

The Timer



The Properties window for a Timer control. The title bar reads "Properties". The main area shows the control name "Timer1" and its type "System.Windows.Forms.Timer". Below this is a table of properties:

+ (ApplicationSettings)	
(Name)	Timer1
Enabled	False
GenerateMember	True
Interval	1
Modifiers	Friend
Tag	

Below the table, there is a section titled "Interval" with the description: "The frequency of Elapsed events in milliseconds." At the bottom of the window, there are tabs for "Properties" and "Solution Explorer".

The most important property of the timer tool is the interval property. The value of the interval property determines how often the timer control fires.

A value of 1000 sets the timer to fire once per second. A value of 1 causes the timer to fire 1000 times per second or once every millisecond.

```
Private Sub Timer1_Tick(ByVal sender As System.Object  
  
    'code to be run when timer is fired  
  
End Sub
```

Countdown Timer

Beginning number

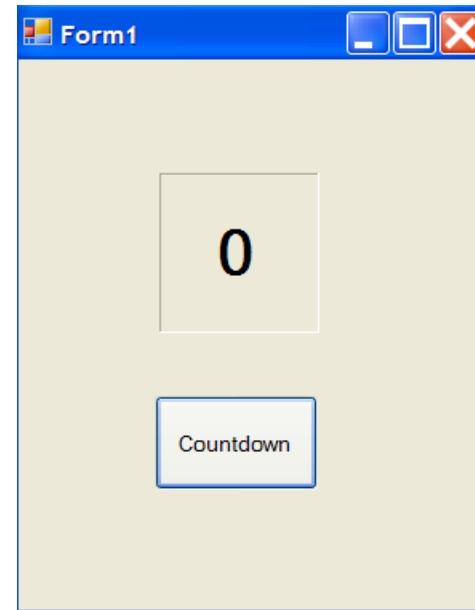
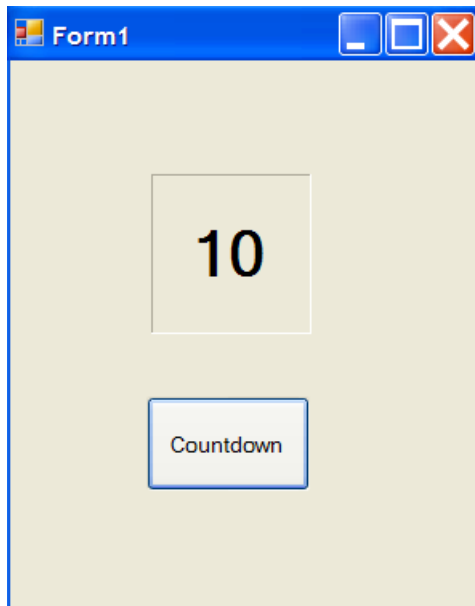
```
Public Class Form1
    Dim number As Integer = 10

    Private Sub Timer1_Tick(ByVal sender
        If (number > 0) Then
            number = number - 1
            lblCountDown.Text = number
        End If
    End Sub

    Private Sub btnCountDown_Click(ByVal
        Timer1.Start()
    End Sub
End Class
```

Timer fires every second causing the value of number to be decrement by 1 and be displayed in a label.

btnCountDown button starts timer

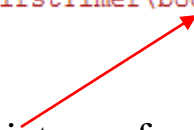


Ka-boom!

Modify the previous program so that an image is display when the countdown reaches 0.(Rocket launch,explosion etc.)


```
Public Class Form1
    Dim number As Integer = 10
    Dim path As String = "C:\Documents and Settings\James\My Documents\Visual Studio 2005\Projects\FirstTimer\boom.jpg"
    Dim imgBoom As Image = Image.FromFile(path)
```

Pretty picture of an explosion!



```
Private Sub Timer1_Tick(ByVal sender As System.
    If (number > 0) Then
        number = number - 1
        lblCountDown.Text = number
    Else
        Timer1.Stop()
        lblCountDown.Visible = False
        Me.BackgroundImage = imgBoom
    End If
End Sub
```

When countdown reaches 0, make label visible property false and display image as background image of form.



```
Private Sub btnCountDown_Click(ByVal sender As
    Timer1.Start()
    btnCountDown.Visible = False
End Sub
```

Exciting Moments in Visual Basic Programming!!



The Sub()

In our programmin, we have seen a lot of examples of sub(), short for sub-routine, like the one shown below.

These subs() have been created for us by double-clicking on the various toos we have drawn on the form like buttons, text boxes and even forms themselves.

```
Private Sub btnCountDown_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCountDown.Click
    Timer1.Start()
    btnCountDown.Visible = False
End Sub
```

There will be times when it will be useful to have sections of code run that are triggered, not by form components like buttons, but by other sections of code.

Here is an example of a sub called 'MySub' that is created by the programmer.

Every Sub has the keyword sub in its declaration.

```
Public Class Form1
    Private Sub MySub()
        MsgBox("I have created my first sub")
    End Sub
End Class
```

Every Sub ends with the keywords End Sub.

Calling A Sub()

```
Public Class Form1
    Private Sub MySub()
        MsgBox("I have created my first sub")
    End Sub
End Class
```

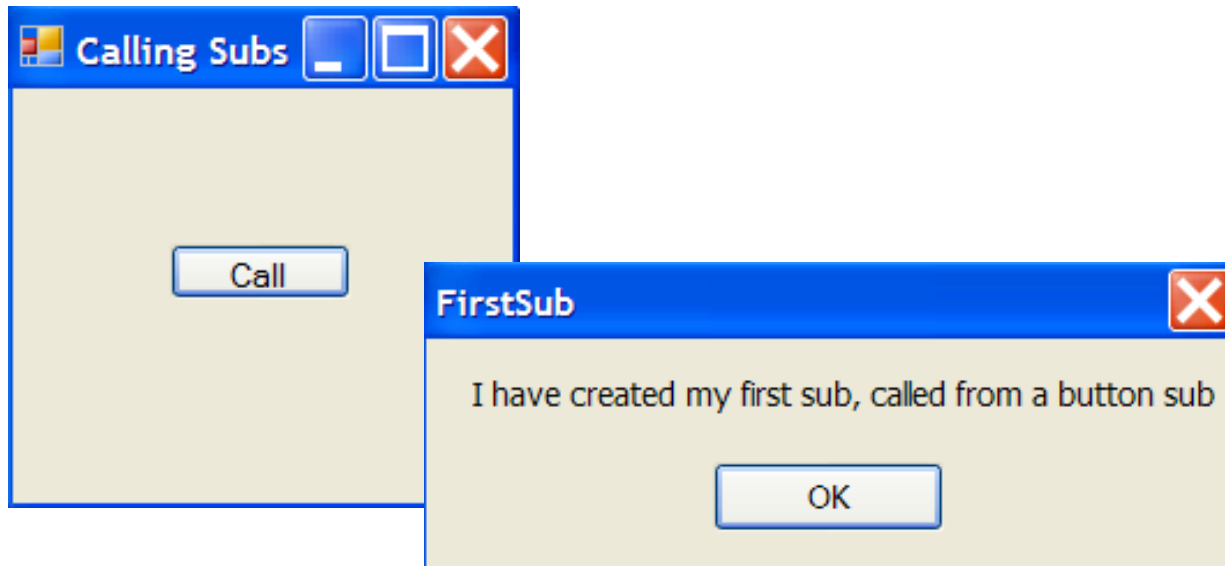
The purpose of the above Sub is obviously to display a message in a message box. In order for this Sub to run it must first be called.

Calling a Sub is very simple. To call the above Sub you would simply code the following:

MySub()

Subs can be called from other Sub such as buttons, form loads or even other programmer created subs.

Calling A Sub From A Button Sub

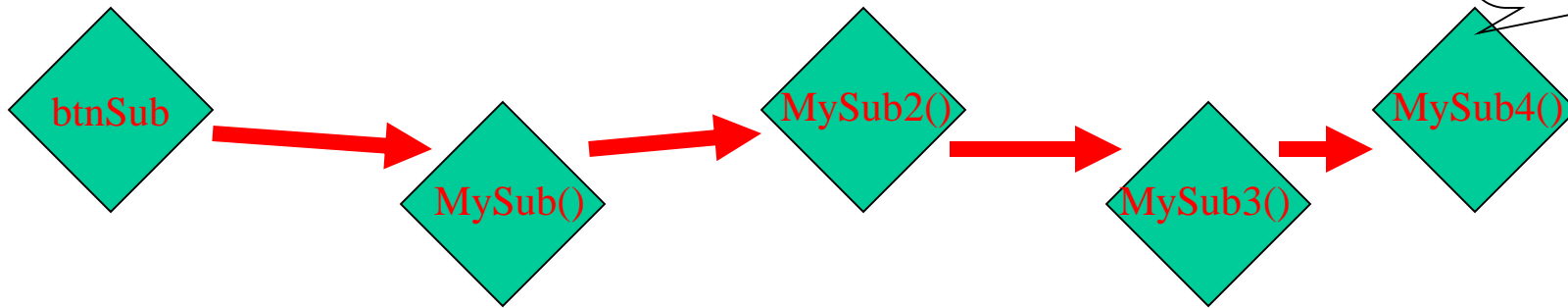


```
Public Class Form1
    Private Sub MySub()
        MsgBox("I have created my first sub, called from a button sub")
    End Sub

    Private Sub btnCall_Click(ByVal sender As System.Object, ByVal e As
        MySub()
    End Sub
End Class
```

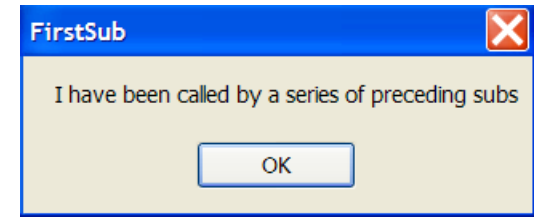
Clicking on the btnCall sub calls in turn MySub() which displays the message.

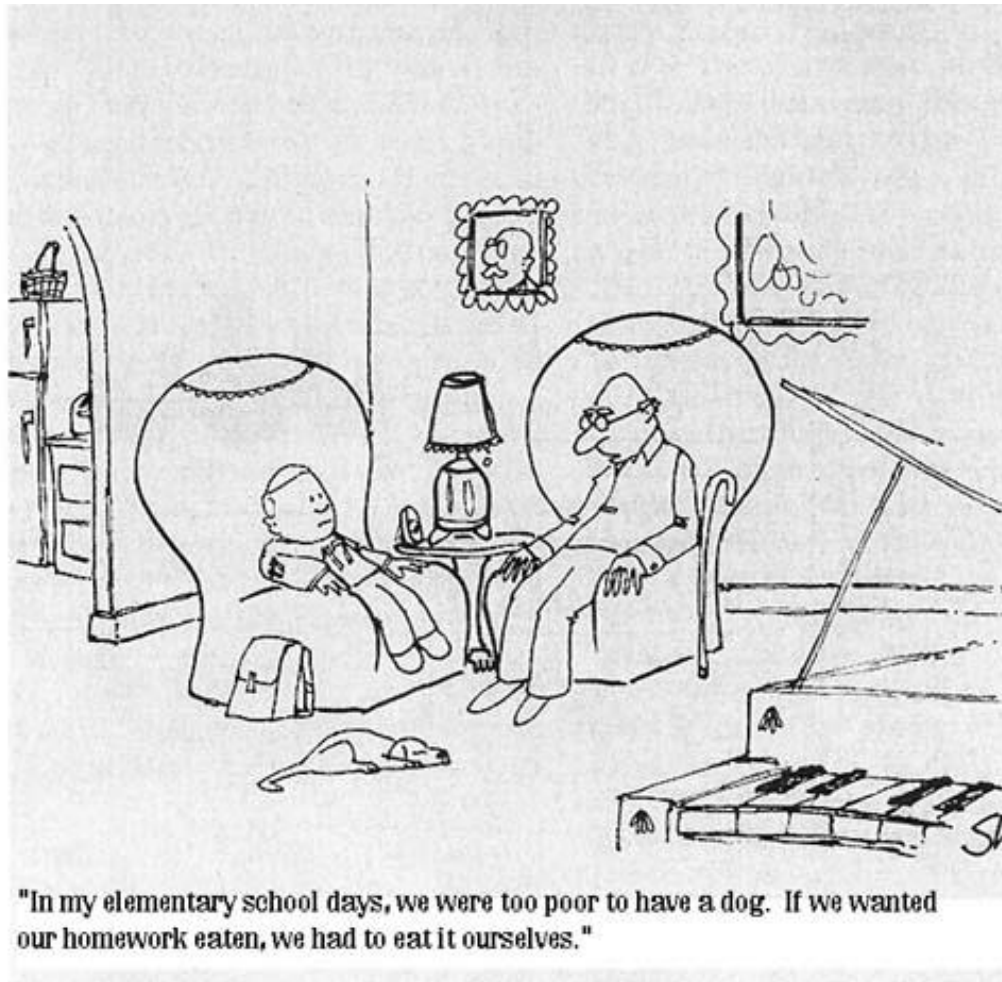
Subs calling subs calling subs calling subs....



I have been called by a series of preceding subs

```
Public Class Form1
    Private Sub btnCall_Click(ByVal sender As System.Object, ByVal e As EventArgs)
        MySub ()
    End Sub
    Private Sub MySub ()
        MySub2 ()
    End Sub
    Private Sub MySub2 ()
        MySub3 ()
    End Sub
    Private Sub MySub3 ()
        MySub4 ()
    End Sub
    Private Sub MySub4 ()
        MsgBox("I have been called by a series of preceding subs")
    End Sub
End Class
```





Me.BackColor = System.Drawing.Color.FromArgb(red, green, blue)