Unit 4

Applications Development on .Net

VB.NET Form Controls

A **Form** is used in VB.NET to create a form-based or window-based application. Using the form, we can build a attractive user interface. It is like a container for holding different control that allows the user to interact with an application. The controls are an object in a form such as buttons, Textboxes, Textarea, labels, etc. to perform some action. However, we can add any control to the runtime by creating an instance of it.

A Form uses a **System.Windows.Form** namespace, and it has a wide family of controls that add both forms and functions in a Window-based user interface.

VB.NET Form Properties

The following are the most important list of properties related to a form. And these properties can be set or read while the application is being executed.

Properties	**Description**
BackColor	It is used to set the background color for the form.
BackgroundImage	It is used to set the background image of the form.
Cursor	It is used to set the cursor image when it hovers over the form.
AllowDrop	Using the AllowDrop control in a form, it allows whether to drag and drop on the form.
Font	It is used to get or set the font used in a form.
Locked	It determines whether the form is locked or not.
FormBorderStyle	It is used to set or get border style in a form.
Text	It is used to set the title for a form window.
MinimizeBox	MinimizeBox It is used to display the minimum option on the title bar of the form.
IsMDIChild	It is used to authenticate whether a form is a container of a Multiple Document Interface (MDI) child form.
Autoscroll	It allows whether to enable auto-scrolling in a form.
MaximizeBox	It is used to display the maximum option on the title bar of the form.
MaximumSize	It is used to set the maximum height and width of the form.
Language	It is used to specifies the localized language in a form.
AcceptButton	It is used to set the form button if the enter key is pressed.

Top, Left	It is used to set the top-left corner coordinates of the form in pixel.
Name	It is used to define the name of the form.
MinimumSize	It is used to set the minimum height and width of the form.
Enabled	It uses the True or False value to enable mouse or keyboard events in the form.
TopMost	It uses a Boolean value that represents whether you want to put the window form on top of the other form. By default, it is False.

Form Events

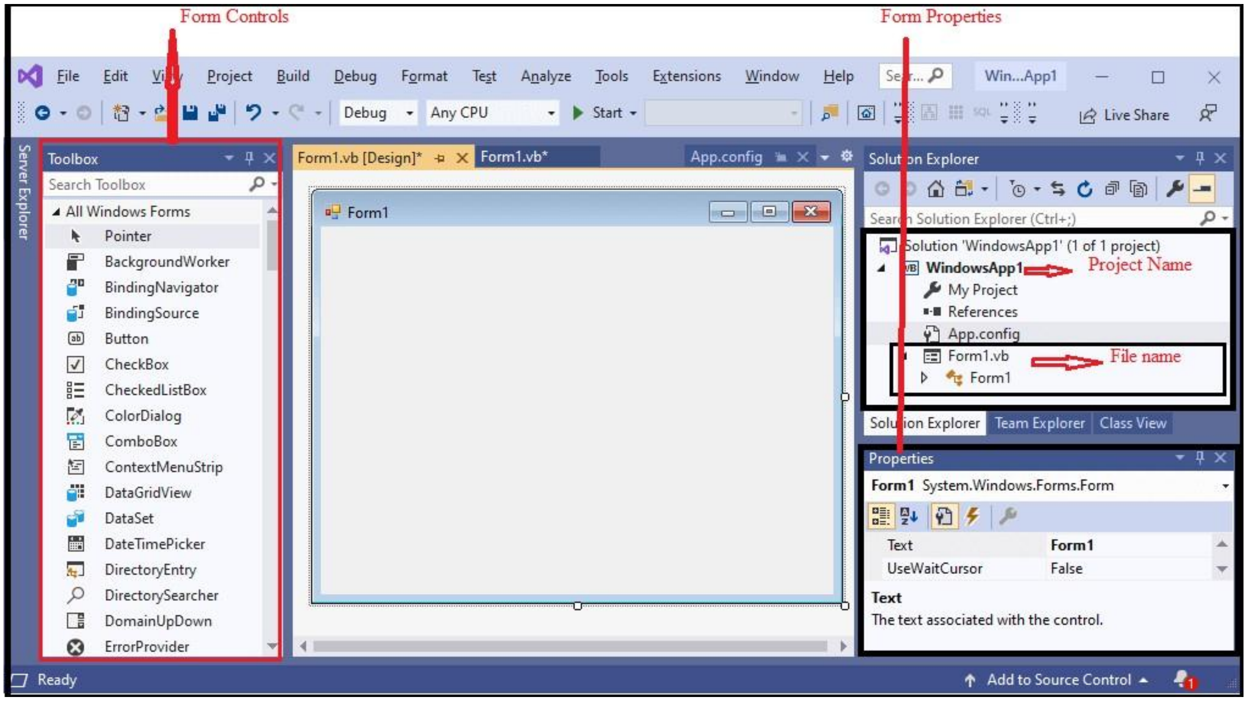
The following are the most important list of events related to a form.

Events	**Description**
Activated	An activated event is found when the user or program activates the form.
Click	A click event is active when the form is clicked.
Closed	A closed event is found before closing the form.
Closing	It exists when a form is closing.
DoubleClick	DoubleClick The DoubleClick event is activated when a user double clicks on the form.
DragDrop	A DragDrop event is activated when a drag and drop operation is performed.
MouseDown	A MouseDown event is activated when the mouse pointer is on the form, and the mouse button is pressed.
GotFocus	A GotFocus event is activated when the form control receives a focus.
HelpButtonClicked	It is activated when a user clicked on the help button.
KeyDown	A KeyDown event is activated when a key is pressed while focussing on the form.
KeyUp	A KeyUp event is activated when a key is released while focusing on the form.
Load	The load event is used to load a form before it is first displayed.
LostFocus	It is activated when the form loses focus.
MouseEnter	A MouseEnter event is activated when the mouse pointer enters the form.
MouseHover	A MouseHover event is activated when the mouse pointer put on the form.
MouseLeave	A MouseLeave event is activated when the mouse pointer leaves the form surface.
Shown	It is activated whenever the form is displayed for the first time.
Scroll	A Scroll event is activated when a form is scrolled through a user or code.
Resize	A Resize event is activated when a form is resized.

Move	A Move event is activated when a form is moved.

For creating a Windows Forms application in VB.NET, we need to follow the following steps in Microsoft Visual Studio.

- GoTo File Menu.
- Click on New Project.
- Click on Windows Forms App or Application
- And finally, click on the 'Create' button to create your project, and then, it displays the following window form with a name Form1.
- Now create a simple program of Windows form control in VB.NET.



Form1 System.Windows.Forms.Form

Text Form1

UseWaitCursor False

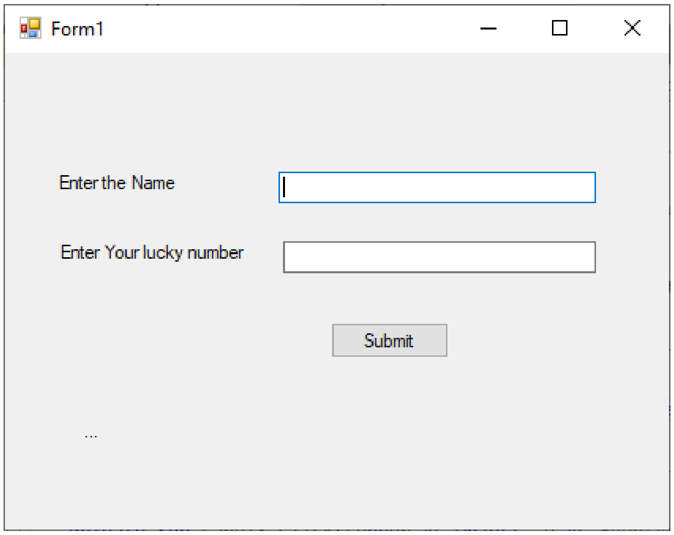
Text

The text associated with the control.

Form1.vb

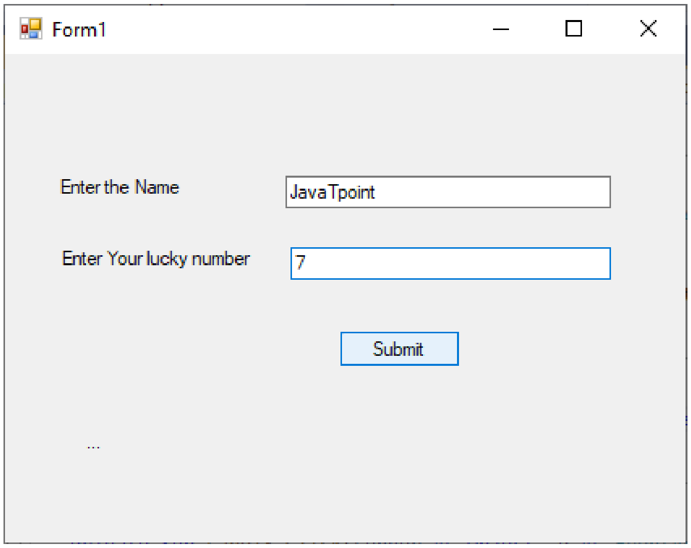
```
Public Class Form1

' Create nameStr and num variables

    Dim nameStr As String

    Dim num As Integer

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    End Sub

   ' It is Label1

    Private Sub Label1_Click(sender As Object, e As EventArgs) Handles Label1.Click

    End Sub

  ' It is TextBox1 for inserting the value.

    Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles TextBox1.TextChanged

    End Sub

    ' It is Label2

    Private Sub Label2_Click(sender As Object, e As EventArgs) Handles Label2.Click

    End Sub

' It is a Button1 for transferring the control.

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

        nameStr = TextBox1.Text

        num = TextBox2.Text

        Label3.Text = "You have entered the Name " & nameStr + " Number " & num

    End Sub

    ' It is TextBox2 for inserting the value.

    Private Sub TextBox2_TextChanged(sender As Object, e As EventArgs) Handles TextBox2.TextChanged

    End Sub

' It is label3

    Private Sub Label3_Click(sender As Object, e As EventArgs) Handles Label3.Click

    End Sub

End Class
```

Output: Now enter the following details in the form:



Enter the Name

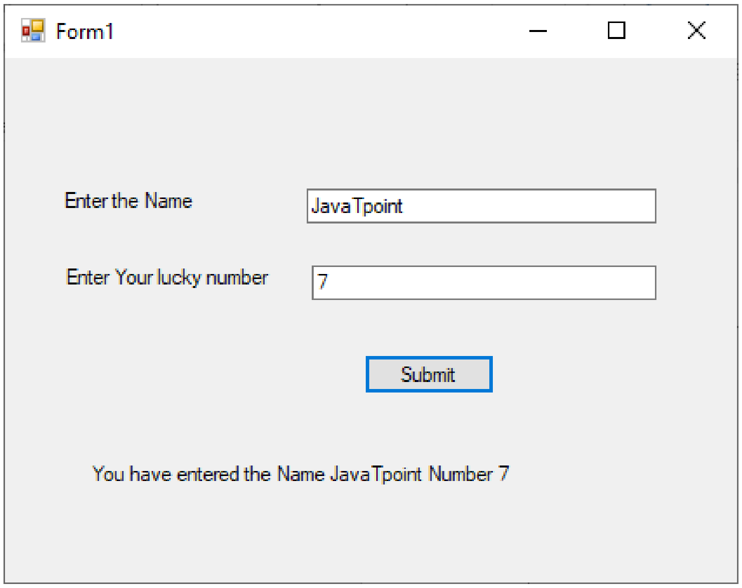
Enter Your lucky number



Enter the Name JavaTpoint

Enter Your lucky number 7

- After filling all the details, click on the **Submit** button. After that, it shows the following Output:



Enter the Name JavaTpoint

Enter Your lucky number 7

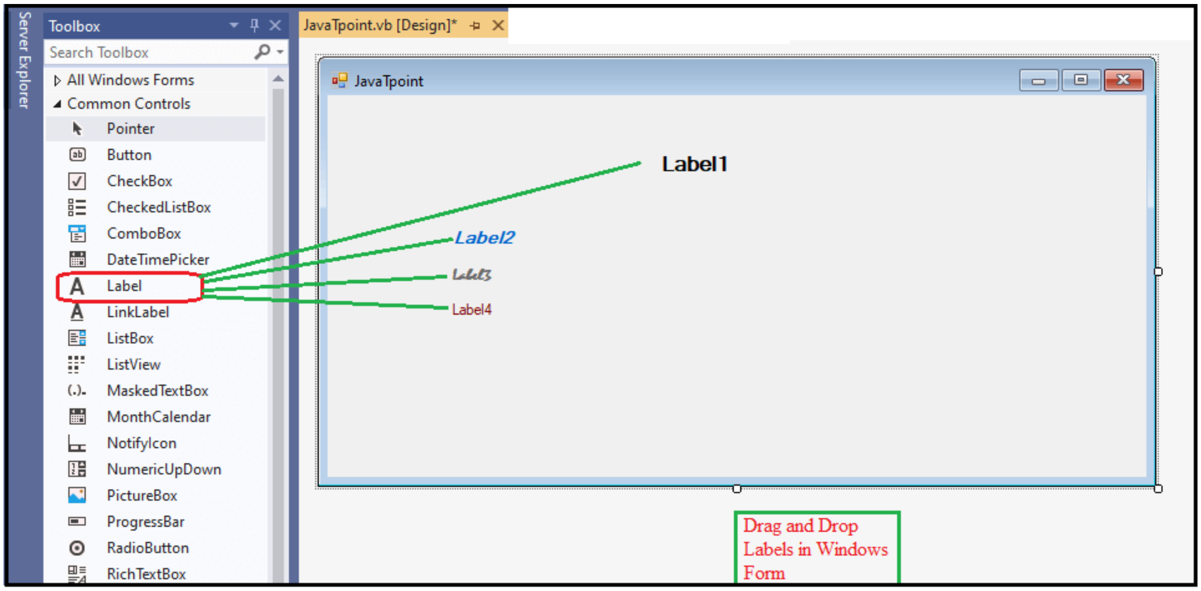
You have entered the Name JavaTpoint Number 7

VB.NET Label Control

In VB.NET, a label control is used to display descriptive text for the form in control. It does not participate in user input or keyboard or mouse events. Also, we cannot rename labels at runtime. The labels are defined in the class System.Windows.Forms namespace.

Let's create a label in the VB.NET Windows by using the following steps:

Step 1: We have to drag the Label control from the Toolbox and drop it on the Windows form, as shown below.



Step 2: Once the Label is added to the form, we can set various properties to the Label by clicking on the Label control.

VB.NET Label Properties

Properties	**Description**
AutoSize	As the name defines, an AutoSize property of label control is used to set or get a value if it is automatically resized to display all its contents.
Border Style	It is used to set the style of the border in the Windows form.
PreferredWidth	It is used to set or get the preferred width for the Label control.
Font	It is used to get or set the font of the text displayed on a Windows form.
PreferredHeight	It is used to set the height for the Label Control.
TextAlign	It is used to set the alignment of text such as centre, bottom, top, left, or right.
ForeColor	It is used to set the color of the text.
Text	It is used to set the name of a label in the Windows Form.
ContextMenu	It is used to get or sets the shortcut menu associated with the Label control.
DefaultSize	It is used to get the default size of the Label control.
Image	It is used to set the image to a label in Windows Form.
ImageIndex	It is used to set the index value to a label control displayed on the Windows form.

VB.NET Label Events

Events	Description
AutoSizeChanged	An AutoSizeChanged event occurs in the Label control when the value of AutoSize property is changed.
Click	Click event is occurring in the Label Control to perform a click.
DoubleClick	When a user performs a double-clicked in the Label control, the DoubleClick event occurs.
GotFocus	It occurs when the Label Control receives focus on the Window Form.
Leave	The Leave event is found when the input focus leaves the Label Control.
TabIndexChanged	It occurs when the value of Tabindex property is changed in the Label control.
ControlRemoved	When the control is removed from the Control.ControlCollection, a ControlRemoved event, occurs.
TabStopChanged	It occurs when the property of TabStop is changed in the Label Control.
BackColorChanged	A BackColorChanged event occurs in the Label control when the value of the BackColor property is changed.
ControlAdded	When a new control is added to the Control.ControlCollection, a ControlAdded event occurs.
DragDrop	A DragDrop event occurs in the Label control when a drag and drop operation is completed.

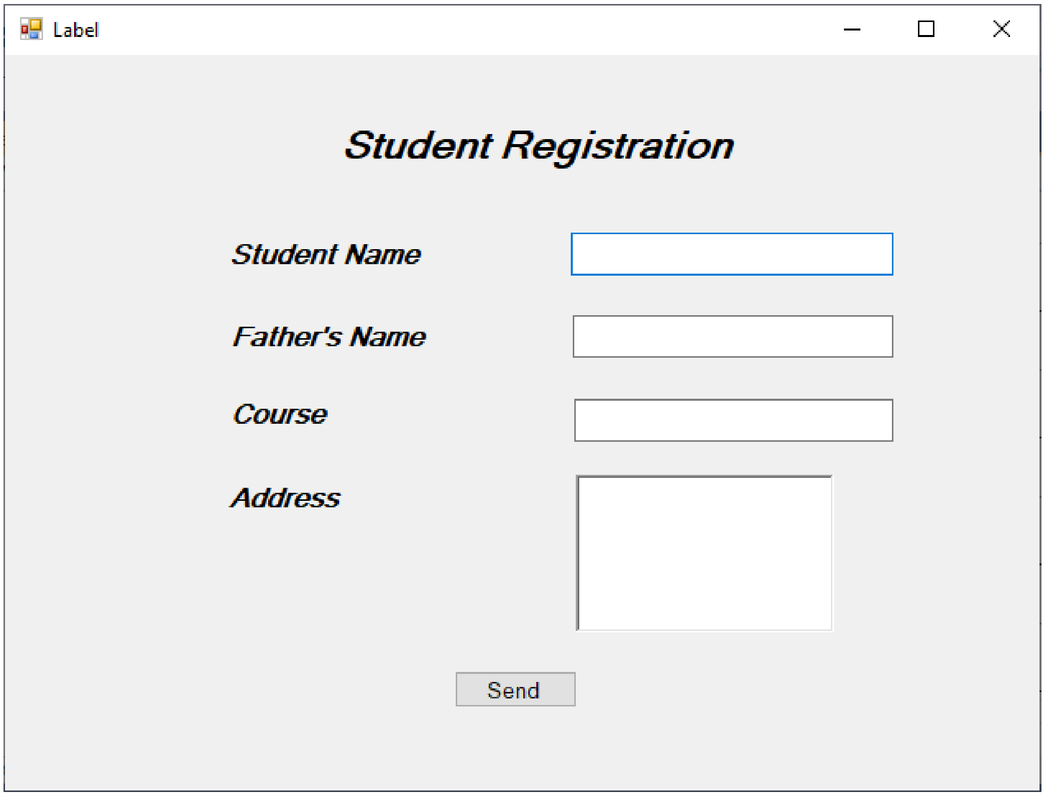
Furthermore, we can also refer to the VB.NET Microsoft documentation to get a complete list of Label properties and events.

Let us create a program to display the Label controls in VB.NET.

Label.vb

```
Public Class Label
    Private Sub Label_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.Text = "javatpoint.com" 'Set the title for a Windows Form
        Label1.Text = "Student Registration"
        Label1.Font = New Font("Microsoft Sans Serif", "style = Bold", "Italic", 18)  ' Set Font style
        Label2.Text = "Student Name"
        Label2.Font = New Font("Microsoft Sans Serif", "style = Bold", "Italic", 12)
        Label3.Text = "Father's Name"
        Label3.Font = New Font("Microsoft Sans Serif", "style = Bold", "Italic", 12)
        Label4.Text = "Course "
        Label4.Font = New Font("Microsoft Sans Serif", "style = Bold", "Italic", 12)
        Label5.Text = "Address"
        Label5.Font = New Font("Microsoft Sans Serif", "style = Bold", "Italic", 12)
        Button1.Text = "Send"
        TextBox1.Text = " "
        TextBox2.Text = " "
        TextBox3.Text = " "
        RichTextBox1.Text = " "
    End Sub
End Class
```

Output:



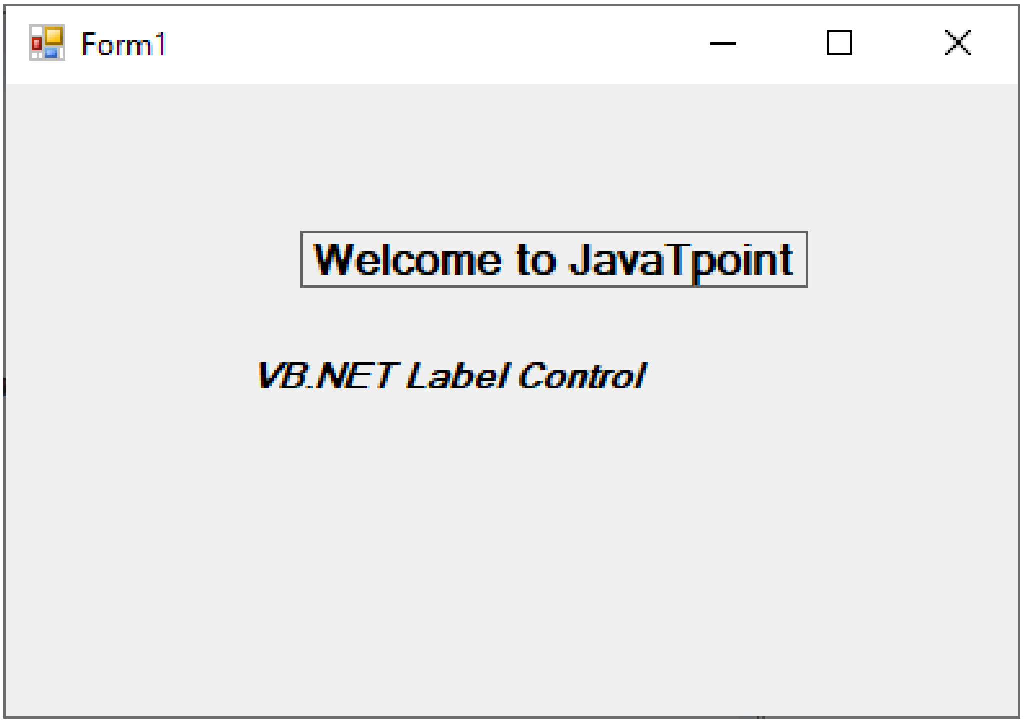
We have created 5 Labels on the Windows Form by using drag and drop operation in the above output.

Example2: Write a program to display only Labels on Windows forms.

Form1.vb

```
Public Class Form1
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'Set Text for Label1 and their property
        Label1.Text = ?Welcome to JavaTpoint?
        Label1.BorderStyle = BorderStyle.FixedSingle
        Label1.TextAlign = ContentAlignment.MiddleCenter
        'Set Text for Label2
        Label2.Text = ? VB.NET Label Control?
    End Sub
End Class
```

Output:



Welcome to JavaTpoint

VB.NET Label Control

VB.NET TextBox Control

A TextBox control is used to display, accept the text from the user as an input, or a single line of text on a VB.NET Windows form at runtime. Furthermore, we can add multiple text and scroll bars in textbox control. However, we can set the text on the textbox that displays on the form.

Let's create a TextBox control in the VB.NET Windows form by using the following steps:

Step 1: We have to drag the TextBox control from the Toolbox and drop it on the Windows form, as shown below.

Step 2: Once the TextBox is added to the form, we can set various properties of the TextBox by clicking on the TextBox control.

Properties	Description
AutoCompleteMode	It is used to get or set a value that indicates how the automatic completion works for the textbox control.
Font	It is used to set the font style of the text displayed on a Windows form.
Lines	It is used to set the number of lines in a TextBox control.
CharacterCasing	It is used to get or set a value representing whether the TextBox control can modify the character's case as they typed.
Multiline	It is used to enter more than one line in a TextBox control, by changing the Multiline property value from False to True.
AcceptsReturn	It is used to get or set a value that indicates whether pressing the enter button in a multiline textbox; it creates a new line of text in control.
PasswordChar	It is used to set the password character that can be a mask in a single line of a TextBox control.
PreferredHeight	It is used to set the preferred height of the textbox control in the window form.
ScrollBars	It is used to display a scrollbar on a multiline textbox by setting a value for a Textbox

	control.
Text	It is used to get or set the text associated with the textbox control.
Visible	The Visible property sets a value that indicates whether the textbox should be displayed on a Windows Form.
WordWrap	The WordWrap properties validate whether the multiline Textbox control automatically wraps words to the beginning of the next line when necessary.

VB.NET TextBox Events

Events	**Description**
Click	When a textbox is clicked, a click event is called in the textbox control.
CausesValidationChanged	It occurs in the TextBox Control when the value of CauseValidation property is changed.
AcceptTabsChanged	It is found in the TextBox control when the property value of the AcceptTab is changed.
BackColorChanged	It is found in the TextBox Control when the property value of the BackColor is changed.
BorderStyleChanged	It is found in the TextBox Control when the value of the BorderStyle is changed.
ControlAdded	It is found when the new control is added to the Control.ControlCollection.
CursorChanged	It is found in TextBox, when the textbox control is removed from the Control.ControlCollection.
FontChanged	It occurs when the property of the Font is changed.
GetFocus	It is found in TextBox control to get the focus.
MouseClick	A MouseClick event occurs when the mouse clicks the control.
MultilineChanged	It is found in a textbox control when the value of multiline changes.

Furthermore, we can also refer to the VB.NET Microsoft documentation to get a complete list of TextBox properties and events.

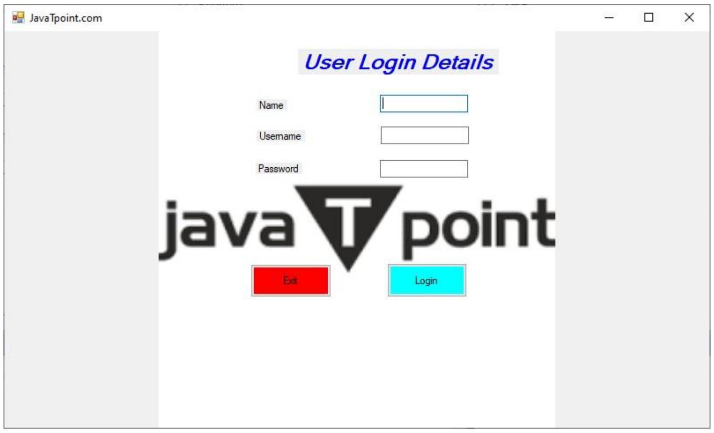
Let us create a program that displays the login details.

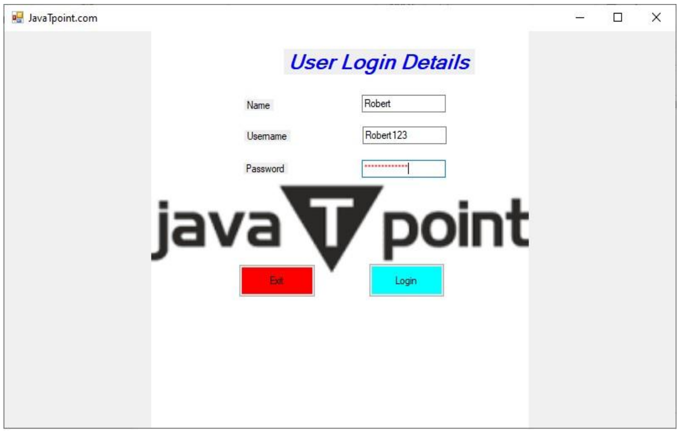
JavatPoint1.vb

```
Public Class JavatPoint1
    Private Sub JavatPoint1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.Text = "JavaTpoint.com" ' title name
        Label1.Text = "User Login Details" ' Set the title name for Label1
        Label2.Text = "Name"            ' Set the name for label2
        Label3.Text = "Username" ' Set the username for label2
        Label4.Text = "Password"    ' Set the label name Passowrd
        Text3.PasswordChar = "*"
        Button1.Text = "Login" ' Set the name of Button1 as Login
        Button2.Text = "Exit"      ' Set the name of Button2 As Exit
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        End ' terminate the program when the user clinks button 2
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim name As String
        Dim Uname As String
        Dim pass As String
        name = text1.Text
        Uname = Text2.Text
        pass = Text3.Text
        ' Display the user details, when the Button1 is clicked
        MsgBox(" Your Name: " & name + vbCrLf + "Your UserName: " & Uname + vbCrLf + "Your Password:
" & pass)
    End Sub
End Class
```

Output:

Now enter all the details of the User Login form, it shows the following image, as shown below.

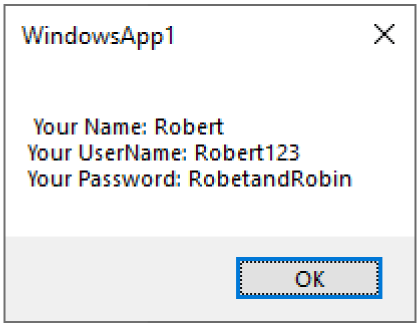




Name: Robert
Username: Robert123

Now, click on the **Login** button. It shows all the details filled by the user in the form.

The Exit button in the form used to terminate the program.



Your Name: Robert
Your UserName: Robert123
Your Password: RobetandRobin

VB.NET Date & Time

In VB.NET, we use the Date and Time function to perform various operations related to date and time. Sometimes we need to display the date and time in our application or web application, such as when the last post edited, upgradation of new software version or patch-up details, etc.

In **DateTime** class, Date datatype stores date values, time values or date, and time values. Furthermore, to perform the date and time function, we need to import the **System.DateTime** class. The default value of DateTime is between 00:00:00 midnight, Jan 1, 0001 to 11:59:59 P.M., Dec 31, 9999 A.D.

Properties and method of DateTime

Date: It is used to return the date component of the DateTime Object.

Day: It is used to return the day of the month represented by the DateTime object.

DayOfWeek: It is used to return a particular day of the week represented by the DateTime object.

Minute: The Minute property is used to return the minute component by the DateTime object.

DateOfYear: It is used to return a day of the year represented by the DateTime object.

Hour: It is used to return the hour of the component of the date represented by the DateTime object.

Now: It is used to return the current date and time of the local system.

Month: The Month property is used to return the month name of the Datetime object.

Second: It is used to return the second of the DateTime object.

Ticks: It is used to return the number of ticks that refer to the DateTime object.

Today: It is used to return the current date of the system.

Year: It is used to return the year of the date represented by the DateTime object.

TimeOfDay: It is used to return the time of the day represented by the DateTime object.

Methods

The following are the most commonly used methods of the DateTime.

DaysInMonth: The DaysInMonth method is used to return the total number of days in the specified month of the year.

Add: It is used to return a new DateTime value by adding the timespan value to the DateTime object value.

AddHours: It is used to return a new time by adding the hours to the value of the Datetime object.

AddYears: It is used to return the year by adding the year to the value of the DateTime object.

AddDays: It is used to return the new Day by adding the days to the value of the DateTime object.

AddMinutes: It is used to display the new time by adding the minutes to the Datetime object.

AddMonths: It is used to return the new time by adding the months to the value of the Datetime object.

AddSeconds: It is used to return the new time by adding the seconds to the value of the Datetime object.

IsLeapYear: It uses a Boolean value that represents whether the particular year is a leap year or not.

Syntax: Let us create an object of the DateTime.

```
Dim obj_name As DateTime = New DateTime()
```

Here, **DateTime** is a class for creating objects with the new keyword, and **obj_name** is the name of the object.

Let us create a program to show the different functions of DateTime Class in VB.NET.

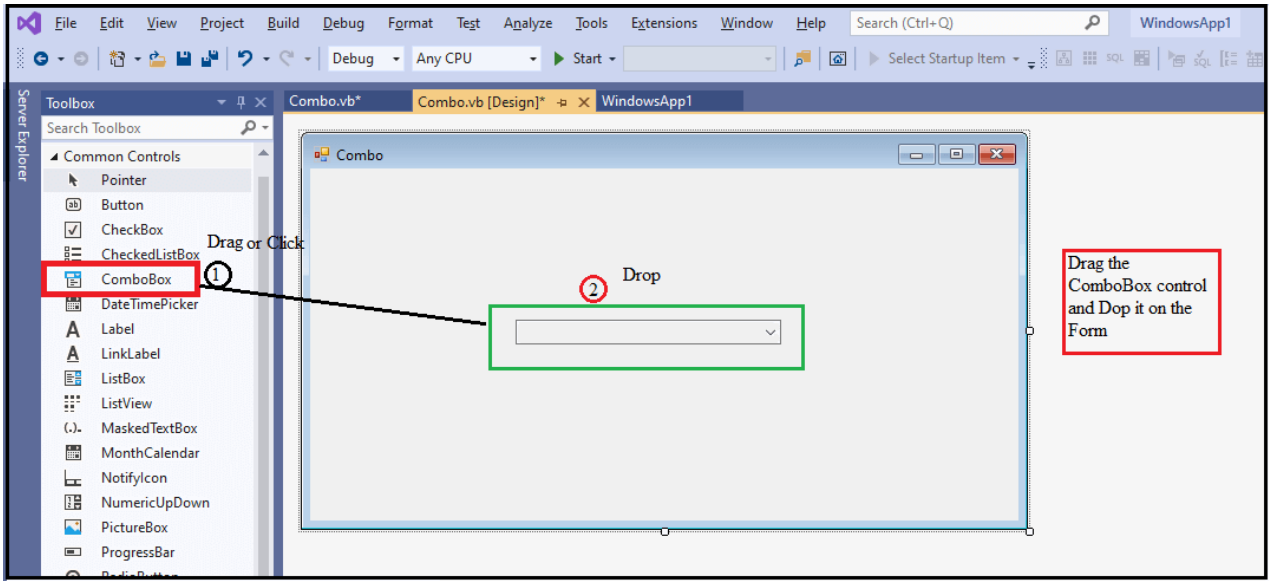
DiffDateTime.vb

VB.NET ComboBox Control

The **ComboBox** control is used to display more than one item in a drop-down list. It is a combination of **Listbox** and **Textbox** in which the user can input only one item. Furthermore, it also allows a user to select an item from a drop-down list.

Let us create a ComboBox control in the VB.NET Windows by using the following steps.

Step 1: We need to drag the combo box control from the toolbox and drop it to the Windows form, as shown below.

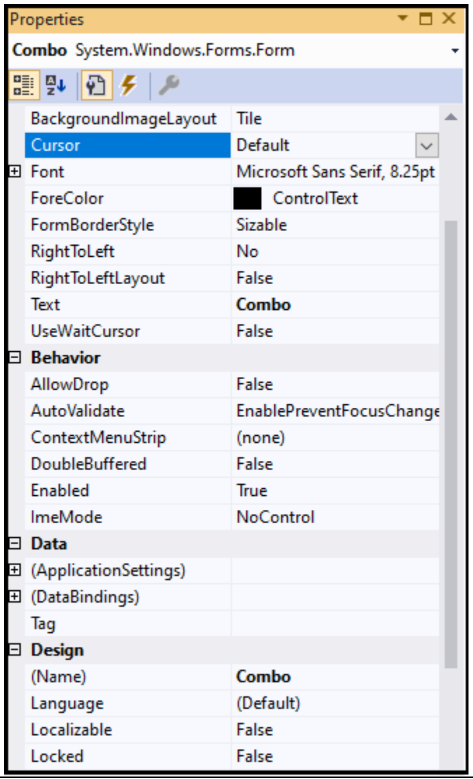


Drag or Click

Drop

Drag the ComboBox control and Dop it on the Form

Step 2: Once the ComboBox is added to the form, we can set various properties of the ComboBox by clicking on the ComboBox control.



Properties

Combo System.Windows.Forms.Form

BackgroundImageLayout	Tile
Cursor	Default
⊞ Font	Microsoft Sans Serif, 8.25pt
ForeColor	■ ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	**Combo**
UseWaitCursor	False
⊟ **Behavior**	
AllowDrop	False
AutoValidate	EnablePreventFocusChange
ContextMenuStrip	(none)
DoubleBuffered	False
Enabled	True
ImeMode	NoControl
⊟ **Data**	
⊞ (ApplicationSettings)	
⊞ (DataBindings)	
Tag	
⊟ **Design**	
(Name)	**Combo**
Language	(Default)
Localizable	False
Locked	False

ComboBox Properties

There are following properties of the ComboBox control.

Property	**Description**
AllowSelection	The AllowSelection property takes the value that indicates whether the list allows selecting the list item.
AutoCompleteMode	It takes a value that represents how automatic completion work for the ComboBox.
Created	It takes a value that determines whether the control is created or not.
DataBinding	It is used to bind the data with a ComboBox Control.
BackColor	The BackColor property is used to set the background color of the combo box control.
DataSource	It is used to get or set the data source for a ComboBox Control.
FlatStyle	It is used to set the style or appearance for the ComboBox Control.
MaxDropDownItems	The MaxDropDownItems property is used in the combo box control to display the maximum number of items by setting a value.
MaxLength	It is used by the user to enter maximum characters in the editable area of the combo box.
SelectedItem	It is used to set or get the selected item in the ComboBox Control.
Sorted	The Sorted property is used to sort all the items in the ComboBox by setting the value.

ComboBox Events

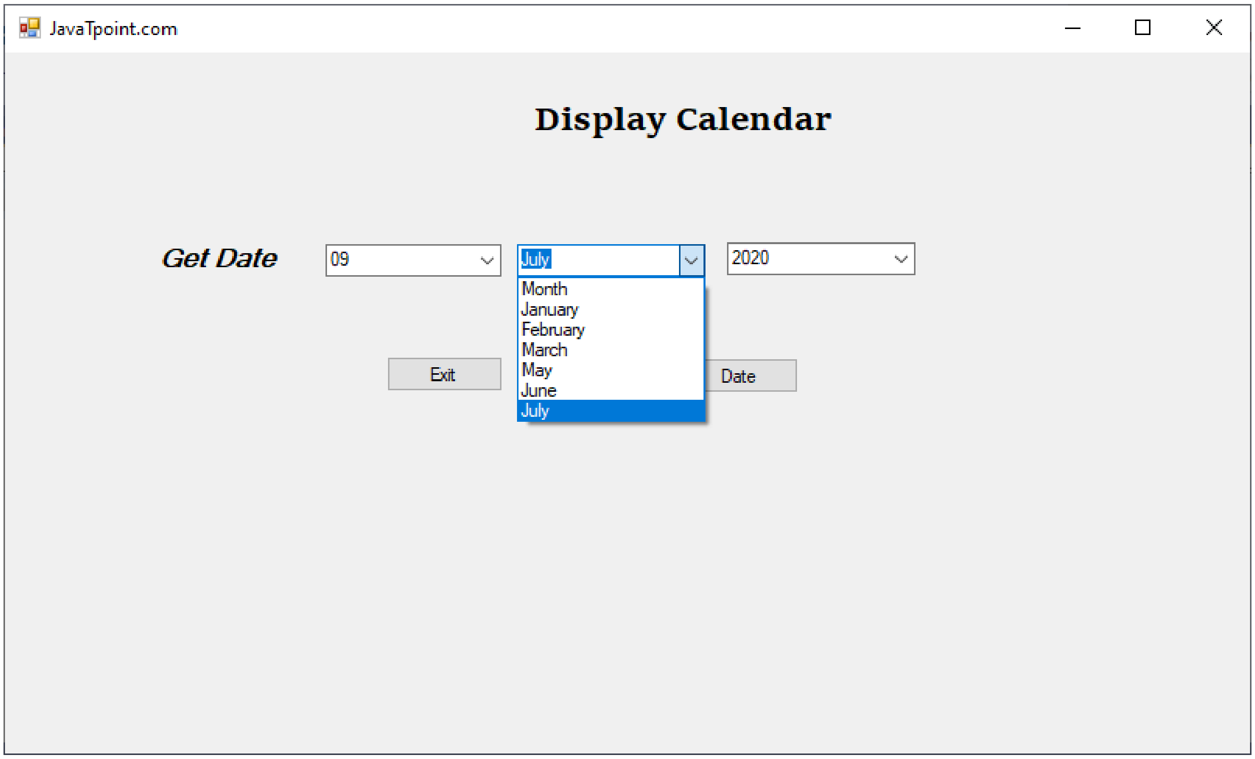
Events	Description
FontChanged	It occurs when the property of the font value is changed.
Format	When the data is bound with a combo box control, a format event is called.
SelectIndexChanged	It occurs when the property value of SelectIndexChanged is changed.
HelpRequested	When the user requests for help in control, the HelpRequested event is called.
Leave	It occurs when the user leaves the focus on the ComboBox Control.
MarginChanged	It occurs when the property of margin is changed in the ComboBox control.

Let us create a program to display the Calendar in the VB.NET Windows Form.

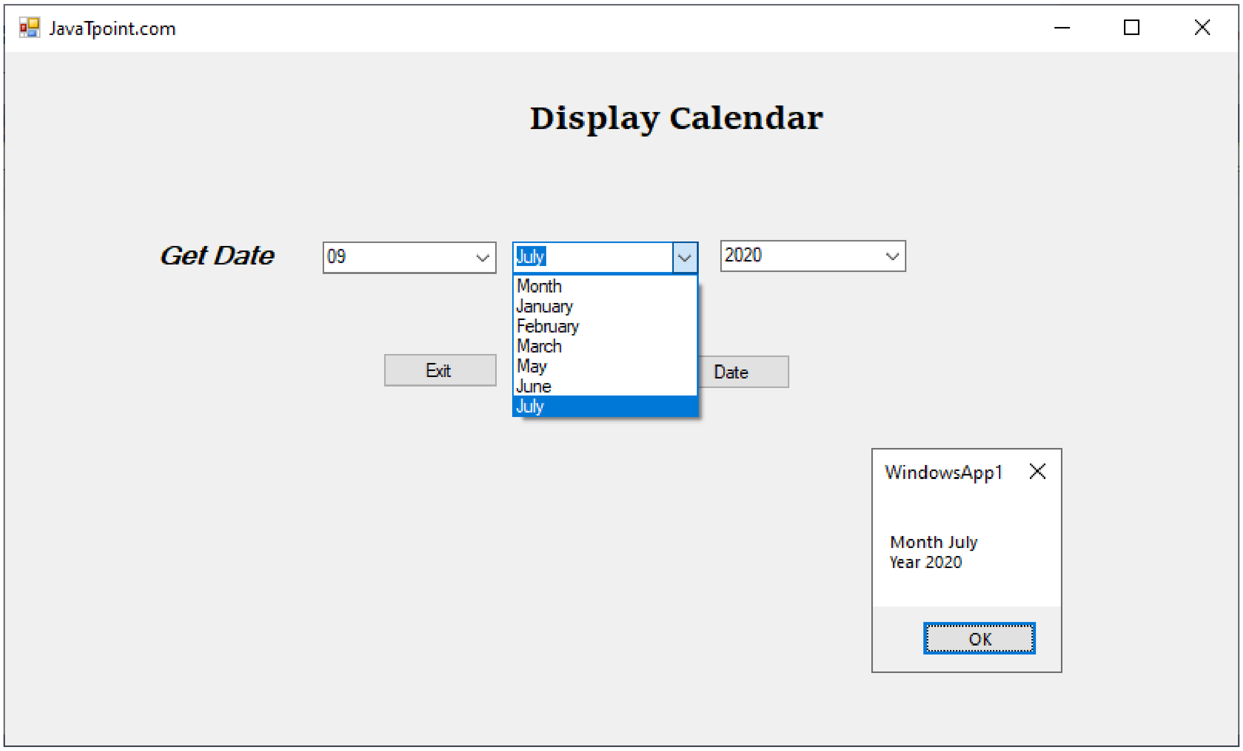
ComboBox_Control.vb

```
Public Class ComboBox_Control
    Dim DT As Integer
    Dim MM As String
    Dim YY As Integer
    Private Sub ComboBox_Control_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.Text = "JavaTpoint.com"
        Label1.Text = "Display Calendar"
        Label2.Text = "Get Date"
        Button1.Text = "Date"
        Button2.Text = "Exit"
        ComboBox1.Items.Add("Date")
        ComboBox1.Items.Add("01")
        ComboBox1.Items.Add("02")
        ComboBox1.Items.Add("03")
        ComboBox1.Items.Add("04")
        ComboBox1.Items.Add("05")
        ComboBox1.Items.Add("06")
        ComboBox1.Items.Add("07")
        ComboBox1.Items.Add("08")
        ComboBox1.Items.Add("09")
        ComboBox2.Items.Add("Month")
        ComboBox2.Items.Add("January")
        ComboBox2.Items.Add("February")
        ComboBox2.Items.Add("March")
        ComboBox2.Items.Add("May")
        ComboBox2.Items.Add("June")
        ComboBox2.Items.Add("July")
        ComboBox3.Items.Add("Year")
        ComboBox3.Items.Add("2016")
        ComboBox3.Items.Add("2017")
        ComboBox3.Items.Add("2018")
        ComboBox3.Items.Add("2019")
        ComboBox3.Items.Add("2020")
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        DT = ComboBox1.Text
        MM = ComboBox2.Text
        YY = ComboBox3.Text
        MsgBox("Month " & MM + vbCrLf + "Year " & YY)
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        End
    End Sub
End Class
```

Output:



Now select the day, month, and year from dropdown box and then click on the Date button to display the date in the form.

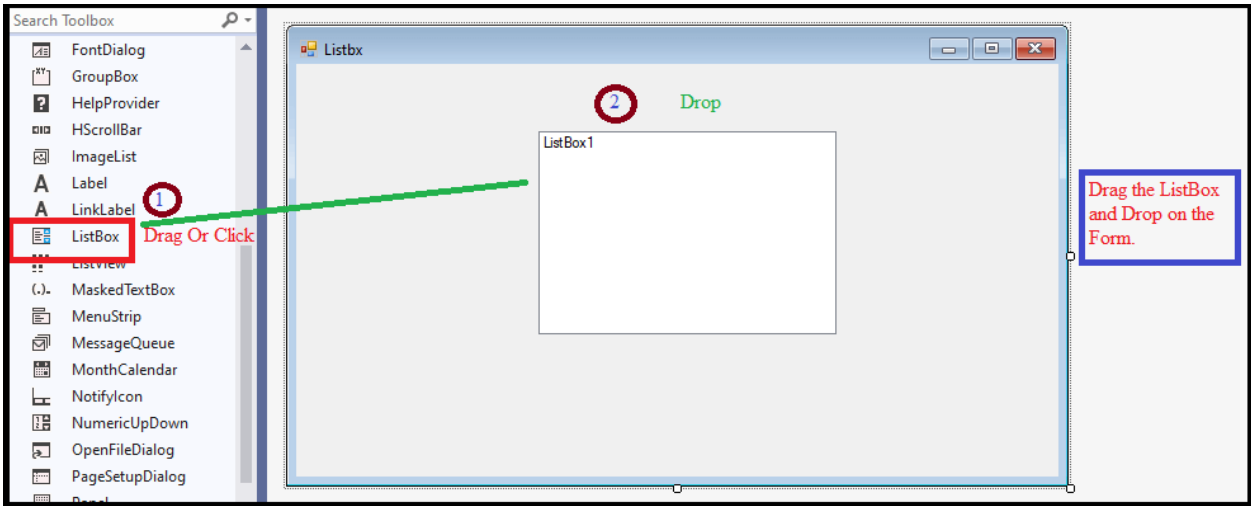


Month July
Year 2020

VB.NET ListBox Control

The ListBox control is used to display a list of items in Windows form. It allows the user to select one or more items from the ListBox Control. Furthermore, we can add or design the list box by using the properties and events window at runtime.

Let us create a ListBox control in the VB.NET Windows by using the following steps.



Step 1: Drag the ListBox control from the Toolbox and drop it to the Windows form, as shown below.

Step 2: Once the ListBox is added to the Form, we can set various properties of the Listbox by clicking on the ListBox control.

ListBox Properties

There are following properties of the ListBox control.

Properties Name	Description
AllowSelection	It takes a value that defines whether the list box allows the user to select the item from the list.
CanSelect	It obtains a value that determines whether the Listbox control can be selected.
ColumnWidth	It is used to get or set the width of the columns in a multicolumn Listbox.
Container	As the name defines, a container gets the IContainer that stores the component of ListBox control.
Controls	It is used to get the collection of controls contained within the control.

Created	It takes a value that determines whether the control is created or not.
Width	It is used to set the width of the ListBox control.
Visible	It takes a value that determines whether the ListBox control and all its child are displayed on the Windows Form.
SelectionMode	It is used to get or set the method that determines which items are selected in the ListBox.
MultiColumn	It allows multiple columns of the item to be displayed by setting the True value in the Listbox.

ListBox Methods

Method Name	**Description**
Add()	The Add() method is used to add items to an item collection.
Remove	It is used to remove an item from an item collection. However, we can remove items using the item name.
Clear	It is used to remove all items from the item collection at the same time.
Contains	It is used to check whether the particular item exists in the ListBox or not.
Show()	It is used to display the control to the user.
Sort()	As the name suggests, a Sort() method is used to arrange or sort the elements in the ListBox.
ResetText()	A ResetText() method is used to reset ListBox's text property and set the default value.
ResetBackColor()	It is used to reset the backColor property of the ListBox and set the default value.
OnNotifyMessage	It is used to notify the message of the ListBox to Windows.
GetSelected	The GetSelected method is used to validate whether the specified item is selected.

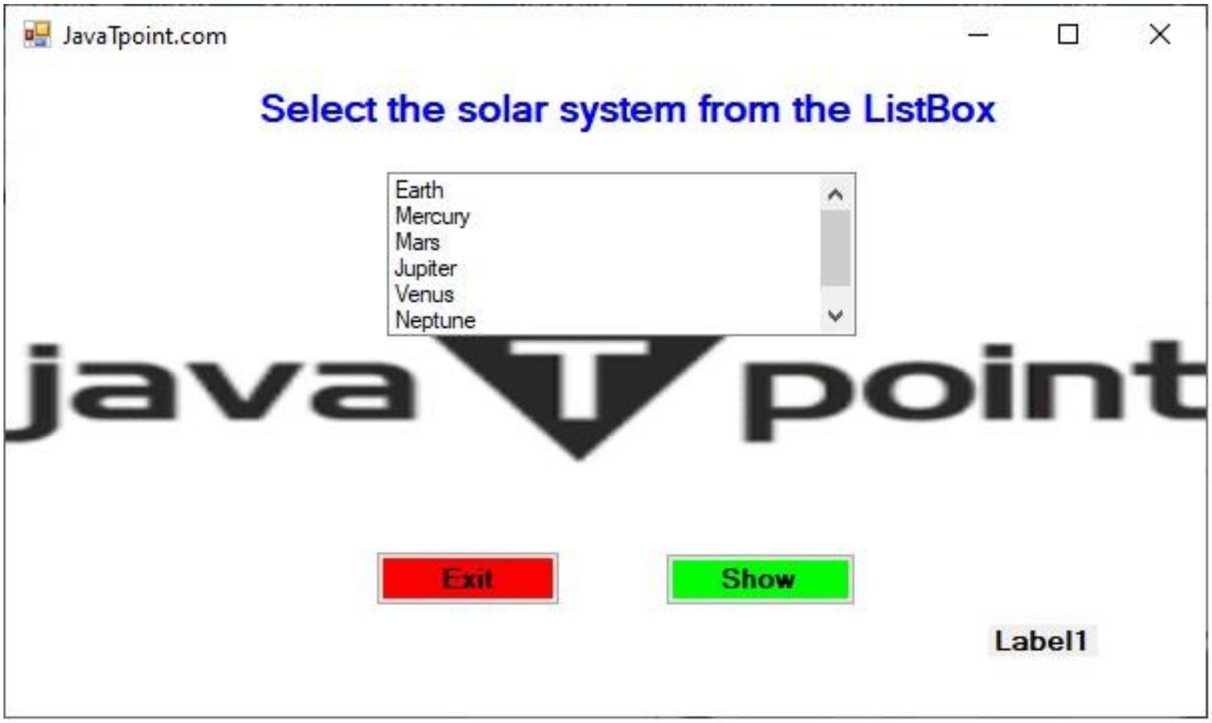
Furthermore, we can also refer to VB.NET Microsoft documentation to get a complete list of ListBox properties, and methods.

Let's create a program to select an item from the ListBox in the VB.NET form.

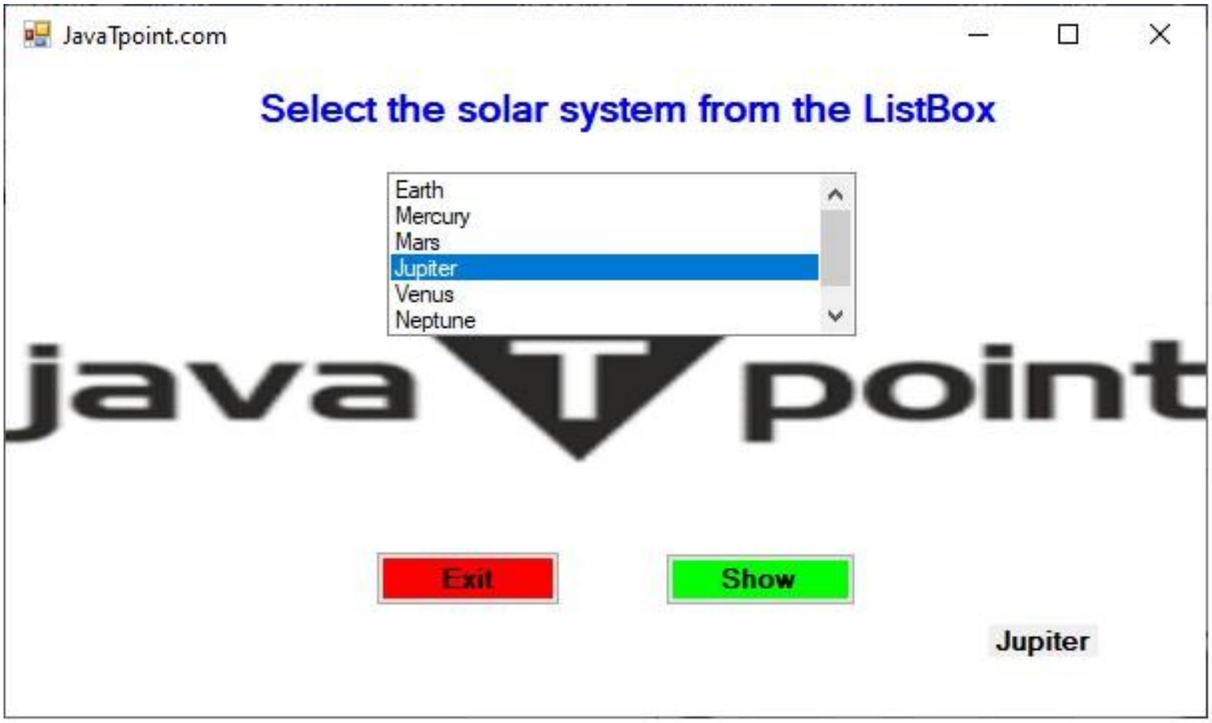
Listbx.vb

```
Public Class Listbx
    Private Sub Listbx_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        ' set the title of the Form.
        Me.Text = "JavaTpoint.com"
        ' Add the items into the ListBox
        ListBox1.Items.Add("Earth")
        ListBox1.Items.Add("Mercury")
        ListBox1.Items.Add("Mars")
        ListBox1.Items.Add("Jupiter")
        ListBox1.Items.Add("Venus")
        ListBox1.Items.Add("Neptune")
        ListBox1.Items.Add("Uranus")
        ' Set the name of the Button1 and Button2
        Button1.Text = "Show"
        Button2.Text = "Exit"
        Label2.Text = "Select the solar system from the ListBox"
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim lt As String  ' define a local variable.
        lt = ListBox1.Text  'accept the data from the ListBox1
        MsgBox(" Selected Solar System is " & lt)  ' Display the selected item
    End Sub
    Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ListBox1.Selected
IndexChanged
        Label1.Text = ListBox1.SelectedItem.ToString() 'When a user clicks on an item, it displays the item name
.
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        End 'End or exit an application
    End Sub
End Class
```

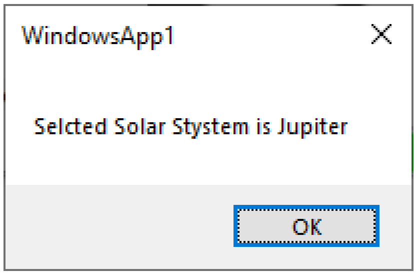
Output:



Now select an item from the list. We have selected Jupiter.



Click on the **Show** button to display the selected item in Windows Form, as follows.



WindowsApp1

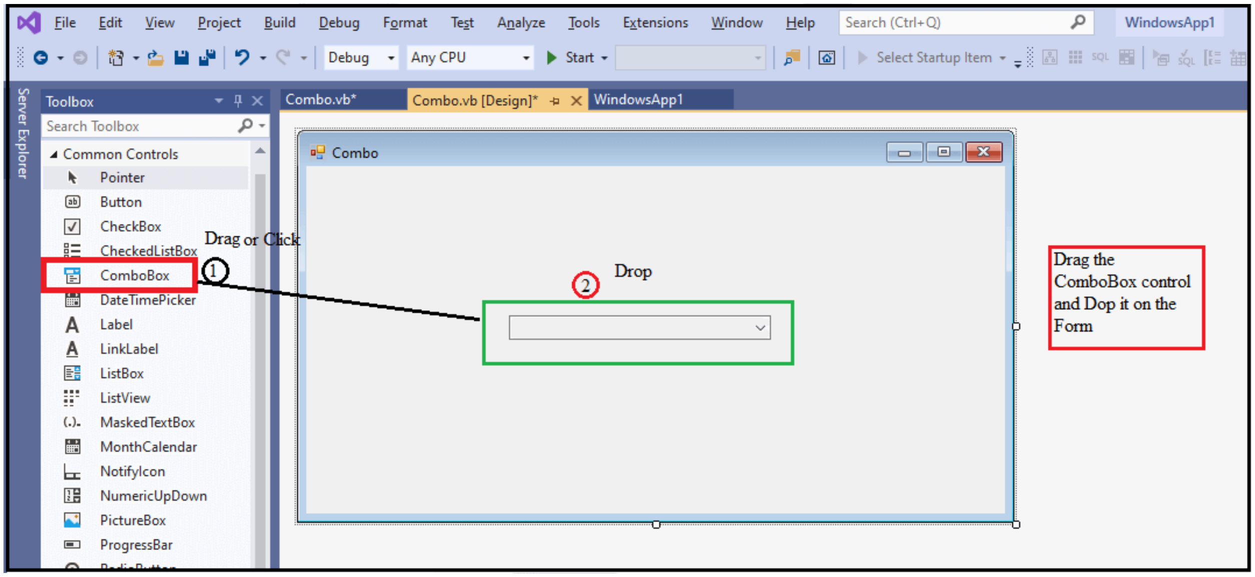
Selcted Solar Stystem is Jupiter

VB.NET ComboBox Control

The **ComboBox** control is used to display more than one item in a drop-down list. It is a combination of **Listbox** and **Textbox** in which the user can input only one item. Furthermore, it also allows a user to select an item from a drop-down list.

Let us create a ComboBox control in the VB.NET Windows by using the following steps.

Step 1: We need to drag the combo box control from the toolbox and drop it to the Windows form, as shown below.



Drag or Click

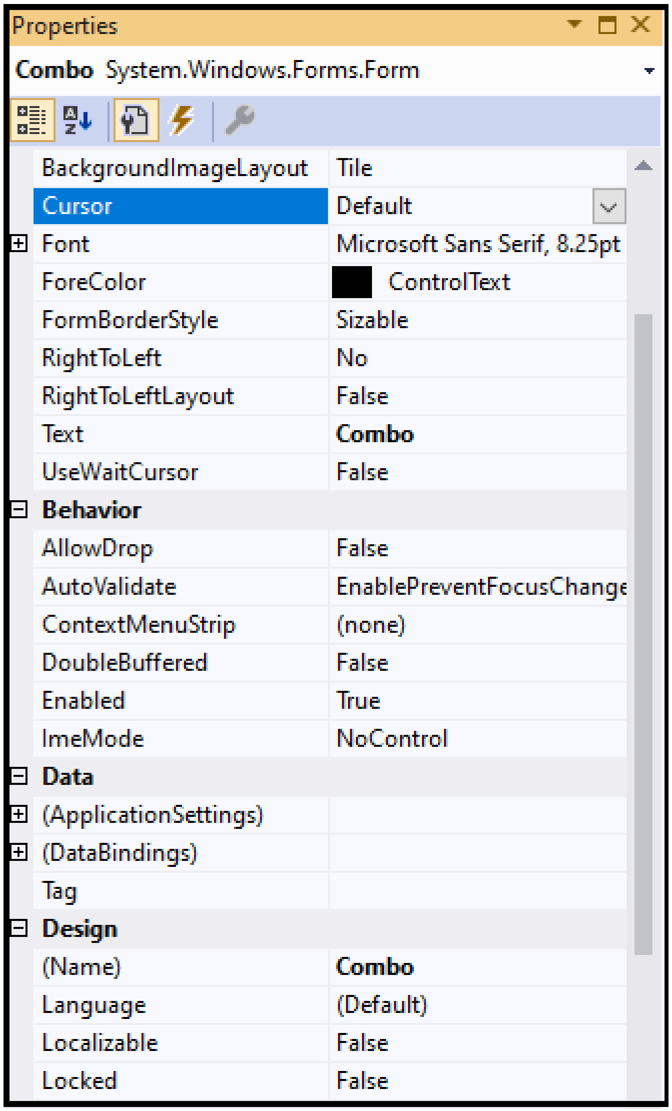
Drop

Drag the ComboBox control and Dop it on the Form

Step 2: Once the ComboBox is added to the form, we can set various properties of the ComboBox by clicking on the ComboBox control.

ComboBox Properties

There are following properties of the ComboBox control.



Properties

Combo System.Windows.Forms.Form

BackgroundImageLayout	Tile
Cursor	Default
Font	Microsoft Sans Serif, 8.25pt
ForeColor	■ ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	**Combo**
UseWaitCursor	False
Behavior	
AllowDrop	False
AutoValidate	EnablePreventFocusChange
ContextMenuStrip	(none)
DoubleBuffered	False
Enabled	True
ImeMode	NoControl
Data	
(ApplicationSettings)	
(DataBindings)	
Tag	
Design	
(Name)	**Combo**
Language	(Default)
Localizable	False
Locked	False

Property	**Description**
AllowSelection	The AllowSelection property takes the value that indicates whether the list allows selecting the list item.
AutoCompleteMode	It takes a value that represents how automatic completion work for the ComboBox.
Created	It takes a value that determines whether the control is created or not.
DataBinding	It is used to bind the data with a ComboBox Control.
BackColor	The BackColor property is used to set the background color of the combo box

	control.
DataSource	It is used to get or set the data source for a ComboBox Control.
FlatStyle	It is used to set the style or appearance for the ComboBox Control.
MaxDropDownItems	The MaxDropDownItems property is used in the combo box control to display the maximum number of items by setting a value.
MaxLength	It is used by the user to enter maximum characters in the editable area of the combo box.
SelectedItem	It is used to set or get the selected item in the ComboBox Control.
Sorted	The Sorted property is used to sort all the items in the ComboBox by setting the value.

ComboBox Events

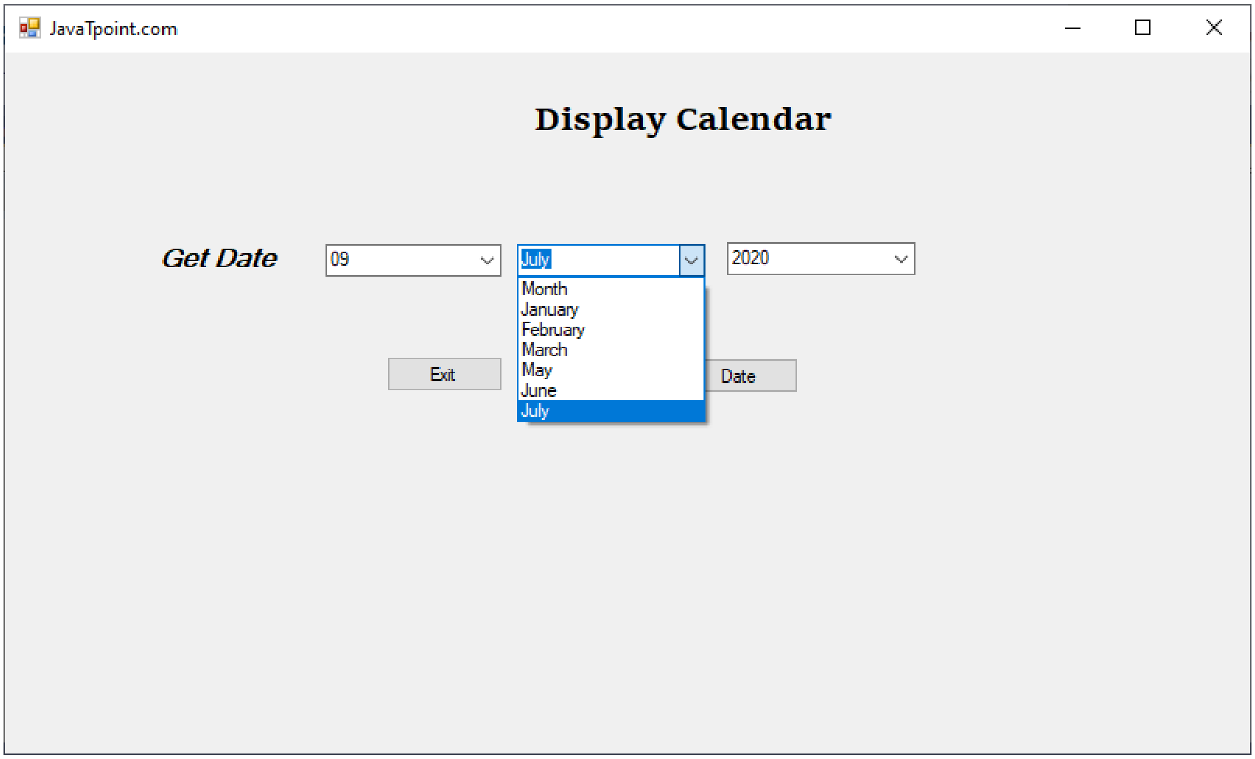
Events	**Description**
FontChanged	It occurs when the property of the font value is changed.
Format	When the data is bound with a combo box control, a format event is called.
SelectIndexChanged	It occurs when the property value of SelectIndexChanged is changed.
HelpRequested	When the user requests for help in control, the HelpRequested event is called.
Leave	It occurs when the user leaves the focus on the ComboBox Control.
MarginChanged	It occurs when the property of margin is changed in the ComboBox control.

Let us create a program to display the Calendar in the VB.NET Windows Form.

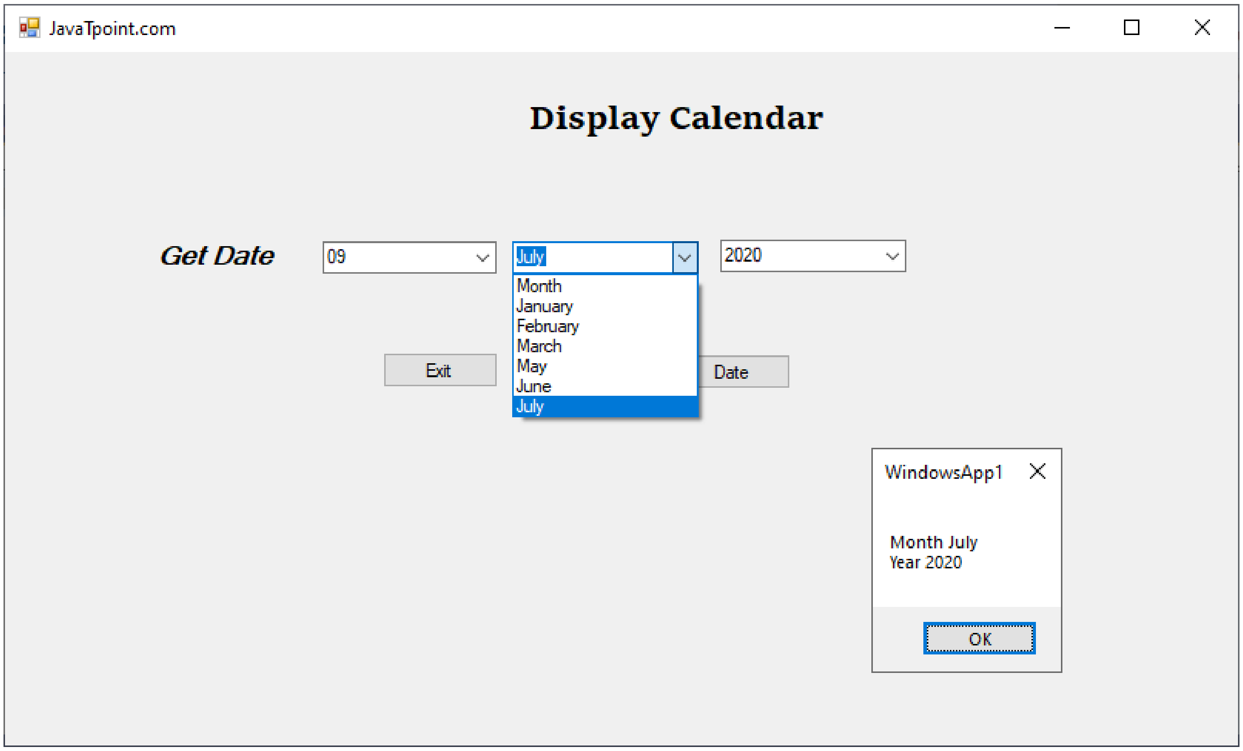
ComboBox_Control.vb

```
Public Class ComboBox_Control
    Dim DT As Integer
    Dim MM As String
    Dim YY As Integer
    Private Sub ComboBox_Control_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.Text = "JavaTpoint.com"
        Label1.Text = "Display Calendar"
        Label2.Text = "Get Date"
        Button1.Text = "Date"
        Button2.Text = "Exit"
        ComboBox1.Items.Add("Date")
        ComboBox1.Items.Add("01")
        ComboBox1.Items.Add("02")
        ComboBox1.Items.Add("03")
        ComboBox1.Items.Add("04")
        ComboBox1.Items.Add("05")
        ComboBox1.Items.Add("06")
        ComboBox1.Items.Add("07")
        ComboBox1.Items.Add("08")
        ComboBox1.Items.Add("09")
        ComboBox2.Items.Add("Month")
        ComboBox2.Items.Add("January")
        ComboBox2.Items.Add("February")
        ComboBox2.Items.Add("March")
        ComboBox2.Items.Add("May")
        ComboBox2.Items.Add("June")
        ComboBox2.Items.Add("July")
        ComboBox3.Items.Add("Year")
        ComboBox3.Items.Add("2016")
        ComboBox3.Items.Add("2017")
        ComboBox3.Items.Add("2018")
        ComboBox3.Items.Add("2019")
        ComboBox3.Items.Add("2020")
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        DT = ComboBox1.Text
        MM = ComboBox2.Text
        YY = ComboBox3.Text
        MsgBox("Month " & MM + vbCrLf + "Year " & YY)
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        End
    End Sub
End Class
```

Output:



Now select the day, month, and year from dropdown box and then click on the Date button to display the date in the form.



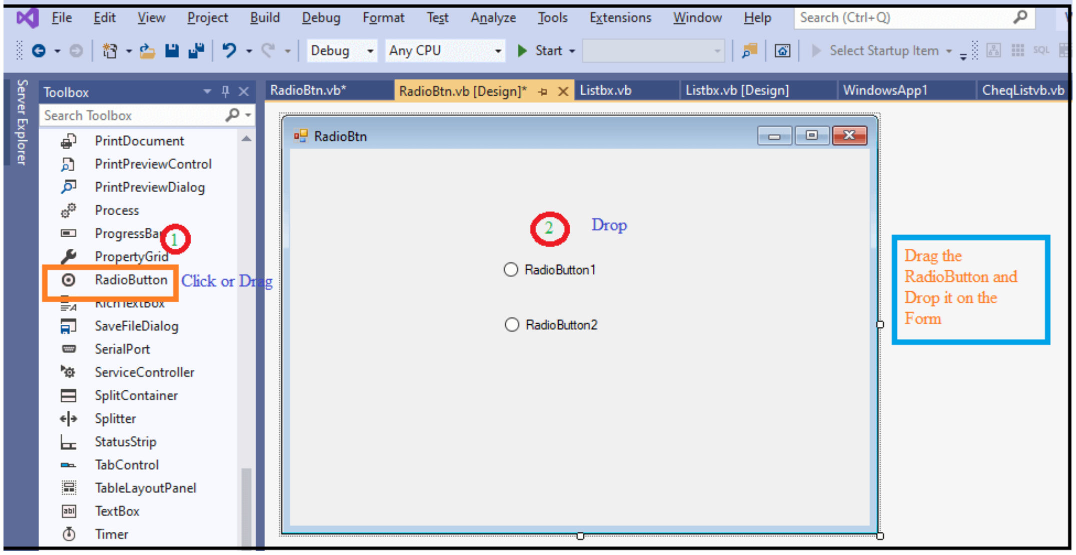
Month July
Year 2020

RadioButton Control

The **RadioButton** is used to select one option from the number of choices. If we want to select only one item from a related or group of items in the windows forms, we can use the radio button. The **RadioButton** is mutually exclusive that represents only one item is active and the remains unchecked in the form.

Let us create a **RadioButton** control in the VB.NET Windows by using the following steps.

Step 1: Drag the **RadioButton** control from the toolbox and drop it to the Windows form, as shown below.



1 Click or Drag

2 Drop

Drag the RadioButton and Drop it on the Form

Step 2: Once the RadioButton is added to the form, we can set various properties of the RadioButton by clicking on the Radio control.

RadioButton Properties

There are following properties of the VB.NET RadioButton control.

Property	Description
AllowDrop	It is used to set or get a value representing whether the RadioButton allows the user to drag on the form.
Appearance	It is used to get or set a value that represents the appearance of the RadioButton.
AutoScrollOffset	It is used to get or set the radio control in ScrollControlIntoView(Control).
AutoCheck	The AutoCheck property is used to check whether the checked value or appearance of control can be automatically changed when the user clicked on the RadioButton control.
AutoSize	The AutoSize property is used to check whether the radio control can be automatically resized by setting a value in the RadioButton control.
CanSelect	A CanSelect property is used to validate whether a radio control can be selected by setting a value in the RadioButton control.
CheckAlign	It is used to obtain or set a value that indicates the location of the check portion in the radioButton control.
Text	The Text property is used to set the name of the RadioButton control.

RadioButton Methods

Method Name	Description
Contains(Control)	The Contains() method is used to check if the defined control is available in the RadioButton control.
DefWndProc(Message)	It is used to send the specified message to the Window procedure.
DestroHandle()	It is used to destroy the handle associated with the RadioButton Control.
Focus()	The Focus() method is used to set the input focus to the window form's RadioButton control.
GetAutoSizeMode()	It is used to return a value that represents how the control will operate when the AutoSize property is enabled in the RadioButton control of the Window form.

ResetText()	As the name suggests, a ResetText() method is used to reset the property of text to its default value or empty.
Update()	It is used to reroute an invalid field, which causes control in the client region.

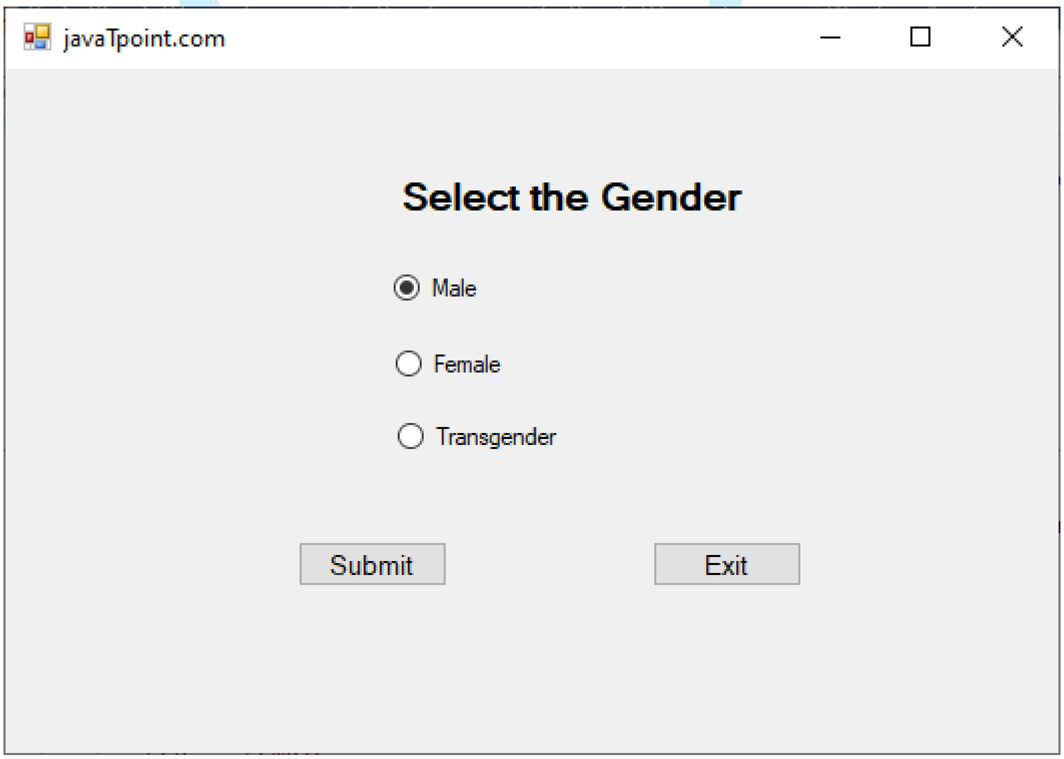
We can also refer to Microsoft documentation to get a complete list of RadioButton Control properties and methods in the VB .NET.

Let us create a program to understand the uses of Radio button control in the VB.NET form.

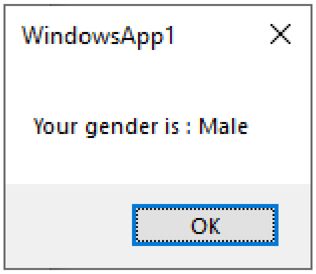
RadioBtn.vb

```
Public Class RadioBtn
  Private Sub RadioBtn_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.Text = "javaTpoint.com" ' Set the title of the form
        Label1.Text = "Select the Gender"
        RadioButton1.Text = "Male" ' Set the radiobutton1 and radiobutton2
        RadioButton2.Text = "Female"
        RadioButton3.Text = "Transgender"
        Button1.Text = "Submit"  ' Set the button name
        Button2.Text = "Exit"
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim gen As String
        If RadioButton1.Checked = True Then
            gen = "Male"
            MsgBox(" Your gender is : " & gen)

        ElseIf RadioButton2.Checked = True Then
            gen = "Female"
            MsgBox(" Your gender is : " & gen)
        Else
            gen = "Transgender"
            MsgBox(" You have Selected the gender : " & gen)
        End If

    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        End 'Terminate the program
    End Sub
End Class
```

Output:



Click on the **Submit** button. It shows the following message on the screen.



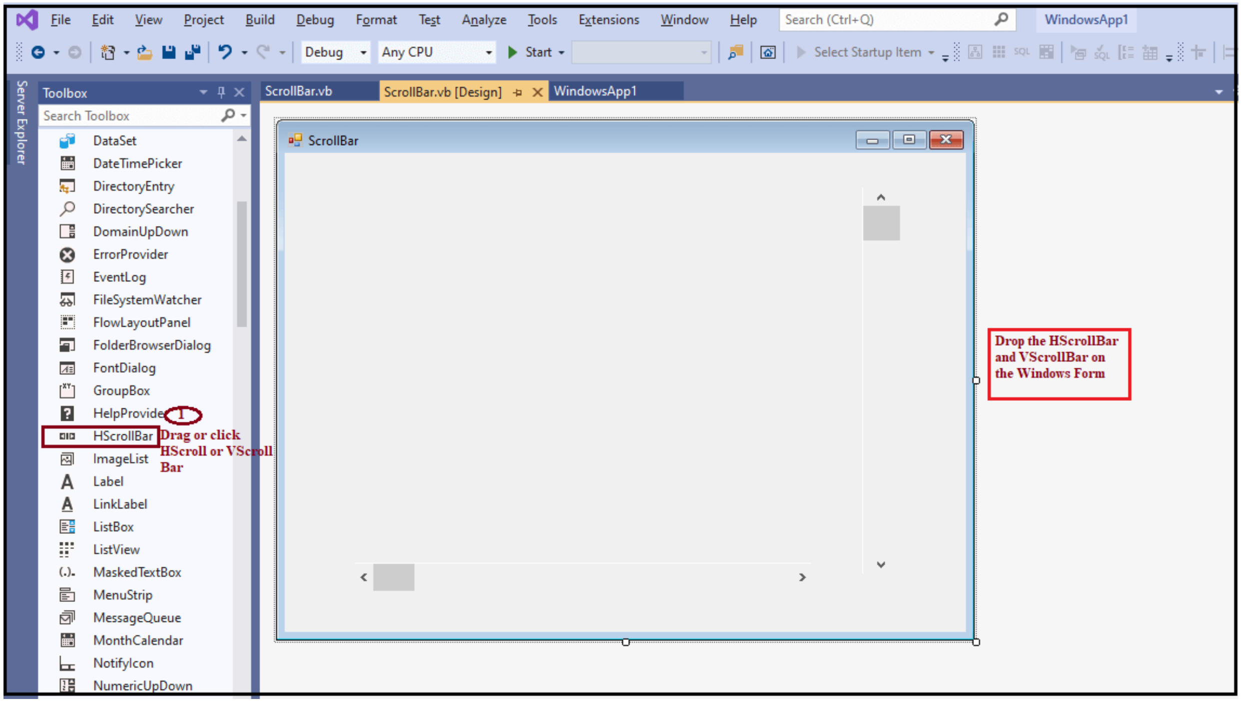
Your gender is : Male

VB.NET ScrollBars Control

A ScrollBar control is used to create and display vertical and horizontal scroll bars on the Windows form. It is used when we have large information in a form, and we are unable to see all the data. Therefore, we used VB.NET ScrollBar control. Generally, ScrollBar is of two types: HScrollBar for displaying scroll bars and VScrollBar for displaying Vertical Scroll bars.

Let's create a ScrollBar control in the VB.NET Windows form using the following steps.

Step 1: The first step is to drag the HScrollBar and VScrollBar control from the toolbox and drop it on to the form.



1 Drag or click HScroll or VScroll Bar

Drop the HScrollBar and VScrollBar on the Windows Form

Step 2: Once the ScrollBar is added to the form, we can set various properties of the ScrollBar by clicking on the HScrollBar and VScrollBar control.

Properties of the ScrollBar Control

There are following properties of the VB.NET ScrollBar control.

Property	**Description**
BackColor	The BackColor property is used to set the back color of the scroll bar.
Maximum	It is used to set or get the maximum value of the Scroll Bar control. By default, it is 100.
Minimum	It is used to get or set the minimum value of the Scroll bar control. By default, it is 0.
SmallChange	It is used to obtain or set a value that will be added or subtracted from the property of the scroll bar control when the scroll bar is moved a short distance.
AutoSize	As the name suggests, the AutoSize property is used to get or set a value representing whether the scroll bar can be resized automatically or not with its contents.
LargeChange	It is used to obtain or set a value that will be added or subtracted from the property of the scroll bar control when the scroll bar is moved a large distance.
Value	It is used to obtain or set a value in a scroll bar control that indicates a scroll box's current position.
DefaultImeMode	It is used to get the default input method Editor (IME) that are supported by ScrollBar controls in the Windows Form.

Methods of the ScrollBar Control

Method	**Description**
UpdateScrollInfo	It is used to update the ScrollBar control using the Minimum, maximum, and the value of LargeChange properties.
OnScroll(ScrollEventArgs)	It is used to raise the Scroll event in the ScrollBar Control.
OnEnabledChanged	It is used to raise the EnabledChanged event in the ScrollBar control.
Select	It is used to activate or start the ScrollBar control.
OnValueChanged(EventArgs)	It is used to raise the ValueChanged event in the ScrollBar control.

Events of the ScrollBar Control

Event	Description
AutoSizeChanged	The AutoSizeChanged event is found in the ScrollBar control when the value of the AutoSize property changes.
Scroll	The Scroll event is found when the Scroll control is moved.
TextChangedEvent	It occurs in the ScrollBar control when the value of the text property changes.
ValueChanged	A ValueChanged event occurs when the property of the value is changed programmatically or by a scroll event in the Scrollbar Control.

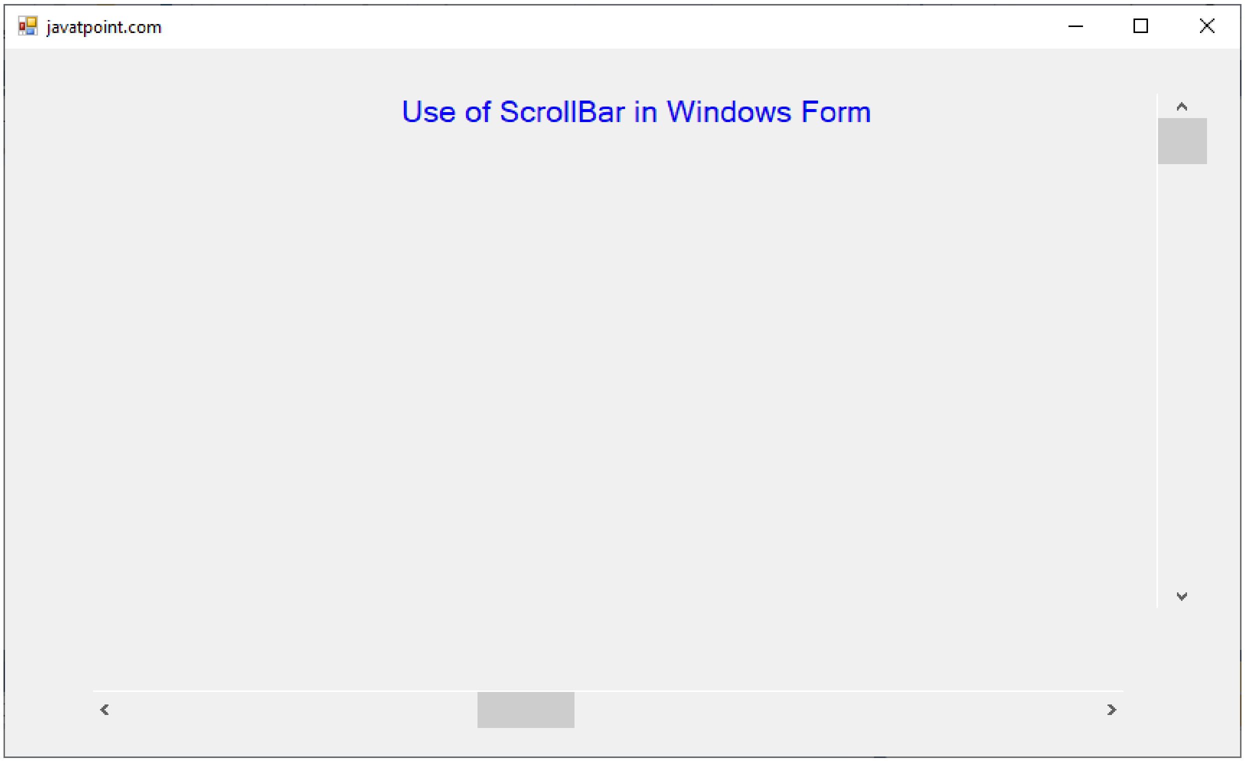
Furthermore, we can also refer to VB.NET Microsoft documentation to get a complete list of **ScrollBar** control properties, methods, and events in the VB.NET.

Let us create a simple program to understand the use of **ScrollBar** Control in the VB.NET Windows Forms.

ScrollBar.vb

```
Public Class ScrollBar
    Private Sub ScrollBar_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.Text = "javatpoint.com"  'Set the title for a Windows Form
        Label1.Text = "Use of ScrollBar in Windows Form"
        Label1.ForeColor = Color.Blue
        Me.AutoScroll = True
        Me.VScrollBar1.Minimum = 0
        Me.VScrollBar1.Maximum = 100

        Me.VScrollBar1.BackColor = Color.Blue
        Me.HScrollBar1.Minimum = 0
        Me.HScrollBar1.Maximum = 100
        Me.HScrollBar1.Value = 35
    End Sub
End Class
```

Output:



Subroutines and functions in vb.net

.NET is a software framework that is designed and developed by Microsoft. The first version of the .Net framework was 1.0 which came in the year 2002. In other words, it is a virtual machine for compiling and executing programs written in different languages like C#, VB.Net, etc.

Sub Procedures or subroutines:

A subprocedure is a group of VB.NET statements. It begins with a Sub keyword and ends with End Sub keywords. A subprocedure is also called a subroutine. It is used to execute a certain block of statements consists the body of the procedure. It is called explicitly by its name whenever it is required to perform a certain task. It can be called any number of times. The subprocedure returns control to the calling code after performing a task.

Structure of Subprocedure:

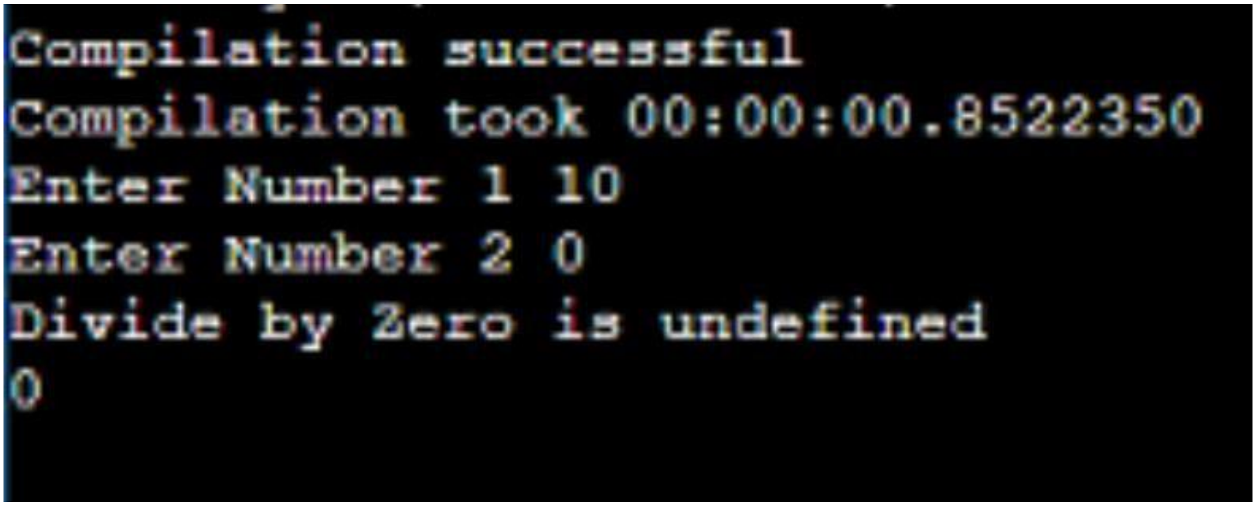
```
Sub <subname> [(parameter list)]
Vb statements
End Sub
```

Example:

```
Module module1
 Sub SubDivide(ByVal num1 As Integer,
  ByVal num2 As Integer)

    Dim res As Integer

    If (num2 <> 0) Then
    res = num1/num2
       Console.WriteLine("Divide by Zero is possible")
    Else
      Console.WriteLine("Divide by Zero is undefined")
    End If
 End Sub
 Sub Main()
    Dim a As Integer
    Dim b As Integer
    Dim res As Integer
    Console.Write("Enter Number 1")
    a = Console.ReadLine()
```

```
   Console.Write("Enter Number 2")

   b = Console.ReadLine()

   SubDivide(a, b)


    Console.WriteLine(res)


 End Sub

End Module
```

Output:



```
Compilation successful
Compilation took 00:00:00.8522350
Enter Number 1 10
Enter Number 2 0
Divide by Zero is undefined
0
```

Function Procedures:

A function procedure is a group of VB.NET statements. It begins with a Function keyword and ends with an End Function keyword. It is generally used to perform a task and return a value back to the calling code. It may have multiple return points to the calling code. A part from return stamens, End Function, or Exit function also returns control to the calling procedure.

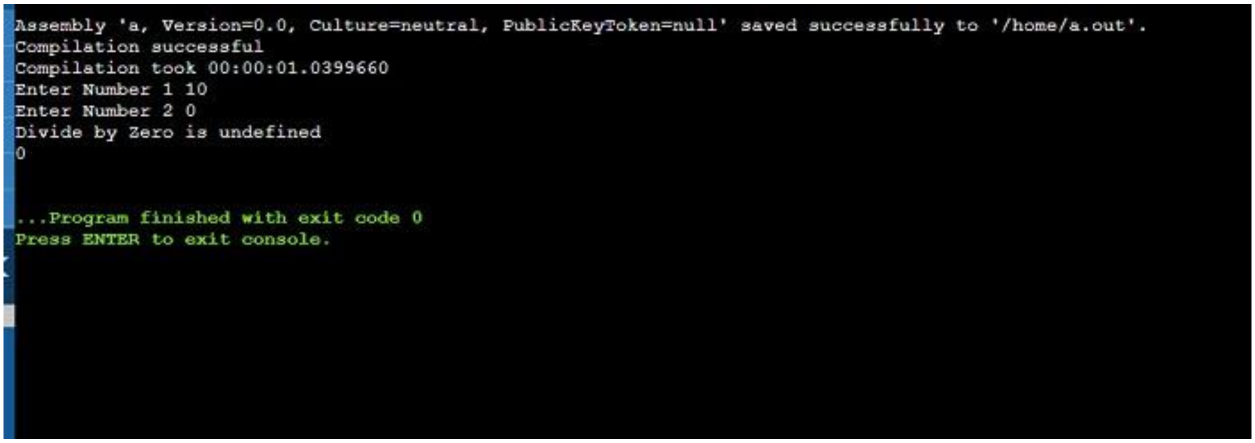
Structure of Function Procedure:

```
Function <Functionname> [(parameter list)] As return type

Vb statements

End Function
```

Example:

```
Module module1

 Function FunctionDivide(ByVal num1 As Integer,

   ByVal num2 As Integer) As Integer



    Dim res As Integer



    If (num2 <> 0) Then

    res = num1/num2

       return res

    Else

      Console.WriteLine("Divide by Zero is undefined")

    End If

 End Function

 Sub Main()

    Dim a As Integer

    Dim b As Integer

    Dim res As Integer

    Console.Write("Enter Number 1")

    a = Console.ReadLine()

    Console.Write("Enter Number 2")

    b = Console.ReadLine()

    res = FunctionDivide(a, b)
    Console.WriteLine(res)
 End Sub
End Module
```

Output:



```
Assembly 'a, Version=0.0, Culture=neutral, PublicKeyToken=null' saved successfully to '/home/a.out'.
Compilation successful
Compilation took 00:00:01.0399660
Enter Number 1 10
Enter Number 2 0
Divide by Zero is undefined
0


...Program finished with exit code 0
Press ENTER to exit console.
```

Comparison between SubProcedure and Function:

Parameters	Sub Procedures	Functions
1	A subprocedure is not associated with an event.	A function is also not associated with an event.
2	A subprocedure is called, whenever it is required to perform certain tasks. It returns control to the calling code after performing a task.	A function is called, whenever a value is required to be returned to the calling code after performing a task.
3	A subprocedure does not return a value to the calling code.	Functions return a value to the calling code.
4	A sub procedure cannot be used with an expression.	Functions are used in an expression.
5	Subprocedure helps to make the code readable, and easy to modify and debug.	In functions, it is not easy to modify and debug the code.
6	Sub procedure is a generalized type of function.	A function is a specific type of procedure
7.	A subprocedure is declared with the keyword Sub.	A function is declared with the keyword Function.