Exercise Program (A Calculator project)

In the next few pages, you're going to create a Calculator. It won't be a very sophisticated calculator, and the only thing it can do is add up. What the project will give you is more confidence in using variables, and shifting values from one control to another. So create a new project, save it as **Calculator**, and let's get started.

Designing the Form

Let's design the form first. What does a calculator need? Well numbers, for one. A display area for the result. A plus sign button, an equals sign button, and a clear the display button.

Here's how our calculator is going to work. We'll have 10 buttons for the numbers 0 to 9. When a button is clicked its value will be transferred to a display area, which will be a Textbox. Once a number is transferred to the Textbox we can click on the Plus button. Then we need to click back on another number. To get the answer, we'll click on the equals sign. To clear the display, we'll have a Clear button.

To your new form, first add ten Buttons (You can add one, then copy and paste the rest). The Buttons should have the following Properties:

> **Name**: ***btn*** Plus a Number (btnOne, btnTwo, btnThree, etc)
> **Text**: A number from 0 to 9. A different one for each button, obviously
> **Font**: MS Sans Serif, Bold, 14

Next, add a Textbox. Set the following properties for the Textbox:

Textbox

> **Name:** txtDisplay
> **Font**: MS Sans Serif, Bold, 14
> **Text**: Erase the default, Textbox1, and leave it blank

Three more buttons need to be added

Plus Button

> **Name**: btnPlus
> **Font**: MS Sans Serif, Bold, 14
> **Text**: +

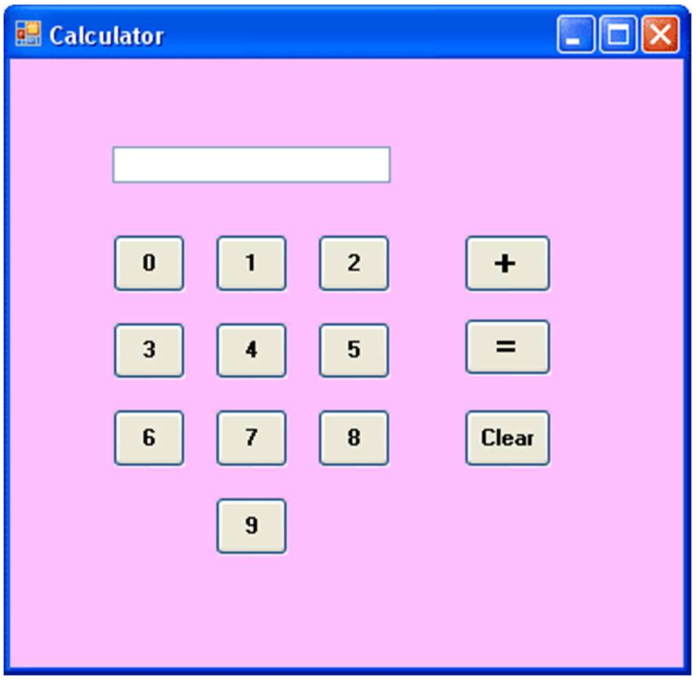
Equals Button

Name: btnEquals
Font MS Sans Serif, Bold, 14
Text: =

Clear Button

Name: btnClear
Font MS Sans Serif, Bold, 14
Text: Clear

When your form design is finished, it might look something like this:



So if you wanted to add 5 + 9, we'd do this:

1. Click first on the 5
2. A 5 appear in the textbox
3. Click the + symbol
4. The 5 disappears from the textbox
5. Click on the number 9
6. A 9 appears in the textbox
7. Click on the = symbol
8. The 9 disappears from the textbox
9. The answer to 5 + 9 appears in the textbox
10. Click the "Clear" button to clear the textbox

The Code for the VB NET Calculator

You might be thinking that all this is terribly complicated at such an early stage. But it isn't really. All we are doing is transferring the Text Properties from the Buttons to the textbox. And you already know how to do that. The number buttons don't do anything else. All the work is done with the Plus button and the Equals buttons. And there are only two lines of code needed for the Plus button, and three for the Equals button.

So far, when you've set up a variable, you've set them up behind a Private Subroutine. Like this:

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

        Dim Variable1 As String

End Sub
```

Suppose you had another button on the form, Button2, and the code was this

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

        Dim Variable2 As String

End Sub
```

How can you access what's in **Variable1** from Button2? The answer is, you can't. It's like two people sitting at desks in cubicles. Each person has written something on a piece of paper. They can't see into the other person's cubicle, only whatever is their own cubicle. So how do they share their information?

In VB you can set up your variable declarations outside of the code for a Button. That way, more than one Button can see the code. You can place your variable declarations right at the top of the code window, just beneath the line that begins "Public Class Form1". We'll set up two Single variables there, **total1** and **total2**:

```
1  Public Class Form1
2      Dim total1 As Single
3      Dim total2 As Single
4
```

Now all the controls on your form can see these two variables. Those Buttons you set up can put something in them, and every button has the ability to see what's inside them.

The 0 to 9 Buttons

The Buttons with the Text 0 to 9 only need to do one thing when the button is clicked - have their Text Properties transferred to the Textbox.

So double click the 0 Button and enter the following code:

```
Private Sub btnZero_Click(sender As System.Object, e As System.EventArgs) Handles btnZero.Click

        txtDisplay.Text = btnZero.Text

End Sub
```

This code will transfer the Text Property of a Button called btnZero to the Text Property of a Textbox called txtDisplay. Run your programme and try it out. When the programme is running, click the 0 button to see that it does indeed transfer the Text on the Button to the textbox

Except, there's a problem with that code. If you wrote similar code for all ten of your number buttons, the calculator wouldn't be right. Why is that? Have you spotted what's wrong? What happens when you transfer the number 2 to the Textbox, and then click the number 3? The number 2 will disappear, to be replaced by the number 3. Which is all right if all you wanted to do was add up single numbers, but not much good if you actually wanted the number 23 in the Textbox.

So how do we solve this problem? How do we fix it so that we can have two or more numbers appearing in our Textbox?

What we need is a way to get whatever is in the Textbox to stay where it is, and not disappear on us when we click a different number. It's quite easy. It's this:

txtDisplay.Text = txtDisplay.Text & btnZero.Text

So now we're saying the textbox doesn't just contain the Text on the Button. It must also keep whatever is inside the textbox as well.

So what you need to do now is to add that code to all of your ten number Buttons. Obviously it won't be exactly the same. For the button called btnOne the code would be this:

txtDisplay.Text = txtDisplay.Text & btnOne.Text

When you've finished coding all ten buttons, run the program and click all ten number buttons to see if they do indeed transfer the numbers on the caption to the textbox. Not only that, but test to see if you can have more than one number in the textbox.

The plus + Buttons

Now that we can get numbers into our Textbox display area, we'll write code to do something with those numbers - add them together, in other words. What we're trying to do is this: Click on the Plus symbol and make the number in the Textbox disappear. Before the number vanishes, we can store it in a variable. The variable we're going to be storing the number in is one of those variables we set up at the top of the code. It's this one:

```
Dim total1 As Single
```

We've already seen how to retain a value from a textbox and add it to something else:

```
txtDisplay.Text = txtDisplay.Text & btnZero.Text
```

Here, we kept the value that was already in the textbox and joined it to the Text property of the button.

If we're going to be adding two numbers together, we need Visual Basic to remember the first number. Because when we click the equals sign, that first number will be added to the second number. So if we haven't asked VB to remember the first number, it won't be able to add up.

The variable we're going to be storing the first number in is total1. We could just say this:

```
total1 = txtDisplay.Text
```

Then whatever is in the textbox will get stored in the variable total1.

Except we want VB to remember our numbers. Because we might want to add three or more numbers together: 1 + 2 + 3 + 4. If we don't keep a running total, there's no way for our program to remember what has gone before, it will just erase whatever is in total1 and then start again.

So we need to "tell" our program to remember what was in the variable. We do it like this:

```
total1 = total1 + Val(txtDisplay.Text)
```

That Val( ) part just makes sure that a number in a textbox is kept as a number, and not as text. It's short for Value. The important part is the total1 + txtDisplay.Text. We're

saying "The variable total1 contains whatever is in total1 added to the number in the textbox." An example might clear things up. Suppose we were adding 5 + 9. Now, suppose total1 already contained the number 5, and that number 9 was in the textbox. It would work like this:

Finally, we need to erase the number in the textbox. To erase something from a textbox, just set its Text property to a blank string. We do this with two double quotation marks. Like this:

txtDisplay.Text = ""

That tiny bit of code will erase the Text a textbox. Another way to erase text from a textbox is to use the Clear method. After you typed a full stop, you probably saw the drop down list of Properties and Methods. Scroll up to the top, and locate the word Clear. Double click "Clear" and the drop down list will close.

txtDisplay.Clear( )

So the whole code for our Button called btnPlus is this:

```
Private Sub btnPlus_Click(sender As System.Object, e As System.EventArgs)
Handles btnPlus.Click
        total1 = total1 + Val(txtDisplay.Text)
        txtDisplay.Clear()
End Sub
```

Add that code to your Plus button. All we've done with that code is to store numbers into our variable total1 and then erase whatever was in the textbox.

The equals = Buttons

We now need to write the code for the equals button. There are only three lines in total, and here's a little help.

You need to use the other variable that was set up at the top of the coding window, total2. The variable total1 will be added to whatever is total2

The first line of code will be this

total2 = total1 + (something missing here)

Your job is to supply the missing code. In other words, replace "(something missing here)"

Remember that **total1** contains the first number to be added. And you know where that came from. The only thing left to do is to add the second number.

For the second line of code, you need to transfer the **total2** variable to the textbox.

For the third line of code, all you are doing is resetting the variable total1 to zero. That's because after the equals button has been pressed, we have finished adding up. If you wanted to do some more adding up, that total1 will still hold the value from the last time. If you reset it to zero, you can start adding some new numbers.

The only thing left to do is to write a single line of code for the Clear button. All you are doing there is erasing text from a textbox. Which you already know how to do.

When you're finished, you should have a simple calculator that can add up integers.