Year 12 : Visual Basic Tutorial.

STUDY THIS

Input and Output
(Text Boxes)

The three stages of a computer process…

- Input
- Processing
- Output

Data is usually input using TextBoxes.

HANDS ON

[1] Create a new Windows Application Project called '**Calculator**'.

Set the Form properties:

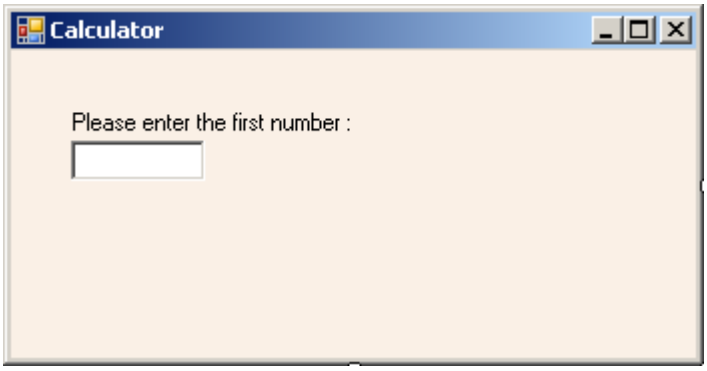
Property	Value
Name	frmCalculator
BackColor	Linen
Text	Calculator
Size	350,180
StartPosition	CentreScreen

[2] Place a TextBox on the form with the following properties:

Property	Value
Name	txtFirstNumber
Location	30,45
Size	67,20

And a Label with the properties:

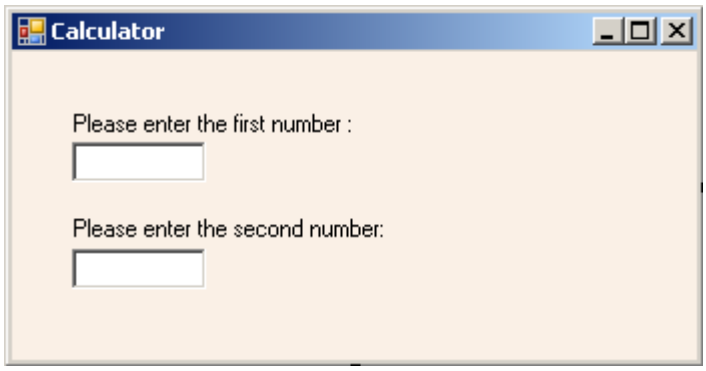
Property	Value
Name	lblFirst
Location	27,29
Text	Please enter the first number:



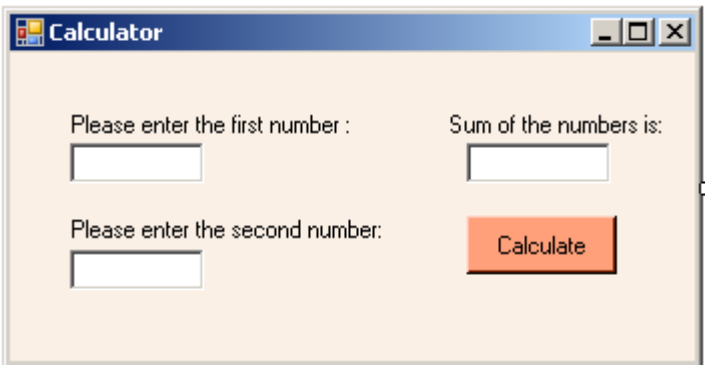
Your form should look like this.

[3] Add another TextBox (txtSecondNumber) and Label (lblSecond) to the form and line them up so the form looks like :

(**NOTICE** the lines on the form that help you line objects up)



[4] Add another TextBox (txtAnswer), a Label (lblAnswer) and a Button (btnAnswer) to your form…



Arrange your objects to look similar to this diagram.

This is the end of the first stage of your program - Creating the interface by adding objects to your form and setting the properties.

[5] Stage 2 – Adding the code for the event handlers….

Only one event-handler to write. The Click event of the Button btnAnswer.

Double-click on btnAnswer to open the Code Window…and type this subroutine….

```
Private Sub btnAnswer_Click

    'Declare the variables
    Dim First As Integer
    Dim Second As Integer

    'Assign values to the variables from the Inputs
    First = txtFirstNumber.Text
    Second = txtSecondNumber.Text

    'Output the answer
    txtAnswer.Text = First + Second

End Sub
```

STUDY THIS

Explanation:

[a] The Green lines are **comments**. These are important! They explain what each section of code does. These are a useful reminders for YOU…and should always be included.

Also useful is 'white space' the blank lines between sections of the code.

[b] **Variables** are quantities that may be different each time a program is run. The computer needs to know what variables you are going to use and what type they are. (see List at the end of this section)

Your subroutine uses two variables called **First** and **Second**. They are both of **integer** type.

[c] First = txtFirstNumber.Text
The input line. This line assigns the value of the Text property of txtFirstNumber to the variable **First**. In other words the value of **First** becomes the number in the Textbox at the time the button is clicked.

Make sure you fully understand how an assignment statement works – you will be using them a lot!

A = B
….assigns the value of B to the variable A. This means that the value of A changes …but the value of B does not.

[d] txtAnswer.Text = First + Second
The output line - txtAnswer will display the result of the calculation.

This is really another assignment statement, where the value of the Text property of txtAnswer is given the value that is the sum of the two numbers.

Phew! – Some important stuff there!

HANDS ON

[6] **Run** the program and check that it works.
(Input two numbers and click the button)

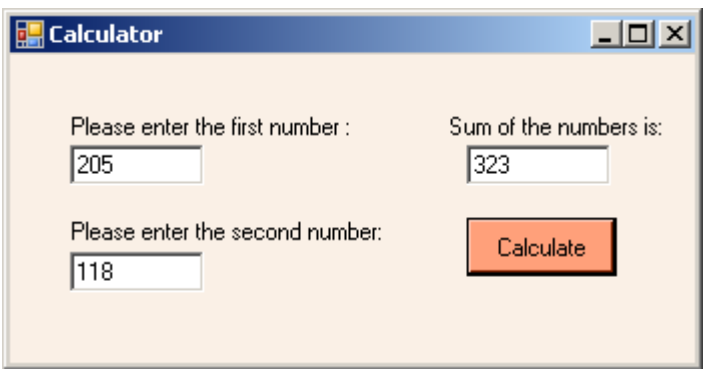


Table of Data Types.

STUDY THIS

Data Type	Comment	Size	Example
Char	Any character	1 byte	'A'
String	Up to about 2 billion characters	2 bytes per character	'Tom Jenkins'
Byte	0 to 255	1 byte	29
SByte	-128 to 127	1 byte	-3
Short	-32,768 to 32,767	2 bytes	3278
UShort	0 to 65,535	2 bytes	49312
Integer	-2,147,483, to 2,147,483,647	4 bytes	629,439
UInteger	0 to 4,294,967,295	4 bytes	3,120,000,000
Long	Massive whole numbers	8 bytes	7,444,555,666,777
ULong	Massive whole numbers	8 bytes	32,456,457,645,999
Single	Real numbers	4 bytes	125.99
Double	Real numbers	8 bytes	3.14159265
Decimal	Real Numbers	16 bytes	36,689.87514
Boolean	True or False	1 bit	True
Date	Jan 1st, 0001 to Dec 31st, 9999	8 bytes	6/3/2012

When a variable is declared in a program, it is really an instruction to the computer to reserve some space in memory, where the value of that variable will be stored. The amount of memory space reserved depends on the type of the variable. (see above table)…so it is good programming practice to declare variables as small types whenever possible.

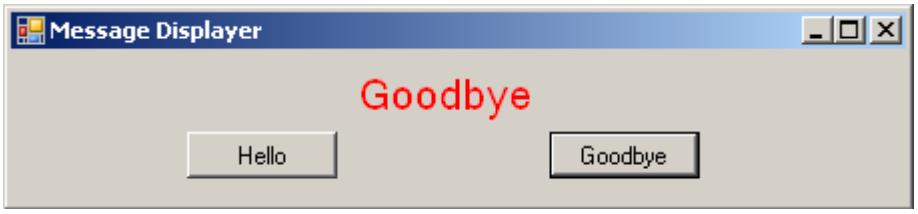
Example - Don't use an Integer when a Byte would do.

When a subroutine has run and finished, the space reserved for local variables declared in that subroutine will be released.

Visual Basic Challenges 1

HANDS ON

[1] Create an application that has a Label and two buttons. When one of the buttons is clicked, the message reads 'Hello' in Green, and when the other button is clicked the message reads 'Goodbye' in Red.



[2] Create an application where a button displays the message "Hello World" in RED when the mouse button is pressed down, and in GREEN when the button is released. (HINT : Use The MouseDown and MouseUp events)

[3] Create an application which looks like this when run…

There is an invisible Label below the button.

When the program is run, the user enters a name into the TextBox and clicks on the button to reveal the message…

HINT : You can add text strings together…
"BULL" + "FROG" is the string "BULLFROG"

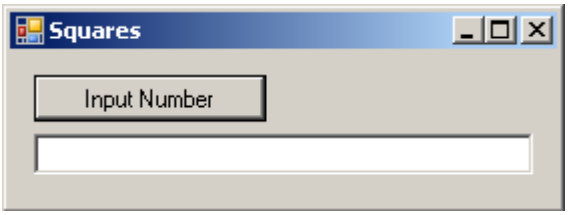
Other Methods of Input and Output
(InputBox; MsgBox)

HANDS ON

Another method of input involves using an **InputBox**….

[1] You are going to write a program that allows the user to input a number, and outputs its square (Eg Input :7 and output:49)

Create an application with a form that has a Button (btnDisplay) and a TextBox (txtMessage). Arrange the objects to look like this…



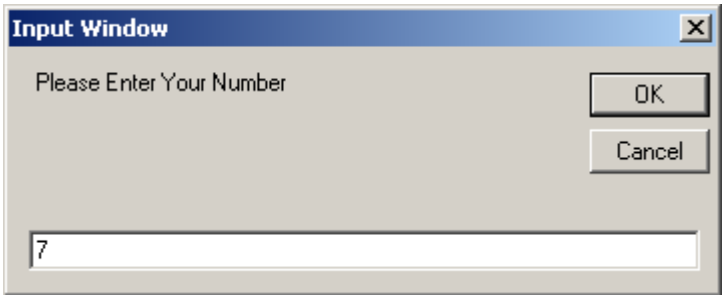
Enter the following event handler for the Click event of btnDisplay…

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        'Declare variables
        Dim Num As Integer
        Dim Answer As String

        'Open the Input Box and assign the value of Num
        Num = InputBox("Please Enter Your Number", "Input Window")

        Answer = "The square of " + Num.ToString + " is " + _
            (Num * Num).ToString

        'Display the message in the TextBox
        txtMessage.Text = Answer

    End Sub
```

The **InputBox** statement in your subroutine has two **parameters** – both are strings. The first string ("Please Enter Your Number") is the message prompt, the second ("Input Window") is the Window title.

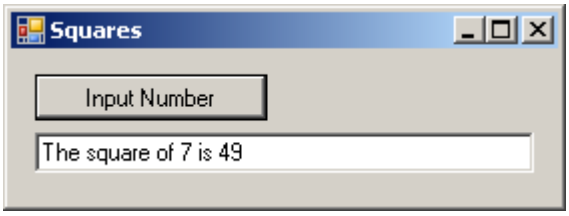
When the program is run, whatever is typed into the InputBox is returned as the value of variable Num… and this variable can then be used in your program.

[2] Run the program….



..and type in your number

When you click the OK button, the message should appear in your form….



NOTE : Strings may be added together, but sometimes you need to turn a number into a string first.

That is why you will see **Num.ToString** in the message

Also…If a line of code is too long, place a <space> and a _ character at the end of the line _ and continue on the next. (like the line above)

Using an MsgBox …. Another way to output data.

HANDS ON

[1] Create a new application and place a Button (btnOutput) on the Form.

Type in the event handler for the Click event of btnOutput…

```
Private Sub btnOutput_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnOutput.Click
        MsgBox("Keep Smiling!")
    End Sub
```

Run the program.

[2] Try changing the subroutine to…

```
Private Sub btnOutput_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnOutput.Click
        MsgBox("Are you still smiling?", MsgBoxStyle.Question,
"Output Demo")
    End Sub
```