TextBox1SelectedCountry.Text =
           ListBox1Country.SelectedItem.ToString()
      End Sub
End Class
```

Output:



1.2 Combo Box Control:

- The ComboBox control is **used to display a drop-down list of various items.**
- It is a combination of a text box and drop-down list, text box in which the user can enters an item and a drop-down list from which the user selects an item.
- A drop-down list (Also called as ComboBox) is a small icon containing some text and a button with symbol of down arrow.
- To make selection from the drop-down list, just click whatever option you want.

- The default behaviour of ComboBox, displays an edit field with a hidden drop-down list.

Properties:

Property Name	Description
DataSource	Gets or sets the data source for this ComboBox.
DropDownStyle	Gets or sets a value specifying the style of the combo box. That is Simple, DropDown or Dropdown List
DroppedDown	Gets or sets a value indicating whether the combo box is displaying its drop-down portion.
MaxDropDownItems	Gets or sets the maximum number of items to be displayed in the drop-down part of the combo box.
MaxLength	Gets or sets the maximum number of characters a user can enter in the editable area of the combo box.
Items	Gets an object representing the collection of the items contained in this ComboBox.
SelectedIndex	Gets or sets the index specifying the currently selected item.
SelectedItem	Gets or sets currently selected item in the ComboBox.
SelectedValue	Gets or sets the value of the member property specified by the Value Member property.
Sorted	Gets or sets a value indicating whether the items in the combo box are sorted.

Check the common properties like, AutoSize, BackColor, Front, ForeColor, Enable and Visible.

Methods:

Method Name	Description
FindString()	Finds the first item in the combo box that starts with the string specified as an argument.
GetSelected()	Returns a value indicating whether the specified item is selected.
SetSelected()	Selects or Clears the selection for the specified item in a List.
ClearSelected()	Unselects all items in the ListBox.

Events:

Event Name	Description
DropDown	Occurs when the drop-down portion of a combo box is displayed.
DropDownClosed	Occurs when the drop-down portion of a combo box is no longer visible.
DropDownStyleChanged	Occurs when the DropDownStyle property of the ComboBox has changed.

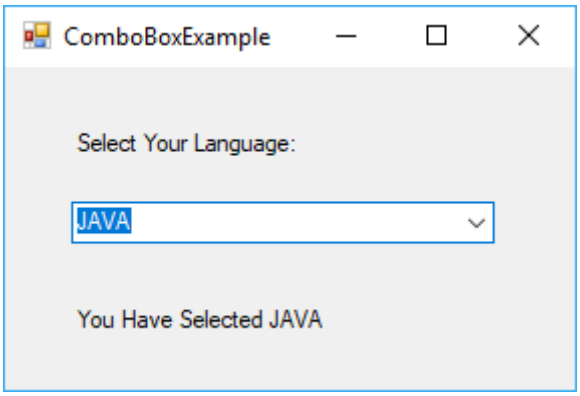
SelectedIndexChanged	Occurs when the SelectedIndex property of a ComboBox control has changed.
SelectionChangeCommitted	Occurs when the selected item has changed and the change appears in the combo box.
StyleChanged	Occurs when the control style changes.
TextChanged	Occurs when the text property value changes.

Other events are Click, KeyUp, KeyDown, KeyPress, MouseMove, MouseUp, MouseDown, MouseEnter events are used in ComboBox control.

Example:

```
Public Class ComboBoxExample

    Private Sub ComboBoxExample_Load(ByVal sender As System.Object, ByVal
    e As System.EventArgs) Handles MyBase.Load
        ComboBox1Language.Items.Add("PHP")
        ComboBox1Language.Items.Add("JAVA")
        ComboBox1Language.Items.Add("C#.NET")
    End Sub

    Private Sub ComboBox1Language_SelectedIndexChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    ComboBox1Language.SelectedIndexChanged
        Label1Language.Text = "You Have Selected " +
        ComboBox1Language.SelectedItem.ToString()
    End Sub
End Class
```

Output:



1.3 Timer Control:

- Timer Control is **used to set time intervals**, this control is visible only at RUN time and not in the design.
- You can change the properties of Timer as needed, right click on it and click on Properties. If you set Enabled to **True**, Timer will start ticking as soon as the form loads. **Default is False**.
- Interval is the frequency of Elapsed events in milliseconds (1000 = 1 second).

Properties:

Property Name	Description
Enabled	Property used to Get or set whether the timer is running.
Interval	Gets or sets the interval, expressed in milliseconds.
CanRaiseEvents	Gets a value indicating whether the component can raise an event or not.

Methods:

Method Name	Description
Start()	Method used to start timer.
Stop()	Method used to stop timer.
OnTick()	Raises the Tick Event.

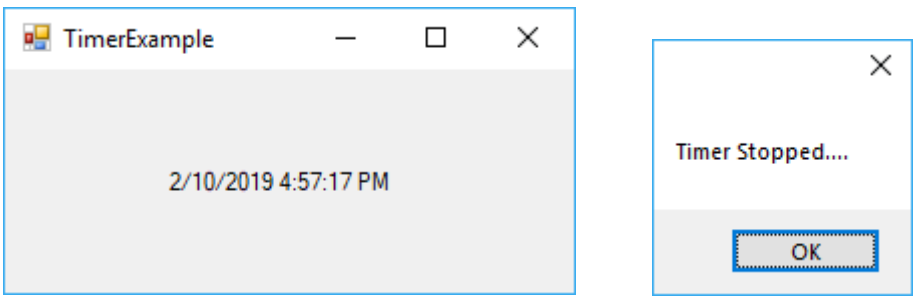
Events:

Event Name	Description
Tick	Triggered when the time interval has elapsed.
Dispose	Occurs when the component is displayed by a call to the disposal method.

Example:

```
Public Class TimerExample
      Dim second As Integer

      Private Sub TimerExample_Load(ByVal sender As System.Object, ByVal e
      As System.EventArgs) Handles MyBase.Load
            Timer1.Interval = 1000
            Timer1.Start() 'Timer starts functioning
      End Sub
      Private Sub Timer1_Tick_1(ByVal sender As System.Object, ByVal e As
      System.EventArgs) Handles Timer1.Tick
            Label1.Text = DateTime.Now.ToString()
            second = second + 1
            If second >= 10 Then
                  Timer1.Stop() 'Timer stops functioning
                  MessageBox.Show("Timer Stopped....")
            End If
      End Sub
End Class
```

Output:



1.4 ScrollBar Control:

- The ScrollBar controls **display vertical and horizontal scroll bars on the form.**
- This is used for navigating through large amount of information, when the information is not fit into the display area.
- There are **two types** of scroll bar controls:
- **HScrollBar** for horizontal scroll bars, and
- **VScrollBar** for vertical scroll bars.
- These are used independently from each other.

Properties:

Property Name	Description
LargeChange	It gets or sets a value to be added to or subtracted from the Value property when the scroll box is moved a large distance.
Maximum	It gets or sets the upper limit of values of the scrollable range.
Minimum	It gets or sets the lower limit of values of the scrollable range.
SmallChange	It gets or sets the value to be added to or subtracted from the Value property when the scroll box is moved a small distance.
Value	It gets or sets a numeric value that represents the current position of the scroll box on the scroll bar control.

Methods:

Method Name	Description
OnClick()	Generates the Click event.
Select()	Activates the control.

Events:

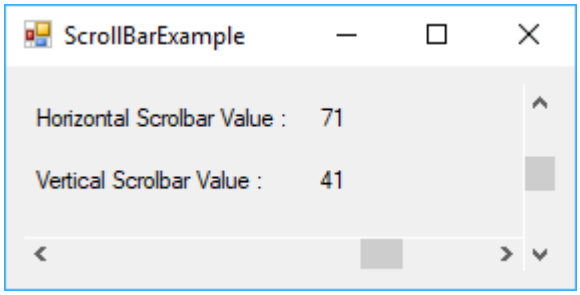
Event Name	Description
Click	Occurs when the control is clicked.
Scroll	Occurs when the control is moved.
DoubleClick	Occurs when the user double-clicks the control.
ValueChanged	Occurs when the Value property changes, either by handling the Scroll event or Programmatically.

Example:

```
Public Class ScrollBarExample

    Private Sub VScrollBar1_Scroll(ByVal sender As System.Object, ByVal e
    As System.Windows.Forms.ScrollEventArgs) Handles VScrollBar1.Scroll

        lblVbar.Text = VScrollBar1.Value
```

```
    End Sub

    Private Sub HScrollBar1_Scroll(ByVal sender As System.Object, ByVal e
    As System.Windows.Forms.ScrollEventArgs) Handles HScrollBar1.Scroll

        lblHbar.Text = HScrollBar1.Value

    End Sub
End Class
```

Output:



1.5 PictureBox Control:

- The PictureBox control is **used for displaying images on the form**.
- The **Image** property of the control allows you to set an image **either at design time or at run time**.
- PictureBox is used to **display graphics** from a bitmap, metafile, icon, jpeg, gif or png file.
- By default PictureBox control is **displayed without any borders**. You can set border by using its **BorderStyle** property.
- PictureBox is not a selectable control, which means that it cannot receive input focus.

Properties:

Property Name	Description
AllowDrop	Specifies whether the picture box accepts data that a user drags on it.
ErrorImage	It gets or specifies an image to be displayed when an error occurs during the image-loading process or if the image load is cancelled.
Image	It gets or sets the image that is displayed in the control.
ImageLocation	It gets or sets the path or the URL for the image displayed in the control.
ClientSize	Gets or Sets the height and width of the client area of the control.
BorderStyle	Indicates that the border style for the control. None, A Single line border, Three Dimensional Border.
SizeMode	Determines the size of the image to be displayed in the control. It which has values: **Normal**: Image Placed at Upper-Left Corner. Image is clipped if it is larger than container.

	StrechImage: Image is Stretched or Shrink to fit in the container. **AutoSize:** The PictureBox is sized equal to the size of image. **CenterImage:** The image is displayed in the center if PictureBox is Larger, If image is larger, then it is placed in center of PictureBox and remaining outer edges are not displayed. **Zoom**: Size is increased or decreased with maintaining the image Size Ratio.

Methods:

Method Name	Description
CancelAsync()	Cancels an asynchronous image load.
Load()	Displays an image in the picture box.
LoadAsync()	Loads image asynchronously.

Events:

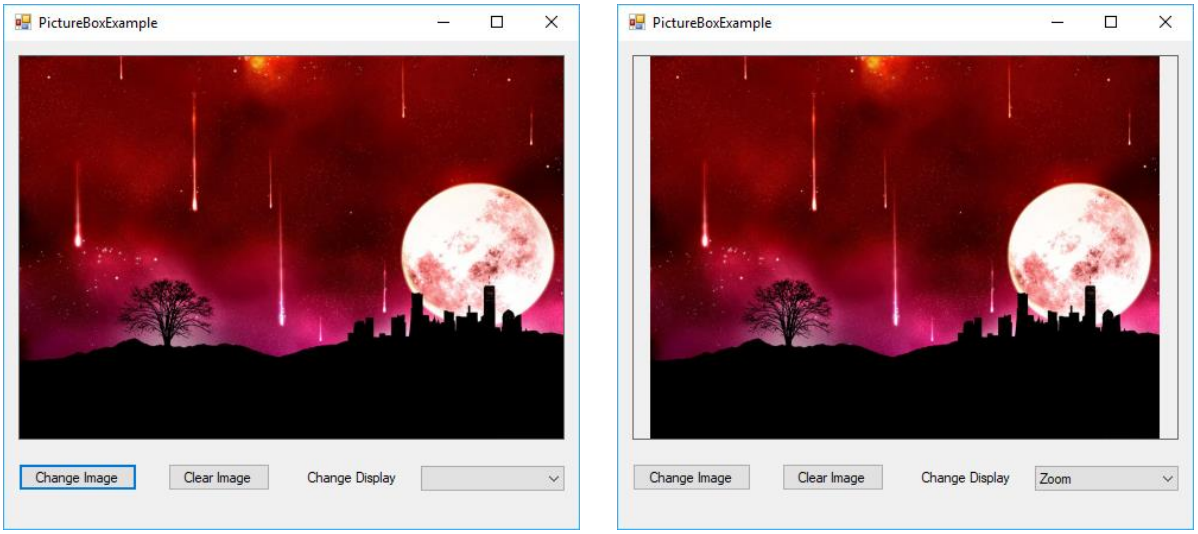
Event Name	Description
LoadCompleted	Occurs when the asynchronous image-load operation is completed, been cancelled, or raised an exception.
LoadProgressChanged	Occurs when the progress of an asynchronous image-loading operation has changed.
ClientSizeChanged	Occurs when the value of ClientSize property Changes.
SizeChanged	Occurs when the Size property value changes.

Example:

```
Public Class PictureBoxExample

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button3.Click
        PictureBox1.Image = Image.FromFile("G:\IMAGES FOR .NET
        PRACTICAL\IMG2.png")
    End Sub

    Private Sub PictureBoxExample_Load(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles MyBase.Load
        ComboBox1.Items.Add("Normal")
        ComboBox1.Items.Add("StreachImage")
        ComboBox1.Items.Add("CenterImage")
        ComboBox1.Items.Add("Zoom")
        ComboBox1.Items.Add("AutoSize")
    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    ComboBox1.SelectedIndexChanged

        If ComboBox1.SelectedItem.ToString() = "Normal" Then
```

```
            PictureBox1.SizeMode = PictureBoxSizeMode.Normal
        End If

        If ComboBox1.SelectedItem.ToString() = "Zoom" Then
            PictureBox1.SizeMode = PictureBoxSizeMode.Zoom
        End If

        If ComboBox1.SelectedItem.ToString() = "CenterImage" Then
            PictureBox1.SizeMode = PictureBoxSizeMode.CenterImage
        End If

        If ComboBox1.SelectedItem.ToString() = "StretchImage" Then
            PictureBox1.SizeMode = PictureBoxSizeMode.StretchImage
        End If

        If ComboBox1.SelectedItem.ToString() = "AutoSize" Then
            PictureBox1.SizeMode = PictureBoxSizeMode.AutoSize
        End If
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button1.Click
        PictureBox1.Image = Nothing
    End Sub
End Class
```

Output:



1.6 RichTextBox Control:

- The RichTextBox control **enables you to display or edit text content including paragraphs.**
- It provides all the **functionality of a TextBox control**; it can handle multiple typefaces, sizes, and attributes, and offers precise control over the margins of the text.
- We can also assign **character and paragraph formatting** to the text within the control.

Properties:

Property Name	Description
AcceptsTab	It gets or sets a value that indicates how the text editing control responds when the user presses the TAB key.
SelectionFont	Gets or sets the font of the current text selected in Control or insertion point.
SelectionColor	Gets or sets the text color of the current text selection or insertion point.
SelectionBullet	Gets or sets a value indicating whether the bullet style is applied to the current selection or insertion point.
SelectionIndent	Gets or sets the length, in pixels, of the indentation of the line where the selection starts.
SelectionRight Indent	The distance (in pixels) between the right edge of the RichTextBox control and the right edge of the text that is selected or added at the current insertion point.
SelectionHanging Indent	Gets or sets the distance between the left edge of the first line of text in the selected paragraph and the left edge of subsequent lines in the same paragraph.
DetectUrls	Gets or sets a value indicating whether or not the RichTextBox will automatically format a Uniform Resource Locator (URL) when it is typed into the control.
BackColor	It gets or sets the background color of the control.
ForeColor	It gets or sets the foreground color of the control.
MaxLength	It gets or sets the maximum number of characters the user can type or paste into the rich text box control.
Text	Gets or sets the current text in the rich text box.
ReadOnly	It gets or sets a value indicating whether text in the text box is read-only.

Methods:

Method Name	Description
Clear()	Clears all text from the text box control.
Copy()	Copies the current selection in the text box to the **Clipboard**
LoadFile()	Loads a rich text format (RTF) or standard ASCII text file into the RichTextBox control.
AppendText()	Appends text to the current text of a text box.
SaveFile()	Saves the contents of the RichTextBox to a rich text format (RTF) file.
Find()	Searches the text of a RichTextBox control for the first instance of a character from a list of characters.

Events:

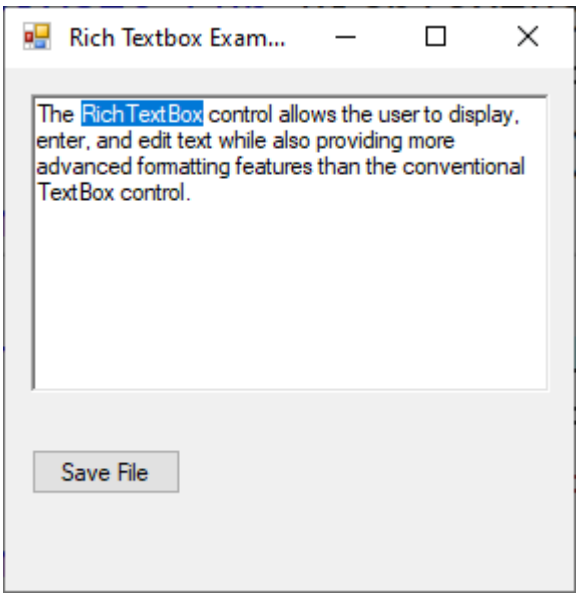
Event Name	Description

Protected	Occurs when the user attempts to modify protected text in the control.
TextChanged	Occurs when content changes in the text element.
LinkClicked	Occurs when the user clicks on the link within the text.

Example:

```
Public Class RichTextBoxExample
      Private Sub RichTextBoxExample_Load(ByVal sender As System.Object,
      ByVal e As System.EventArgs) Handles MyBase.Load
            RichTextBox1.LoadFile("G:\IMAGES FOR .NET
            PRACTICAL\richtextbox.rtf")
            RichTextBox1.Find("RichTextBox")
      End Sub

      Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
      System.EventArgs) Handles Button1.Click
            RichTextBox1.SaveFile("G:\IMAGES FOR .NET
            PRACTICAL\richtextbox.rtf", RichTextBoxStreamType.RichText)
            MessageBox.Show("File Saved Successfully")
      End Sub
End Class
```

Output:



1.7 Masked TextBox Control:

- It provides a mask that **helps the user in entering a value in a particular format.**
- MaskedTextBox provides all the facility of simple TextBox Control.
- It allows user to validate input by specifying mask property.
- **Mask** property is used to specify pattern that must be followed while entering text into MaskedTextBox.
- It does not allow user to enter text that does not comply with the pattern specified in the Mask property.

- Thus MaskedTextBox control is used to restrict invalid input from user.
- The mask determines which characters are allowed at different positions in the text, displaying placeholder characters to help prompt the user and underscores where the user can enter characters.
- If you have define mask for validating input or output, then each character position in masked textbox checks validation you provided.
- You can specify the following input without writing any custom validation logic in your application.
    - Require input characters
    - Optional Input Characters
    - The type of input expected at a given position in the mask, for example, a digit or an alphabetic or alphanumeric character
    - Mask literals or characters like, hyphens (-) in the phone number, or currency symbol in the price
    - Special processing for input characters, for example to convert alphabetic characters to uppercase

Properties:

Property Name	Description
HidePrompt OnLeave	Gets or sets a value indicating whether the prompt characters in the input mask are hidden when the masked textbox loses focus.
BeepOnError	Gets or sets a value indicating whether the masked text box control raises the system beep for each user key stroke that it rejects.
MaskFull	Gets a value indicating whether all required and optional inputs have been entered into the input mask.
TextMaskFormat	Gets or sets a value that determines whether literals and prompt characters are included in the formatted string.
Text	It specifies a current text of the control.
AsciiOnly	Gets or sets a value indicating whether the MaskedTextBox control accepts characters outside of the ASCII character set.
Mask	It represents the format of the input can be accepted by a control.
ReadOnly	It represent that if you want the text box only readable mode, so no one can perform editing on that textbox.
PromptChar	It gets or sets the character used to represent the absence of user input in MaskedTextBox.

Methods:

Method Name	Description
Clear()	Clears all text from the text box control.
AppendText()	Appends text to the current text of a text box.

Copy()	Copies the current selection in the text box to the Clipboard.
SelectAll()	Selects all text in the text box.
ValidateText()	Converts the user input string to an instance of the validating type.

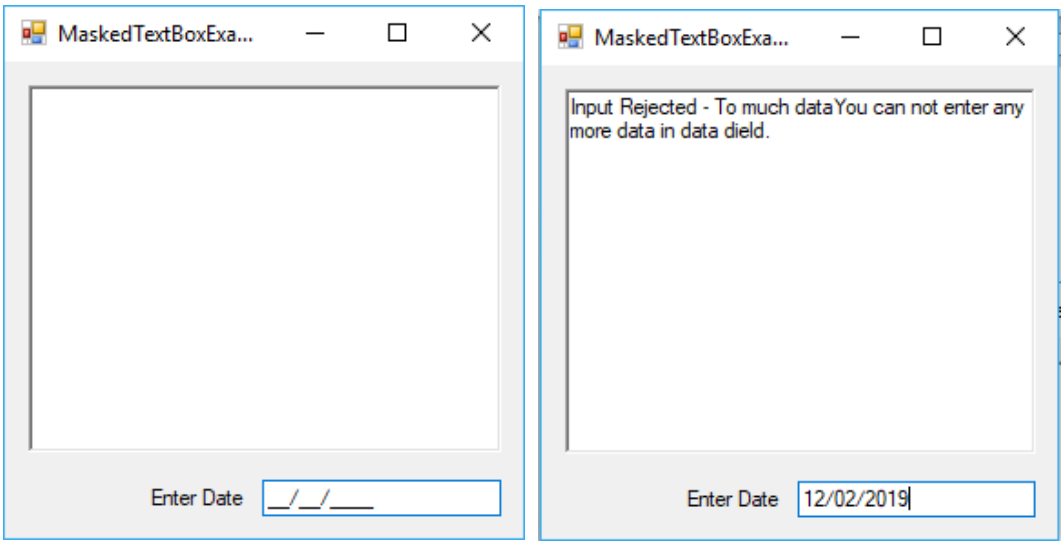
Events:

Event Name	Description
MaskInput Rejected	Occurs when the user's input or assigned character does not match the corresponding format element of the input mask.
MaskChanged	Occurs after the input mask is changed.
TypeValidation Completed	Occurs when MaskedTextBox has finished parsing the current value using the ValidatingType property.
EnabledChanged	Occurs when the Enabled property value has changed.
TextChanged	Occurs when the Text property value changes.

Example:

```
Public Class MaskedTextBoxExample
      Private Sub MaskedTextBoxExample_Load(ByVal sender As System.Object,
      ByVal e As System.EventArgs) Handles MyBase.Load
            MaskedTextBox1.Mask = "00/00/0000"
      End Sub
      Private Sub MaskedTextBox1_MaskInputRejected(ByVal sender As
      System.Object, ByVal e As
      System.Windows.Forms.MaskInputRejectedEventArgs) Handles
      MaskedTextBox1.MaskInputRejected
            If (MaskedTextBox1.MaskFull) Then
                  RichTextBox1.Text = "Input Rejected - To much data"
                  RichTextBox1.AppendText("You can not enter any more data in
                  data field.")
            Else
                  RichTextBox1.Text = "Input Rejected" & vbCrLf
                  RichTextBox1.AppendText("You can only add numeric
                  characters 0-9 into date field.")
            End If
      End Sub
      Private Sub MaskedTextBox1_KeyDown(ByVal sender As System.Object,
      ByVal e As System.Windows.Forms.KeyEventArgs) Handles
      MaskedTextBox1.KeyDown
            RichTextBox1.Text = ""
      End Sub
End Class
```

Output:



1.8 DateTimePicker Control:

- This control is used to allow the **user to select a date and time,** and to **display** that date and time in the specified format.
- If you click the arrow in the DateTimePicker control, it displays a month calendar, like a combo box control.
- The user can make selection by clicking the required date. The new selected value appears in the text box part of the control. The **MinDate** and the **MaxDate** properties allow you to put limits on the date range.

Properties:

Property Name	Description
MaxDate	It gets or sets the maximum date and time that can be selected in the control.
MinDate	It gets or sets the minimum date and time that can be selected in the control.
Calender ForeColor	Gets or sets the foreground color of the calendar.
CalenderFont	Gets or sets the font style applied to the calendar.
CalenderTitle BackColor	Gets or sets the background color of the calendar title.
CalenderTitle ForeColor	Gets or sets the foreground color of the calendar title.
DateTimePicker Format	Gets or sets the format of the date and time displayed in the control. That is Long, Short, Time and Custom.
CustomFormat	Gets or sets the custom date/time format string.
Format	It gets or sets the format of the date and time displayed in the control.
Font	It gets or sets the font of the text displayed by the control.
Value	It gets or sets the date/time value assigned to the control.
ShowCheckBox	It gets or sets a value indicating whether a check box is displayed to the left of the selected date.

ShowUpDown	It gets or sets a value indicating whether a spin button control (also known as an up-down control) is used to adjust the date/time value.
Visible	It gets or sets a value indicating whether the control and all its child controls are displayed.

Methods:

Method Name	Description
ResetText()	Reset the text property to its default value.
Show()	Displays the control to the user.

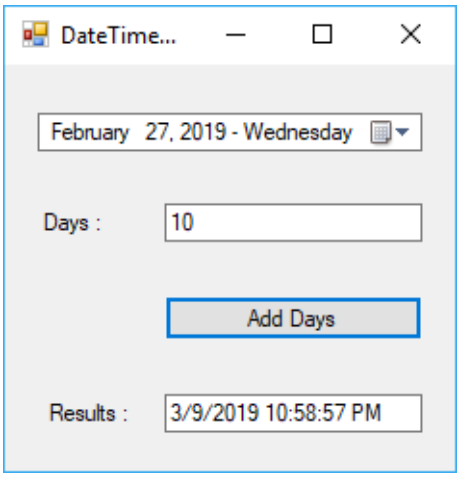
Events:

Event Name	Description
CloseUp	It occurs when the drop-down calendar is dismissed and disappears.
FontChanged	It occurs when the Font property value changes.
FormatChanged	It occurs when the Format property value has changed.
ValueChanged	It occurs when the Value property changes.
VisibleChanged	It occurs when the Visible property value changes.

Example:

```
Public Class DateTimePickerExample

    Private Sub DateTimePickerExample_Load(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles MyBase.Load
        DateTimePicker1.MinDate = New DateTime(2000, 1, 1)
        DateTimePicker1.MaxDate = New DateTime(2019, 12, 31)
        DateTimePicker1.CustomFormat = "MMMM dd, yyyy - dddd"
        DateTimePicker1.Format = DateTimePickerFormat.Custom
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button1.Click
        Dim DT As Date
        DT = DateTimePicker1.Value.AddDays(Val(TextBox1.Text))
        TextBox2.Text = DT.ToString()
    End Sub
End Class
```

Output:



1.9 LinkLabel Control:

- A LinkLabel control is similar to a Label, except **that can display a hyperlink**.
- A LinkLabel control is a **label control that can display a hyperlink**.
- Multiple hyperlink can be specified in the text of the control. Each hyperlink can perform different tasks within an application.
- A LinkLabel control is **inherited from the Label class** so it has all the functionality provided by the Windows Forms Label control.
- LinkLabel control does not participate in user input or capture mouse or keyboard events.

Properties:

Property Name	Description
Links	It gets the collection of links contained within the LinkLabel.
Text	Text Displayed in LinkLabel.
LinkColor	It gets or sets the color used when displaying a normal link.
LinkArea	It gets or sets the range in the text to treat as a link.
LinkVisited	It gets or sets a value indicating whether a link should be displayed as though it were visited.
LinkBehavior	It gets or sets a value that represents the behaviour of a link.
Image	It gets or sets the image that is displayed on a Label.
ActiveLinkColor	It gets or sets the color used to display an active link.
DisabledLinkColor	Gets or sets the color used when displaying a disabled link.
VisitedLinkColor	It gets or sets the color used when displaying a link that that has been previously visited.

Methods:

Method Name	Description
ToString()	Returns a string that represents the current Label

ResetText()	Resets the Text property to its default value.
Focus()	Sets input focus to the control.
Hide()	Hide Control from User.
Show()	Show control to user.
ResetBackColor()	Reset the BackColor property to its default value.
ResetForeColor()	Reset the ForeColor property to its default value.

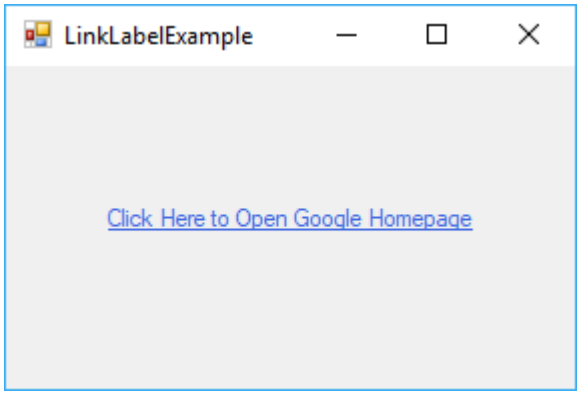
Events:

Event Name	Description
LinkClicked	Occurs when a link is clicked within the control.
TextChanged	Occurs when the Text property value changes.
ForeColorChanged	Occurs when the ForeColor property value changes.

Example:

```
Public Class LinkLabelExample
      Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object,
      ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
      LinkLabel1.LinkClicked
            Process.Start("www.google.co.in")
            LinkLabel1.ActiveLinkColor = Color.Orange
            LinkLabel1.VisitedLinkColor = Color.Green
            LinkLabel1.LinkColor = Color.RoyalBlue
            LinkLabel1.DisabledLinkColor = Color.Gray
      End Sub
End Class
```

Output:



1.10 Checked ListBox Control:

- A CheckedListBox control is **a ListBox control with CheckBox displayed in the left side where user can select a single or multiple items.**
- The CheckedListBox control gives you all the **capability of a list box and also allows you to display a check mark next to the items in the list box**.

Properties:

Property Name	Description
CheckOnClick	It gets or sets a value indicating whether the check box should be toggled when an item is selected.
Checked	It gets or sets a value indicating whether the checkbox is in the checked state.
CheckState	Get or Set the state of Checkbox
CheckedIndices	Collection of checked indexes in this CheckedListBox.
ColumnWidth	It gets or sets the width of columns in a multicolumn ListBox.
Items	It gets the collection of items in this CheckedListBox.
MultiColumn	It gets or sets a value indicating whether the ListBox supports multiple columns.
ScrollAlwaysVisible	It gets or sets a value indicating whether the vertical scroll bar is shown at all times.
SelectedIndex	It gets or sets the zero-based index of the currently selected item in a ListBox.
SelectedItem	It gets or sets the currently selected item in the ListBox.
SelectedValue	Gets or sets the value of the member property specified by the Value Member property.
SelectionMode	Gets or sets a value specifying the selection mode.
Sorted	Displays the list of the content in alphabetic order.
TopIndex	Gets or sets the index of the first visible item in the ListBox.

Methods:

Method Name	Description
SetItemChecked()	Sets the check state of the item at the specified index.
GetSelected()	Returns a value indicating whether the specified item is selected.
GetItemText()	Method used to get the text of an item.
SetSelected()	Selects or clears the selection for the specified item in a ListBox.

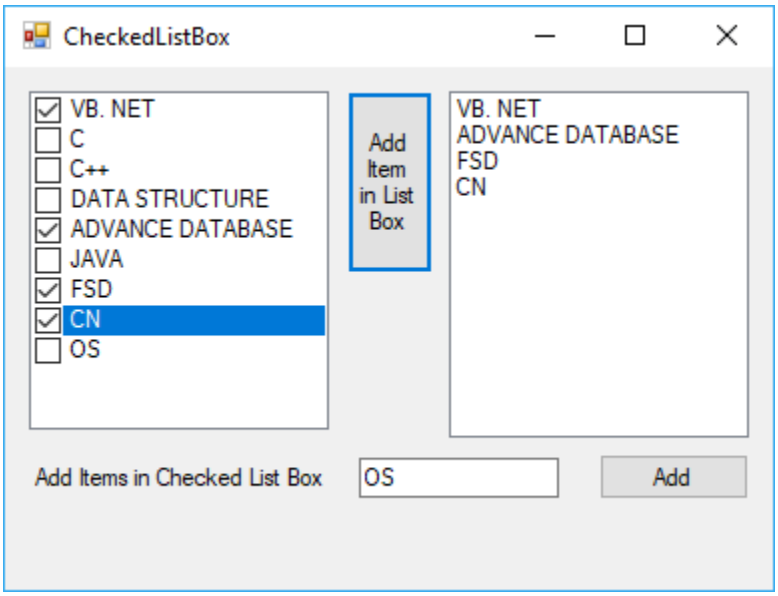
Events:

Event Name	Description
ItemCheck	Occurs when the checked state of an item changes.
CheckStateChanged	Occurs when the Check State property changes.
SelectedValueChanged	Occurs when the SelectedValue property changes.
CheckedChanged	Occurs when the CheckedChanged property changes.
SelectedIndexChanged	Triggered when the SelectedIndex property is changed.
TextChanged	Occurs when the Text property is changed.

Example:

```
Public Class CheckedListBox
      Private Sub CheckedListBox_Load(ByVal sender As System.Object, ByVal e
      As System.EventArgs) Handles MyBase.Load
            CheckedListBox1.Items.Add("VB. NET")
            CheckedListBox1.Items.Add("C")
            CheckedListBox1.Items.Add("JAVA")
            CheckedListBox1.Items.Add("FSD")
            CheckedListBox1.Items.Add("CN")
      End Sub
      Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
      System.EventArgs) Handles Button1.Click
            Dim strItem As String
            For i As Integer = 0 To CheckedListBox1.CheckedItems.Count - 1
                  strItem = CheckedListBox1.CheckedItems(i).ToString()
                  ListBox1.Items.Add(strItem)
            Next
      End Sub

      Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
      System.EventArgs) Handles Button2.Click
            CheckedListBox1.Items.Add(TextBox1.Text)
      End Sub
End Class
```

Output:



1.11 ProgressBar Control:

- It represents a Windows progress bar control.
- It is used to provide **visual feedback to your users about the status of some task**. It shows a bar that **fills in from left to right as the operation progresses**.
- The ProgressBar control is typically **used when an application performs tasks such as copying files or printing documents**.
- To a user the **application might look unresponsive if there is no visual clue**. In such cases, using

the ProgressBar allows the programmer to **provide a visual status of progress**.

Properties:

Property Name	Description
MarqueeAnimation Speed	Gets or sets the time period, in milliseconds that it takes the **progress block to scroll across the progress bar**.
Maximum	Gets or sets the **maximum value of the range of the control**.
Minimum	Gets or sets the **minimum value of the range of the control.**
Step	Gets or sets the **amount by which a call to the PerformStep** method increases the **current position of the progress bar.**
Value	Gets or sets the **current position of the progress bar.**

Methods/ Functions:

Method Name	Description
PerformStep()	Increments the value by the specified step.
Increment()	**Increments the current position of the ProgressBar** control by specified amount.

Events:

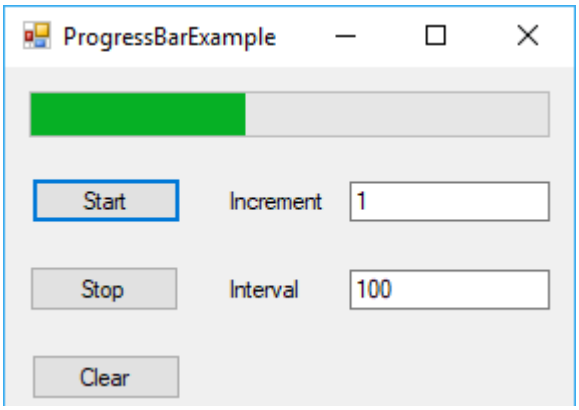
Event Name	Description
BackgroundImage Changed	Occurs when the value of the **BackgroundImage property changes.**
Click	Occurs when the **control is clicked.**
Enter	Occurs when **focus enters the control.**
KeyDown	Occurs when the user **presses a key while the control has focus.**
KeyUp	Occurs when the user **releases a key while the control has focus.**
TextChanged	Occurs when the **Text property changes.**

Example:

```
Public Class ProgressBarExample

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button1.Click
        Timer1.Start()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button2.Click
        Timer1.Stop()
    End Sub
```

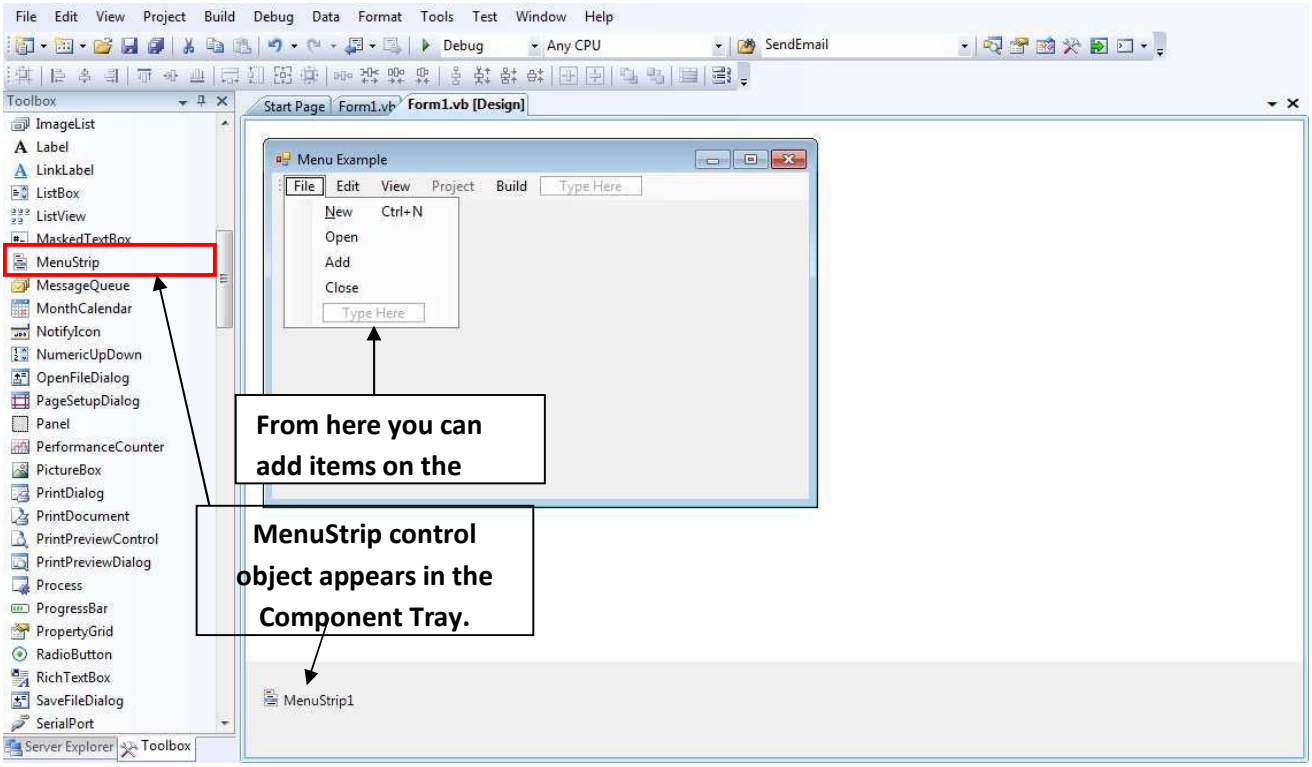
```
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button3.Click
        Timer1.Stop()
        ProgressBar1.Value = 0
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Timer1.Tick
        ProgressBar1.Increment(TextBox1.Text)
        Timer1.Interval = (TextBox2.Text)
    End Sub
End Class
```

Output:



2. WORKING WITH MENUS:

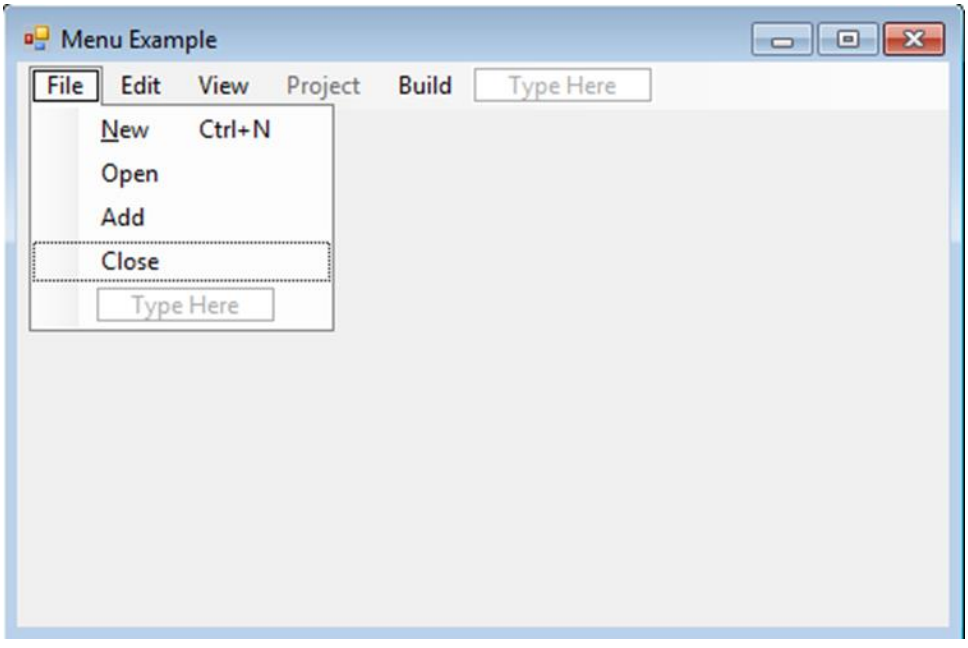
- Menus are parts of many applications and it **provides easy way for navigating through the application**. It provides user **grouping command in single group**, so user can easily use it.
- Menus are controls that allows user to **make selections, saving space in windows application**.
- A menu contains a **list of related commands**.
- Example: A menu will require in application so that you can perform actions such as **opening child forms, copying and pasting data, and arranging windows, etc.**
- In VB.Net the **MenuStrip and ContextMenuStrip** control provides functions of menu related controls.
- ContextMenuStrip represents a **shortcut menu also known as popup menu.**



2.1 Add menu and Sub-Menu items:

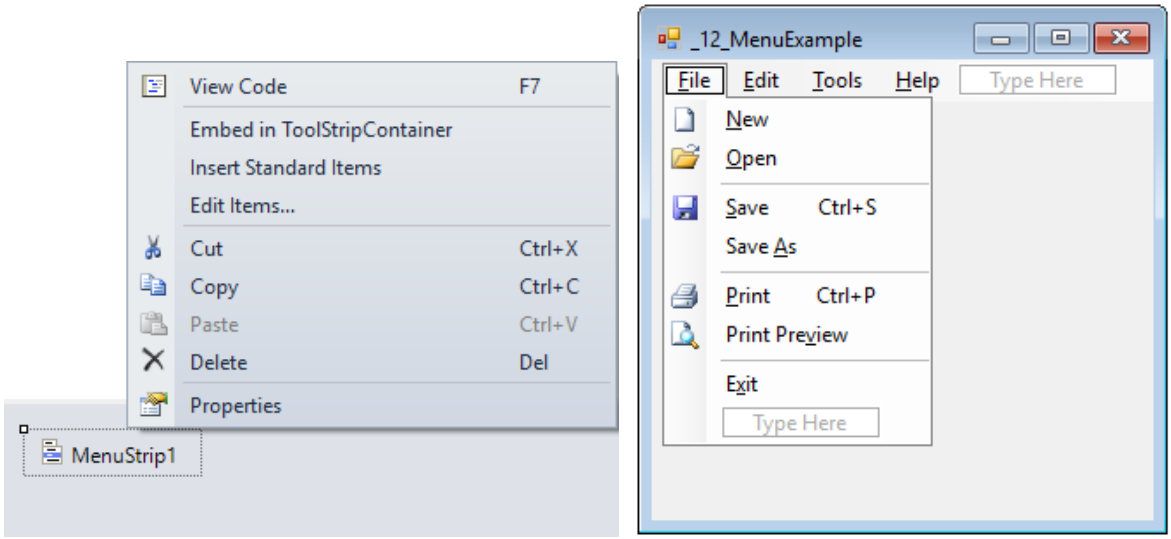
Following are the steps to add menu and sub menu in your application:

- **Drag and drop** or double click on a **MenuStrip control**, to add it to the form.
- Click the **Type Here** text to open a text box and **enter the names of the menu items or sub-menu items you want**. When you add a sub-menu, another text box with **'Type Here'** text opens below it.
- Complete the menu structure shown in figure.
- Add a sub menu **Close** under the **File menu**.

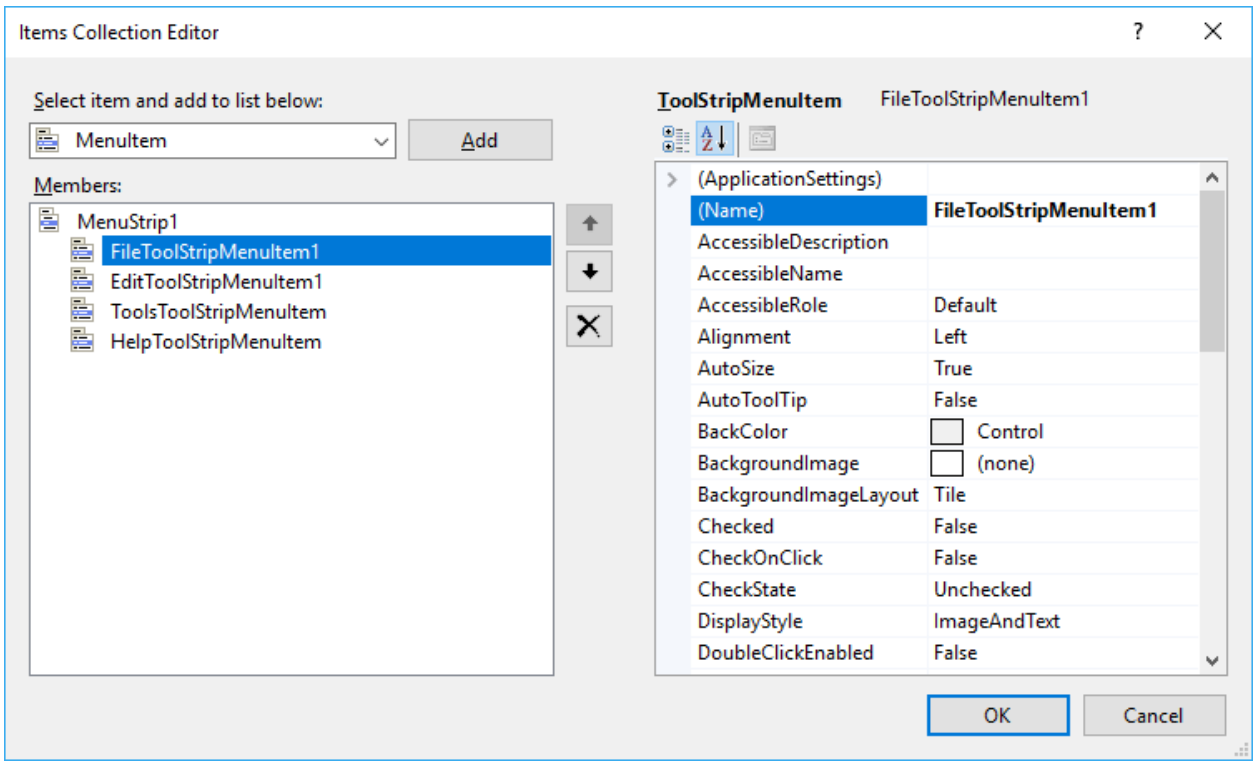


- You can add **standard items** by clicking on MenuStrip1 control and then click on the **Insert**

Standard Item tag. This will add standard controls of menu strip in the form.



- Right click on the MenuStrip1 control, and select **Edit Item Option for make changes in menu strip property.**



Properties:

Property Name	Description
Items	It gets **all the items that belong to a ToolStrip.**
GripStyle	It gets or sets the **visibility of the grip used** to reposition the control.
ShowItemToolTips	It gets or sets a value indicating whether **ToolTips are shown for the MenuStrip.**

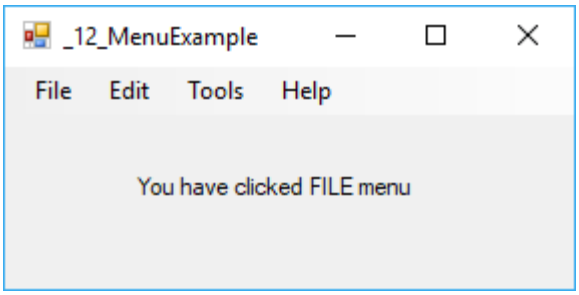
Events:

Event Name	Description
Click	Occurs when the **control is clicked.**
ItemClicked	Occurs when the **ToolStripItem** is clicked.
MenuActivate	Occurs when the **user accesses the menu with the keyboard or mouse.**
MenuDeactivate	Occurs when the **MenuStrip is deactivated.**

Example:

```
Public Class _12_MenuExample
      Private Sub FileToolStripMenuItem1_Click(ByVal sender As
      System.Object, ByVal e As System.EventArgs) Handles
      FileToolStripMenuItem1.Click
            Label1.Text = "You have clicked FILE menu"
      End Sub
      Private Sub EditToolStripMenuItem1_Click(ByVal sender As
      System.Object, ByVal e As System.EventArgs) Handles
      EditToolStripMenuItem1.Click
            Label1.Text = "You have clicked EDIT menu"
      End Sub
      Private Sub ToolsToolStripMenuItem_Click(ByVal sender As
      System.Object, ByVal e As System.EventArgs) Handles
      ToolsToolStripMenuItem.Click
            Label1.Text = "You have clicked TOOLS menu"
      End Sub
      Private Sub SaveToolStripMenuItem_Click(ByVal sender As System.Object,
      ByVal e As System.EventArgs) Handles SaveToolStripMenuItem.Click
            Label1.Text = "You have clicked SAVE menu"
      End Sub
End Class
```

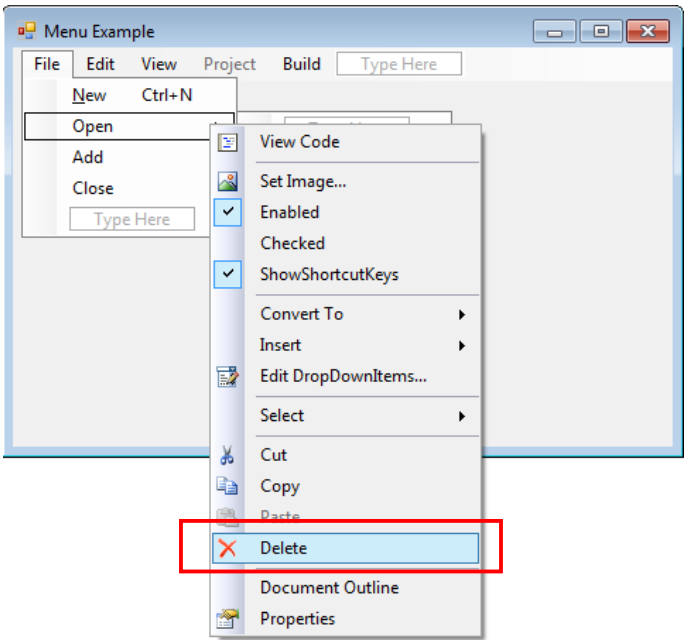
Output:



2.2 Delete Menu and Sub-Menu items:

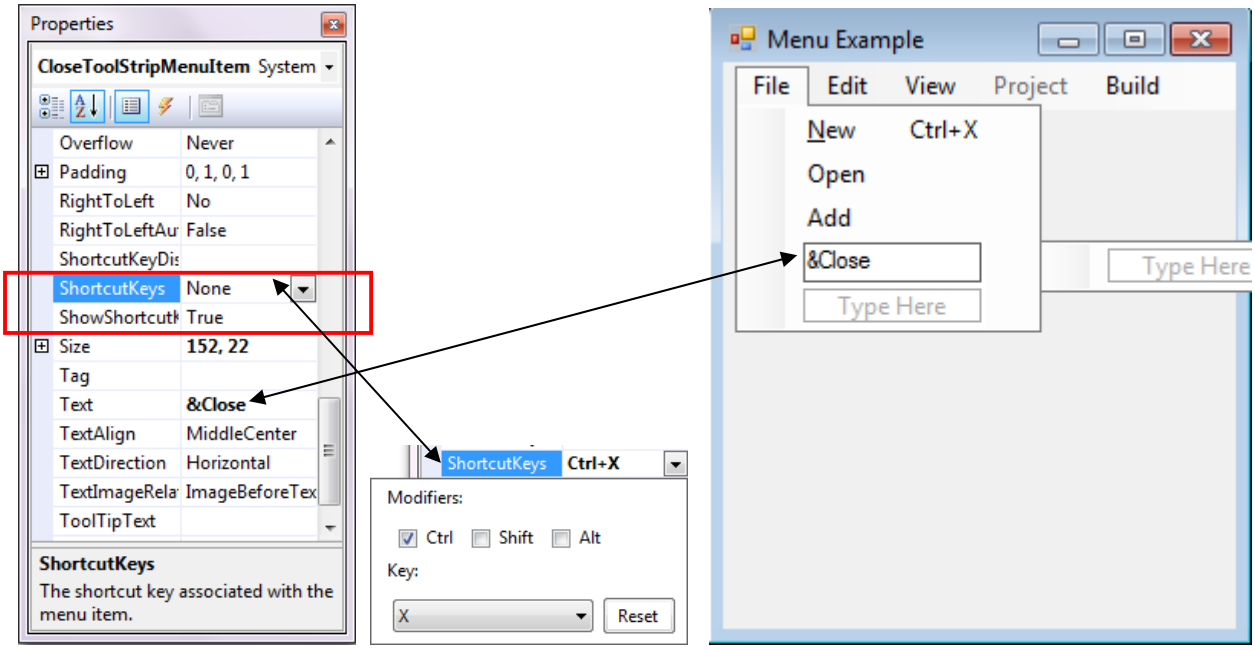
Following are the steps to delete menu and sub menu from your application:

- To delete a menu and sub menu from application, just **Right click** on the **MenuItem** which you want to delete.
- It will open a one dialog box which is shown in Figure.
- Click on the **Delete** option from the opened menu to delete a particular **Menuitem** from the menu.



2.3 Assigning Shortcuts to Menu and Sub-Menu items:

- A **key combination shortcut** is one that appears at the end of a menu item (Ctrl + X, for example). You can easily **add this option to your own programs.** Following are the steps to assign shortcut key to menuitem.
- In Design, **select the Close** item on your created menu and look at the **properties box** on the right side.
- Locate the **ShortcutKeys item** on the properties box.
- The **Modifier is the key you press with your shortcut**. For example, the **CTRL key then the "X"** key on your keyboard. Place a check inside the **Ctrl box**. Then select the letter **"X"** from the Key dropdown list.



Another way to add shortcut to menu item:

- To add a **new menu** or edit existing one, just click the **menu entry and type.**
- You can set a **shortcut by placing an ampersand** in **front of the character** you want to use as a keyboard accelerator e.g. **&Close**. Now if type **Alt + C** at runtime, the program closes itself.

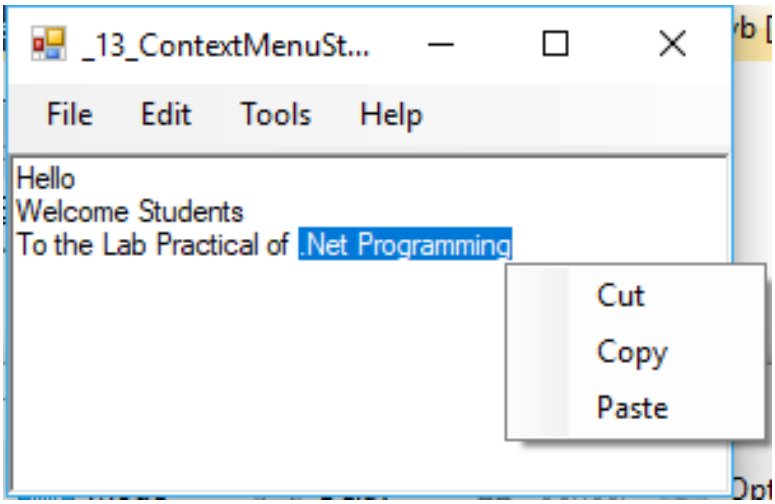
2.4 Creating Context Menu (Popup Menu):

- A **PopUp menu is a menu that is displayed when user right click** either on a form or on a control. It is also known as **Context Menu or Floating Menu.**
- A PopUp menu is useful to represents list of **commonly used commands for form or control.**
- For **create** a **PopUp menu** in application, the **ContextMenuStrip control** which represents a shortcut menu that pops up over controls, **usually when you right click them.** They appear in **context of some specific controls**, so are called context menus. **For example, Cut, Copy or Paste options.**
- To create a Content/Popup menu with the menu options Cut, Copy and Paste. Do the following steps:
  - **Drag and Drop or double click** on a ControlMenuStrip control to add it to the form.
  - Add the **menu items, Cut, Copy and Paste to it.**
  - Add a **RichTextBox control on the form.**
  - Set the **ContextMenuStrip property of the RichTextBox to ContextMenuStrip1** using the **properties window.**

Example:

```
Public Class _13_ContextMenuStripExample

    Private Sub CutToolStripMenuItem_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles HelloToolStripMenuItem.Click
        RichTextBox1.Cut()
    End Sub

    Private Sub CopyToolStripMenuItem1_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    CopyToolStripMenuItem1.Click
        RichTextBox1.Copy()
    End Sub

    Private Sub PasteToolStripMenuItem1_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    PasteToolStripMenuItem1.Click
        RichTextBox1.Paste()
    End Sub

End Class
```

Output:



3. GTU QUESTIONS:

Sr. No	Questions	Marks
	18/02/2021	
1	Develop program using Scrollbar control.	4
2	Give Example of List Box control with Add(), AddRange(),Insert() method.	3
3	Describe POPUP menu with example.	3
4	Explain Masked Textbox control with its important properties and event.	4
5	Explain Selected index changed event of Combobox with suitable example.	4
6	Explain RichTextBox, ProgressBar and PictureBox with example.	7
7	Explain Menu control with suitable example.	4
	29/10/2020	
1	List Timer control properties, method and events.	2
2	Compare Textbox and Richtextbox controls.	3
3	Define DropDownStyle property of ComboBox.	3
4	Describe Listbox control with properties, method and event.	4
5	Develop program using Scrollbar control.	4
6	Describe Masked Text box with example.	3
7	Write step to create MENUS in vb.net.	3
	16/11/2019	
1	Explain DropDownStyle property of a combobox control.	2
2	Explain important property and event of Timer control in brief.	2
3	Explain Text and Value Properties of DateTime Picker Control with Example	3
4	Give Example of ListBox control with Add(),AddRange(),Insert() method.	3
5	Write down code for stop watch using Timer control	4
6	List out properties of Richtext box and explain any two with example.	4
7	Design one application using scroll bars and write code.	4

8	Explain Selected index changed event of Combobox with suitable example.	4
9	Explain Menu control with suitable example.	4
10	Explain Masked TextBox Control with suitable example.	3
17/05/2019		
1	Differentiate between ListBox and ComboBox.	4
2	Explain Timer control with example.	3
3	Distinguish between Textbox and Richtextbox control.	3
4	Design a calculator to find simple interest making use of VB.Net controls.	3
28/11/2018		
1	Write difference between Listbox control and Combo box control	3
2	Explain Timer control with its important property and event.	3
3	Explain Date Time Picker control with its important property and event.	3
4	Explain scroll bar control with its important properties and event.	3
5	Explain Masked Textbox control with its important properties and event.	4
6	Write a code for simple calculator.	7
7	Explain difference between Context menu and Main menu.	2
04/05/2018		
1	Explain DateTimePicker control.	2
2	Differentiate between TextBox and RichTextBox.	2
3	Explain Mask textbox in brief.	2
4	Explain the important Properties of timer.	2
5	Explain I) visible ii) Auto size iii) Enable Properties.	3
6	Write a program making use of following controls (i) Label (ii)Textbox (iii) Button (iv) ListBox.	4
03/05/2017		
1	What is the use of DateTimePicker Control.	2
2	Write four differences between TextBox and RichTextBox controls.	2
3	When should we use the Timer Control?	2
4	Design a form with two Listboxes with multiple selection and two command buttons. Write code to move all selected items from one listbox to other listbox with help of command buttons.	4
5	Create a form with one textbox and one Listbox. Add command buttons for Add, Count and Exit. Write appropriate code for command buttons.	4
6	Differentiate between ListBox and ComboBox.	4
7	Give some examples of Formatting DateTime.	3
8	How does Context Menu differ from Main Menu?	2