**Event Handling**

Events are basically a user action like key press, clicks, mouse movements, etc., or some occurrence like system generated notifications. Applications need to respond to events when they occur.

Clicking on a button, or entering some text in a text box, or clicking on a menu item, all are examples of events. An event is an action that calls a function or may cause another event. Event handlers are functions that tell how to respond to an event.

VB.Net is an event-driven language. There are mainly two types of events −

Mouse events

Keyboard events

**Handling Mouse Events**

Mouse events occur with mouse movements in forms and controls. Following are the various mouse events related with a Control class −

MouseDown − it occurs when a mouse button is pressed

MouseEnter − it occurs when the mouse pointer enters the control

MouseHover − it occurs when the mouse pointer hovers over the control

MouseLeave − it occurs when the mouse pointer leaves the control

MouseMove − it occurs when the mouse pointer moves over the control

MouseUp − it occurs when the mouse pointer is over the control and the mouse button is released

MouseWheel − it occurs when the mouse wheel moves and the control has focus

The event handlers of the mouse events get an argument of type MouseEventArgs. The MouseEventArgs object is used for handling mouse events. It has the following properties −

Buttons − indicates the mouse button pressed

Clicks − indicates the number of click

X − indicates the x-coordinate of mouse click

Y − indicates the y-coordinate of mouse click

**Example**

Following is an example, which shows how to handle mouse events. Take the following steps −

Add three labels, three text boxes and a button control in the form.

Change the text properties of the labels to - Customer ID, Name and Address, respectively.

Change the name properties of the text boxes to txtID, txtName and txtAddress, respectively.

Change the text property of the button to 'Submit'.

Add the following code in the code editor window

Public Class Form1

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

' Set the caption bar text of the form.

Me.Text = "Mouse Events"

End Sub

Private Sub txtID\_MouseEnter(sender As Object, e As EventArgs) Handles txtID.MouseEnter

'code for handling mouse enter on ID textbox

txtID.BackColor = Color.CornflowerBlue



txtID.ForeColor = Color.White

End Sub

Private Sub txtID\_MouseLeave(sender As Object, e As EventArgs) Handles txtID.MouseLeave

'code for handling mouse leave on ID textbox

txtID.BackColor = Color.White

txtID.ForeColor = Color.Blue

End Sub

Private Sub txtName\_MouseEnter(sender As Object, e As EventArgs) Handles txtName.MouseEnter

'code for handling mouse enter on Name textbox

txtName.BackColor = Color.CornflowerBlue

txtName.ForeColor = Color.White

End Sub

Private Sub txtName\_MouseLeave(sender As Object, e As EventArgs) Handles txtName.MouseLeave

'code for handling mouse leave on Name textbox

txtName.BackColor = Color.White

txtName.ForeColor = Color.Blue

End Sub

Private Sub txtAddress\_MouseEnter(sender As Object, e As EventArgs) Handles txtAddress.MouseEnter

'code for handling mouse enter on Address textbox

txtAddress.BackColor = Color.CornflowerBlue

txtAddress.ForeColor = Color.White

End Sub

Private Sub txtAddress\_MouseLeave(sender As Object, e As EventArgs) Handles txtAddress.MouseLeave

'code for handling mouse leave on Address textbox

txtAddress.BackColor = Color.White

txtAddress.ForeColor = Color.Blue

End Sub

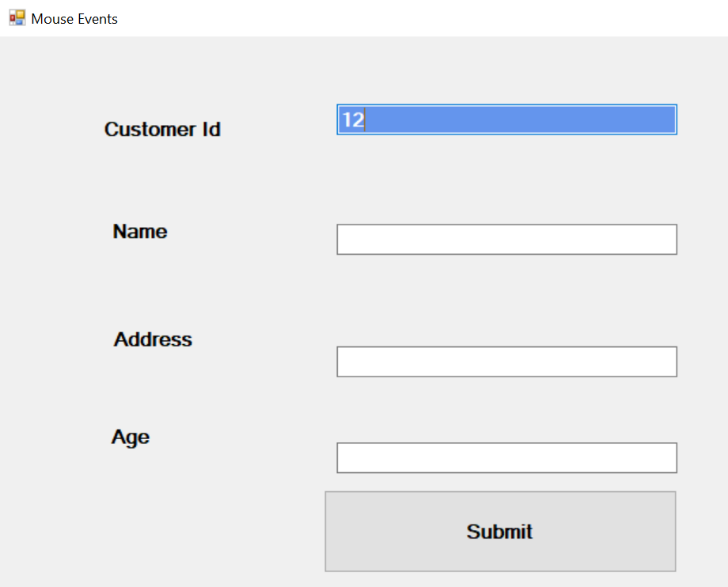
Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

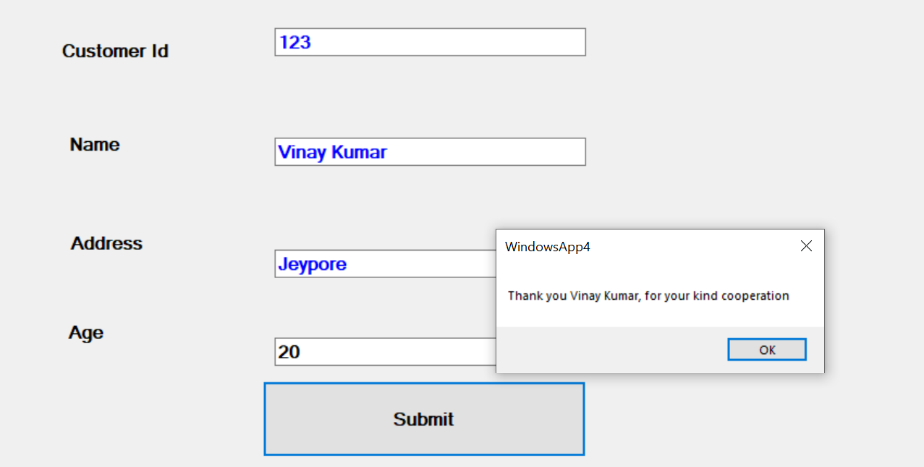
MsgBox("Thank you " & txtName.Text & ", for your kind cooperation")

End Sub

End Class

Output:





**Handling Keyboard Events**

Following are the various keyboard events related with a Control class −

KeyDown − occurs when a key is pressed down and the control has focus

KeyPress − occurs when a key is pressed and the control has focus

KeyUp − occurs when a key is released while the control has focus

The event handlers of the KeyDown and KeyUp events get an argument of type KeyEventArgs. This object has the following properties −

Alt − it indicates whether the ALT key is pressed

Control − it indicates whether the CTRL key is pressed

Shift − it indicates if the Shift key is pressed

The event handlers of the KeyDown and KeyUp events get an argument of type KeyEventArgs. This object has the following properties

Handled − indicates if the KeyPress event is handled

KeyChar − stores the character corresponding to the key pressed

**Example**

Let us continue with the previous example to show how to handle keyboard events. The code will verify that the user enters some numbers for his customer ID and age.

Add a label with text Property as 'Age' and add a corresponding text box named txtAge.

Add the following codes for handling the KeyUP events of the text box txtID.

Private Sub txtID\_KeyUP(sender As Object, e As KeyEventArgs) Handles txtID.KeyUp

If (Not Char.IsNumber(ChrW(e.KeyCode))) Then

MessageBox.Show("Enter numbers for your Customer ID")

txtID.Text = " "

End If

End Sub

Add the following codes for handling the KeyUP events of the text box txtID.

Private Sub txtAge\_KeyUP(sender As Object, e As KeyEventArgs) Handles txtAge.KeyUp

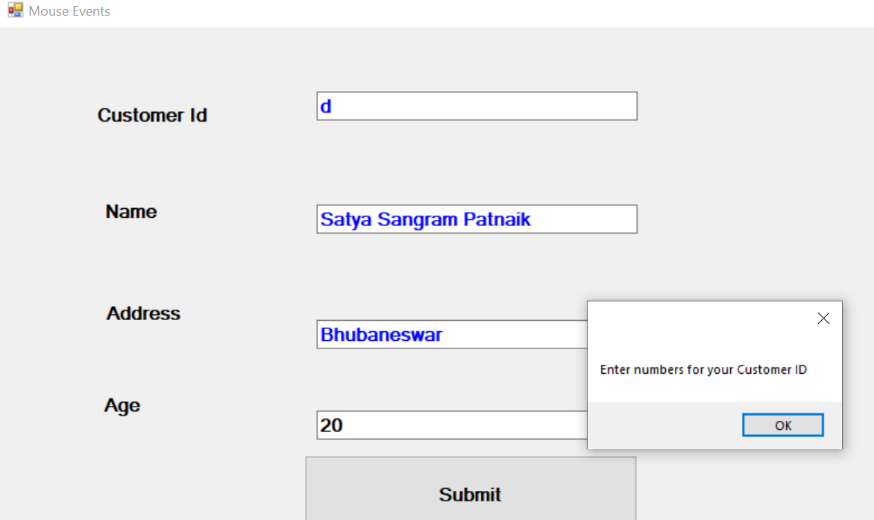
If (Not Char.IsNumber(ChrW(e.keyCode))) Then

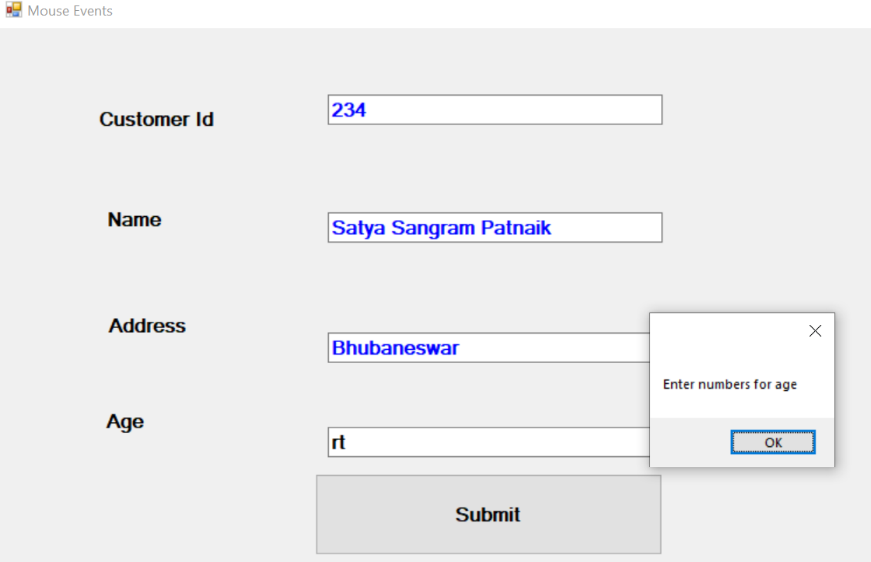
MessageBox.Show("Enter numbers for age")

txtAge.Text = " "

End If

End Sub

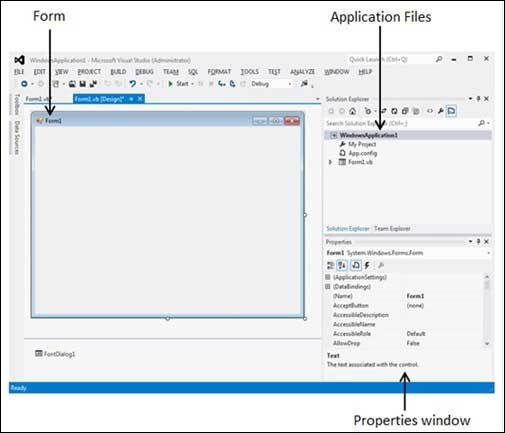




**Form Load Event**

Let's start with creating a Window Forms Application by following the following steps in Microsoft Visual Studio - File → New Project → Windows Forms Applications

Finally, select OK, Microsoft Visual Studio creates your project and displays following window Form with a name Form1



Visual Basic Form is the container for all the controls that make up the user interface. Every window you see in a running visual basic application is a form, thus the terms form and window describe the same entity. Visual Studio creates a default form for you when you create a Windows Forms Application.

Every form will have title bar on which the form's caption is displayed and there will be buttons to close, maximize and minimize the form.

Form Load event occurs before a form is displayed for the first time.

**Example:**

Type in the following as the code for the Load Event:

MessageBox.Show("Form Load Event")

Run your program. You should see the message box display before the Form loads.

To switch off your Button before the Form loads, add this to your code:

Button1.Enabled = False

Run your program again. You should see that button is no longer available for clicking on.