Year 12 : Visual Basic Tutorial.

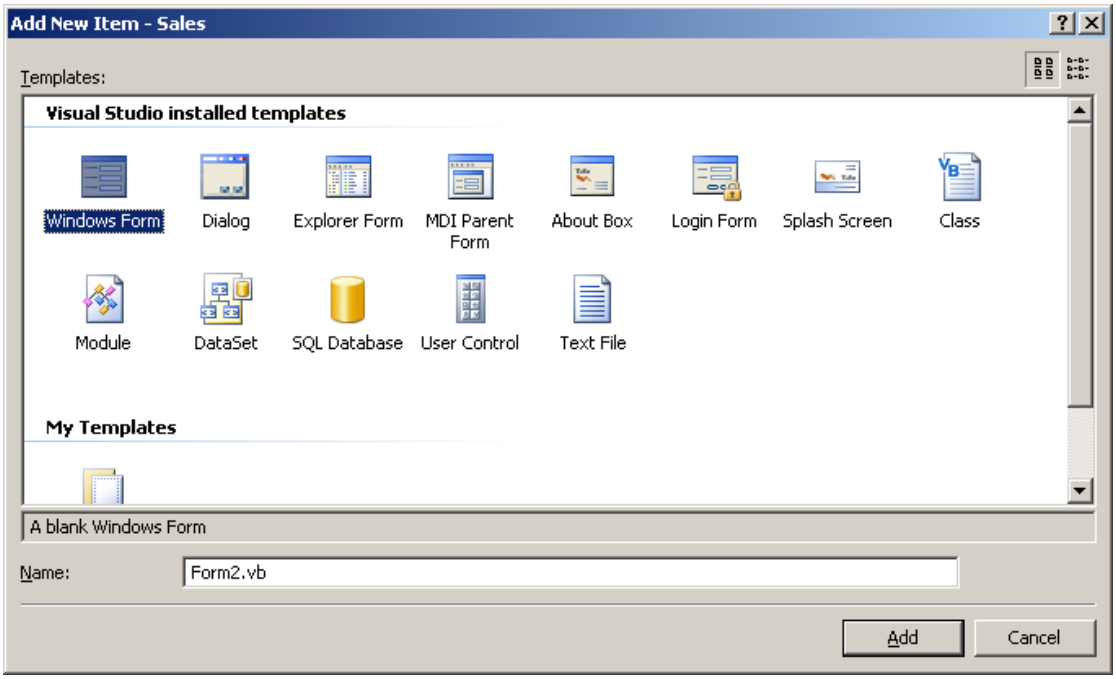
STUDY THIS

Forms.

Most applications have more than one form. Make sure you call them meaningful names (not Form1, Form2 etc!!!). Each form is saved as a different file on disc.

Each form has its own objects, properties, methods and event handlers.

To add a new form to a project…
In the [Project] menu – Add Windows Form…Select Windows Form…



…and change the name of the form to something meaningful.

The new form should appear in the Project Explorer window of your project

My

There is a special object called **My**. This object allows you to access the forms, computer and application of your project easily.

To open a form (called MyForm) in a subroutine, use…

```
My.Forms.MyForm.Show()
```

and to close a form….

```
My.Forms.MyForm.Hide()
```

…but you need to be careful!! A form cannot refer to itself…you need to use…

Me

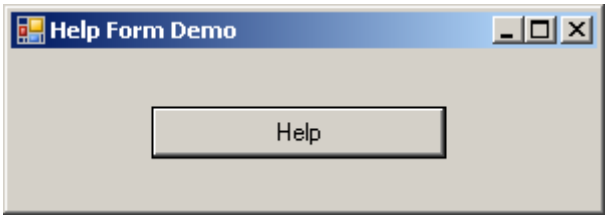
The **Me** object refers to the currently active form. So if you have a form with a button on it, and you want to close the form when the button is clicked you need to use…

```
Me.Hide()
```

HANDS ON

[1] Create a new Windows application.

Add a **Button** (btnHelp)



[2] Add another **Form** to the application and name it frmHelp.

On this form, place a **Button** (btnReturn).

[3] On the Click event of btnHelp enter the code…

```
    Private Sub btnHelp_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnHelp.Click
        My.Forms.frmHelp.Show()
    End Sub
```

[4] On the Click event of btnReturn enter the code…

```
Private Sub btnReturn_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnReturn.Click
        Me.Hide()
End Sub
```

[5] Run the form and you should be able to open the new form …and then close it.

HINT :

You can set the **StartPosition** property of a form to **CenterScreen** to place it in the middle of the screen when it is first opened.

STUDY THIS

Dialog Boxes

There are some special forms already created for you. A form that collects information from the user is called a **Dialog Box**.

An example of a Dialog Box is the **ColorDialog** object that allows the user to select a colour from a palette. The dialog box has its own properties that the programmer can set and then it is opened using the **ShowDialog** method. We will use this one in the next example.

Other Dialog Boxes available are:

- **OpenFileDialog** – for opening files
- **SaveFileDialog** – for saving files
- **FontDialog** – for setting font properties
- **FolderBrowserDialog** – for navigating through a disc's hierarchical folder structure.
- **PrintDialog** – Sets printing options
- **PrintPreviewDialog** – displays a print preview
- **PageSetupDialog** – for setting the propertuis of a page.

HANDS ON

[1] Create a new Windows Application

Add a **Button** (btnColour; Text = 'Set Font Colour') , and a **Label** (lblMessage; Text = "Test Message")

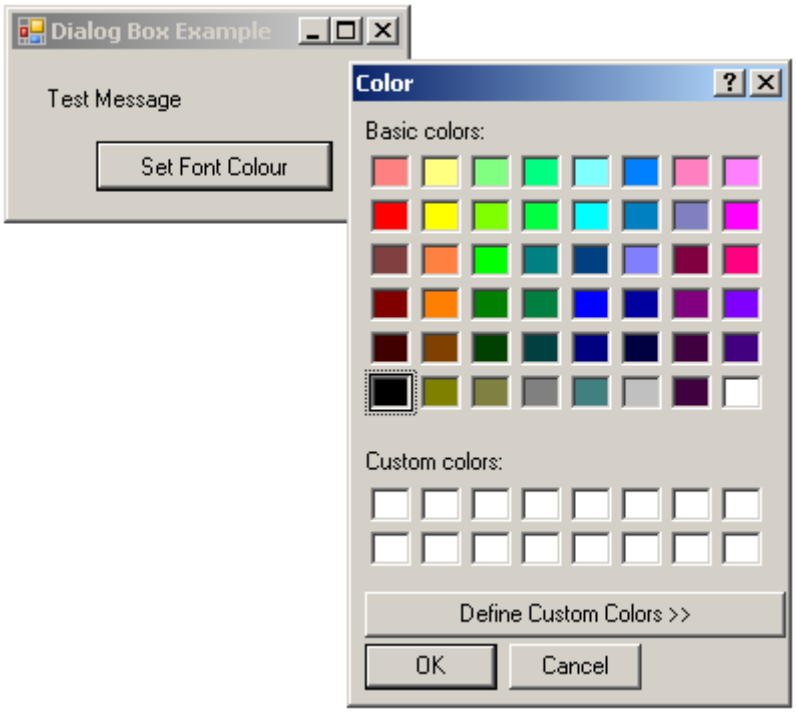
Also drag a **ColorDialog** object from the Toolbox onto the form. It should appear in the space below with the name ColorDialog1

[2] On the Click event handler of the button, enter the following code:

```
Private Sub btnColour_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnColour.Click
        ColorDialog1.ShowDialog()
End Sub
```

This code should open the Colour Dialog box.

Run the program and you should see the standard Windows colour selection dialog box....



[3] There is one more line needed in the code – one that sets the colour of the font of the label to the colour selected in the dialog box....

```
Private Sub btnColour_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnColour.Click
        ColorDialog1.ShowDialog()
        lblMessage.ForeColor = ColorDialog1.Color
End Sub
```

Run the program now, and you should be able to set the label's font colour...