

Lecture No 9 and 10:

Managing Variables:

Variables are like mail boxes in the post office. The contents of the variables change every now and then, just like the mail boxes. In term of VB, variables are areas allocated by the computer memory to hold data. Like the mail boxes, each variable must be given a name. To name a variable in Visual Basic, we have to follow a set of rules.

Variable Names

Rules for naming the variables in Visual Basic

- It must be less than 255 characters
- No spacing is allowed
- It must not begin with a number
- Period is not permitted
- Cannot use exclamation mark (!), or the characters @, &, \$, #
- Cannot repeat names within the same level of scope.

Examples of valid and invalid variable names are displayed in the following table.

Table 5.4: Examples of Valid and Invalid Variable Names	
Valid Name	Invalid Name
My_Car	My.Car
this year	1NewBoy
Long_Name_Can_beUSE	He&HisFather * & is not acceptable

Declaring Variables Explicitly

In Visual Basic, it is a good practice to declare the variables before using them by assigning names and data types. They are normally declared in the general section of the codes' windows using the **Dim** statement. We can use any variable to hold any data, but different types of variables are designed to work efficiently with different data types.

The syntax is as follows:

Dim VariableName As DataType

If we want to declare more variables, we can declare them in separate lines or we may also combine more in one line, separating each variable with a comma, as follows:

```
Dim VariableName1 As DataType1, VariableName2 As DataType2,  
VariableName3 As DataType3
```

For example

```
Dim password As String  
Dim yourName As String  
Dim firstnum As Integer  
Dim secondnum As Integer  
Dim total As Integer  
Dim doDate As Date  
Dim password As String, yourName As String, firstnum As Integer
```

Unlike other programming languages, Visual Basic actually doesn't require to specifically declare a variable before it's used. If a variable isn't declared, VB will automatically declare the variable as a Variant. A variant is data type that can hold any type of data.

For string declaration, there are two possible types, one for the variable-length string and another for the fixed-length string. For the variable-length string, just use the same format as above example. However, for the fixed-length string, we have to use the syntax as shown below:

```
Dim VariableName as String * n
```

where n defines the number of characters the string can hold.

For example,

Dim yourName as String *10

*yourName can holds no more than 10 Characters.

Operators in Visual Basic:

To compute inputs from users and to generate results, we need to use various mathematical operators. In Visual Basic, except for + and -, the symbols for the operators are different from normal mathematical operators, as shown in the following table.

Arithmetic Operators		
Operator	Mathematical function	Example>
^	Exponential	2^4=16
*	Multiplication	4*3=12,
/	Division	12/4=3
Mod	Modulus (returns the remainder from an integer division)	15 Mod 4=3
\	Integer Division(discards the decimal places)	19\4=4
+ or &	String concatenation	"Visual"&"Basic"="Visual Basic"

Conditional or Relational Operators:

To control the VB program flow, we can use various conditional operators. Basically, they resemble mathematical operators. Conditional operators are very powerful tools, they let the VB program compare data values and then decide what action to take, whether to execute a program or terminate the program and more.

Conditional Operators

Operator	Meaning
=	Equal to
>	More than
<	Less Than
>=	More than or equal
<=	Less than or equal
<>	Not Equal to

Logical Operators

In addition to conditional operators, there are a few logical operators shown in below.

Logical Operators

Operator	Description
And	Both sides must be true
Or	One side or other must be true
Xor	One side or other must be true but not both
Not	Negates true

* We can also compare strings with the operators. However, there are certain rules to follow where upper case letters are less than lowercase letters, and number are less than letters.

If Then Else Statements:

To effectively control the VB program flow, we shall use **If...Then...Else** statement together with the conditional operators and logical operators.

If conditions Then

VB expressions

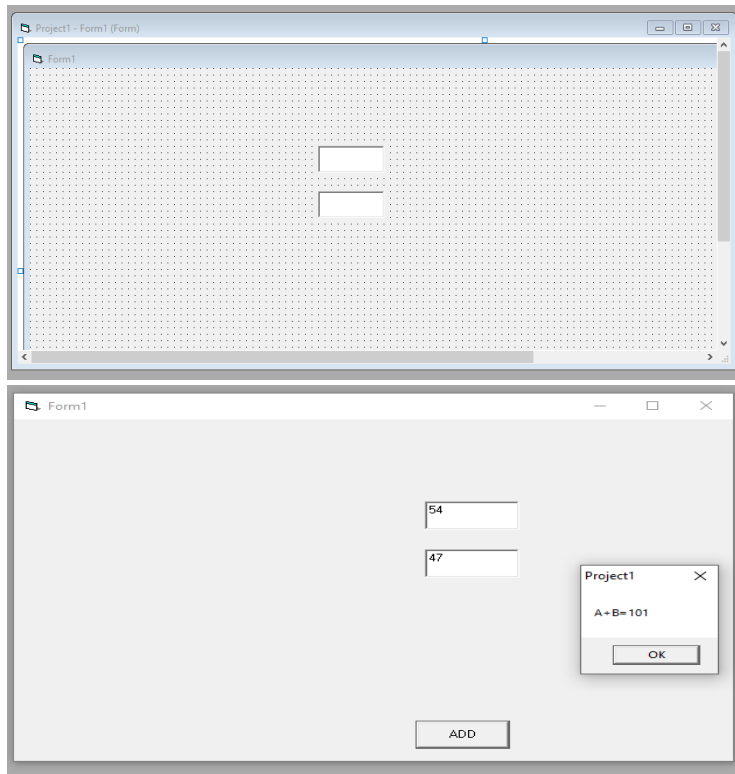
Else

VB expressions

End If

Lecture # 11:

Program for Arithmetic operation using Label and Msg Box :



```
Private Sub Command1_Click()
```

```
Dim A As Single
```

```
Dim B As Single
```

```
A = Text1.Text
```

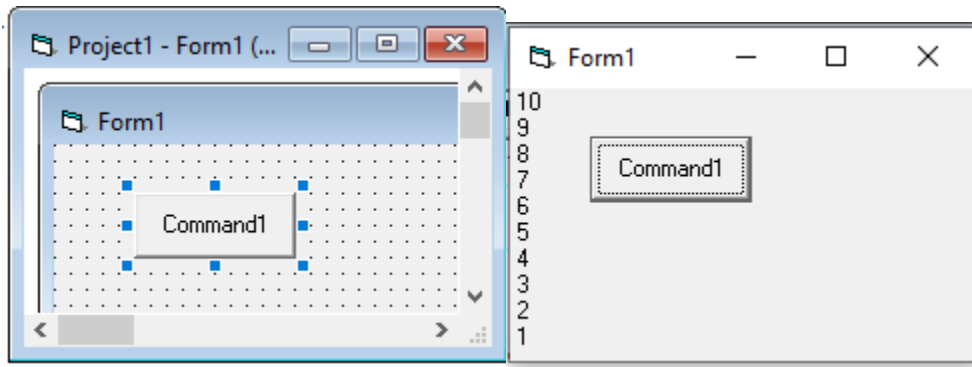
```
B = Text2.Text
```

```
C = A + B
```

```
MsgBox ("A+B=" & C)
```

```
End Sub
```

Simple Programme for printing numbers from 10 to 1, Step -1:



```
Private Sub Command1_Click()
```

```
Dim J As Integer
```

```
Dim N As Integer
```

```
For J = 1 To 10
```

```
Print J
```

```
Next J
```

```
End Sub
```