


Date : 

Page : / /

UNIT-3

* DATA TYPES *

In Visual Basic there is a wide range of data types.

The Visual Basic data types appear in following:-

* Scope of Variables :-

Depending upon, How a variable is declared. A variable has scope as either procedure level (local) or Module level (Global) when a variable is declared within a procedure on the code, within that procedure. We can use or change the value of that variable. Its scope is local to that procedure.

• Local Variable :-

The variable use within a procedure is known as local variable. We declare them with the keyword 'Dim' or 'Static'. Local variable declare with 'Dim' exist as long as the procedure is executed. Whereas local variable declare with 'Static' exist entire time our application is running.

Example :-

```
1) Private sub cmd1_click()
    Dim x as integer
    x = 100
    x = x + 20
    Print x
end sub.
```



```

2) Private sub cmx
    Static x as integer
    x = x + 20
    Print x
end sub.

```

• Global Variable :-

To make a variable available to other Module or procedure we use 'Public' keyword while declaring it, such variable is known as 'Global Variable' or 'Public Variable'. The value in Public variable are available to all procedure in our application. Public variable are declared in the General declaration section at top of Module.

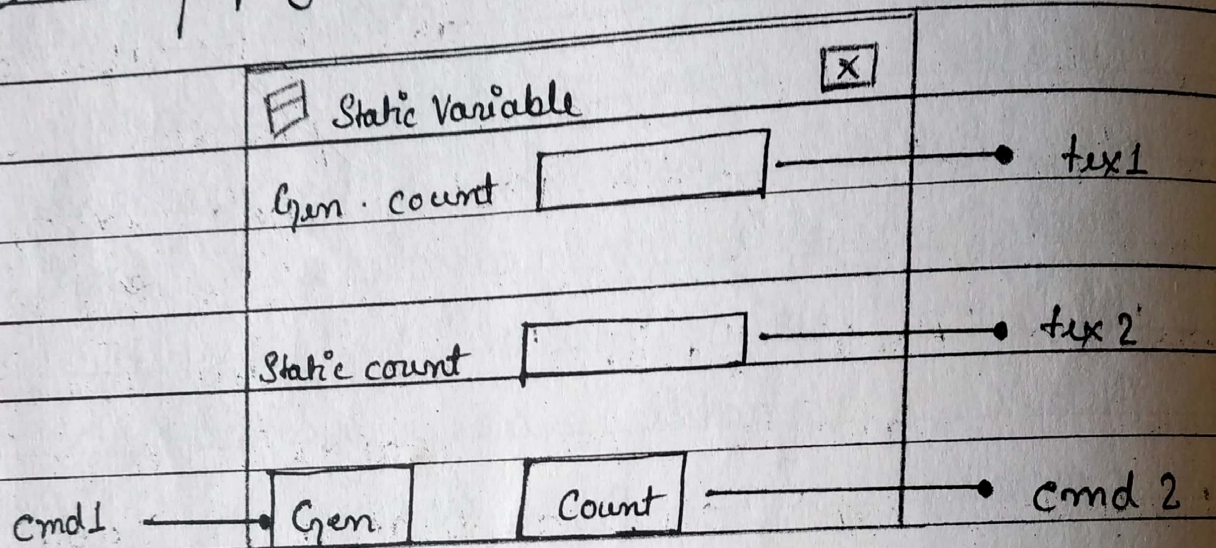
Form 1 (code)		X
General	Declaration	
Public A as integer		
Dim B as integer		

• Static Variable :-

The static variable declared with the keyword 'Static' variable declare with static exist the entire time our

application is running. This means that the static variable retains the last value assigned to it. Even the procedure has finished.

Example: A program to simulate static variable.



Private sub cmd1-click()

Dim x as integer

x = x + 1

tex1.text = x

end sub

Private sub cmd2-click()

Static x as integer

x = x + 1

tex2.text = x

end sub.

• CONSTANT :-

Some variable doesn't change its value during the execution of program. These are constant that appear

Many times in our code. We declare & assign value to constant on same line using 'const' keyword.

Example:- Const pi = 3.1415.

Q) W.A VBP to illustrate constant (Find Area & parameter of circle).

<u>Sol:</u> Private sub cmd Area-click()	Circle <input checked="" type="checkbox"/>
Const pi = 3.1415	Radius <input type="text"/>
Label1.caption = pi * text1.text * text1.text	<input type="text" value="Label1"/>
Private sub cmd parameter-click()	Result
Const pi = 3.1415	<input type="text" value="cmd area"/>
Label1.caption = pi * 2 * val(text1)	<input type="text" value="cmd param"/>
end sub.	

Note:- We can change the variable's data type with 'ReDim' keyword (ReDim).

* ARRAY :-

An array is a set of sequence arrange element of same type. Each element of array have a unique identify number known as Subscript or Index.

Example :-

Dim a(10) as Integer

The statement will define an array of eleven subscript from 0 to 10. To assign value to array element to use following statement.

⇒ Dim a(10) as integer

a(0) = 5

a(1) = 7

a(2) = 13

a(10) = 28

In V.B there are 2 types of Array :-

i) Fixed size array (Standard Array).

ii) Dynamic Array (Variable size Array).

Teacher Signature

i) Fixed size Array :- A fixed size array which always remains the same size as subscript. Its size is not change at run time.

ii) Dynamic Array :- The array whose size can be change at run time is known the size of array at run-time we use 'ReDim' keyword.

Ex:- Dim x(5) as integer.

In another procedure, if we need large size of array then we resize it.

ReDim x(10) as integer.

If increase the size of array x & discard its all previous value to store its previous value we use 'preserve' keyword with 'ReDim' As,

ReDim preserve x(10) as integer.

* Declaring Array ^{o_n}

Array occupies space in Memory. The programmer specifies the array type & the number of element required by the array so that the compiler may reserve the appropriate amount of Memory. Array may be declared as 'public' (in module) & Module or local.

arrays are declare in general declaration using keyword 'Dim' or 'Private'. Local array are declared in a procedure using 'Dim' or 'static'. Array must be declared explicitly with keyword 'As'

* Different Style of Array Declaration :-

- i) Dim a(5) as integer } Store 6 subscript.
- ii) Dim a(0 to 5) as integer }
- iii) Dim a(5) as integer, x(0) as string [1...]
- iv) Dim a(1 to 5) as integer } Store 5 subscript

* Multi-Dimensional Array :-

In V.B we can declare an array of Multiple dimensional which contains more than one subscript.

Example:- i) Dim a(4, 3) as integer.

Rows \swarrow \searrow Column

In the above, a is a two-dimensional array where it has 5 rows & 4 columns.

We can also declare it in other way as:-

Dim a(0 to 4, 0 to 3)

number of rows = 5

number of columns = 4

	a(0,0)	a(0,1)	a(0,2)	a(0,3)
a(1,0)		a(1,1)	a(1,2)	a(1,3)
a(2,0)		a(2,1)	a(2,2)	a(2,3)
a(3,0)		a(3,1)	a(3,2)	a(3,3)
a(4,0)		a(4,1)	a(4,2)	a(4,3)

Q) WAVBP to input the element of Array & display them (One-D & Two-D array).

For 1-D Array:-

1-D Array	Dim a(0 to 5) as Integer
Input Data - Command 1	Private Sub Command1_Click() For i = 0 to 5 a(i) = InputBox("Enter array element") Next
Display Array Content - Command 2	Print a(i) Next Private Sub Command2_Click() For i = 0 to 5 Print a(i) Next

Dim a(0 to 5) As Integer
Dim k as Integer

Private sub command 1 - click()

Static i As Integer

a(i) = Input Box ("enter array element")

i = i + 1

k = i

End sub

Private sub command 2 - click()

Dim j As Integer

Form1.Cls (clear)

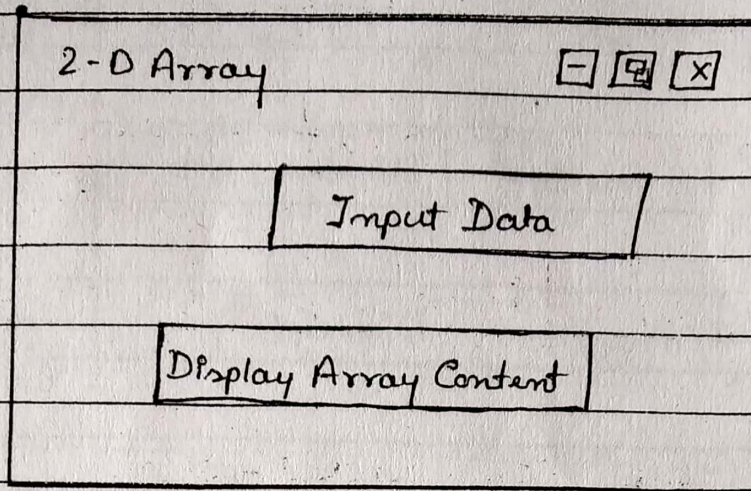
For a(j)

Next

end sub

Teacher Signature

• For 2-D Array



Dim a(2,2) as Integer

Dim(0 to 3, 0 to 2) as Integer

Private sub Command1_click()

For i = 0 to 3

Private sub Command1_click()

For j = 0 to 2

For i = 0 to 2

For j = 0 to 2

a(i,j) = InputBox("Enter array element");

Next
Next
End sub

a(i,j) = InputBox("Enter array element #...")

Next
Next
end sub

Private Sub Command2_click()

For i = 0 to 3

For j = 0 to 2

Private sub Command2_click()

for i = 0 to 2

for j = 0 to 2

Print a(i,j)

Next

Next

End sub

Print a(i,j);

Next

Print

Next

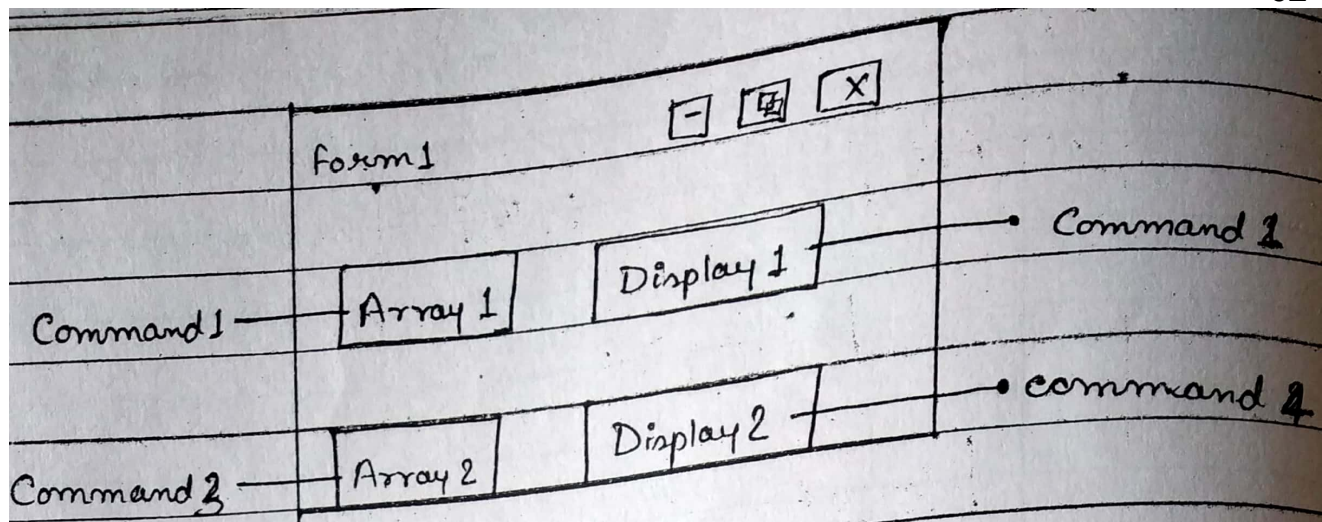
end sub

Print a(i,j)

Next

Next

Display in Row
& column form.



```
Private sub command1 - click ()
    ReDim a(5) as integer
    for i = 0 to 5
        a(i) = Input ("enter array element --- ");
    Next
end sub
```

```
Private sub command2 - click ()
    for i = 0 to 5
        Print a(i)
    Next
end sub
```

```
Private sub command3 - click ()
    ReDim a(10) as integer
    for i = 0 to 10
        a(i) = Input Box ("enter array element");
    Next
end sub
```


Private sub command 4_click()

for i = 0 to 10

Print a(i)

Next

end sub.

2) Dynamic Array :-

A dynamic array can be resized Dynamically at run time, using Re-Dim keyword.

Q). Design a VB application to show the use of Dynamic Array.

Sol.

Step 1 → Create a form along with a command button name - cmd dynamic & it's caption will be a Dynamic Array.

Step 2 → In general declaration section Declare a Array (Dynamic)
Dim da() as Integer.

Step 3 → Add the following code to ~~cmd~~ ^{cmd} dynamic-click event as given below:-


```

Private sub cmdDynamic_Click()
    ReDim da (1 to 10) as integer
    For k=L to 10
        da(k) = k+10
        Print da(k)
    Next k { k is optional }
end sub

```

Note:- In D-Array we can use 'preserve' keyword to hold its (Array) previous value (if contain).

* Control Array :-

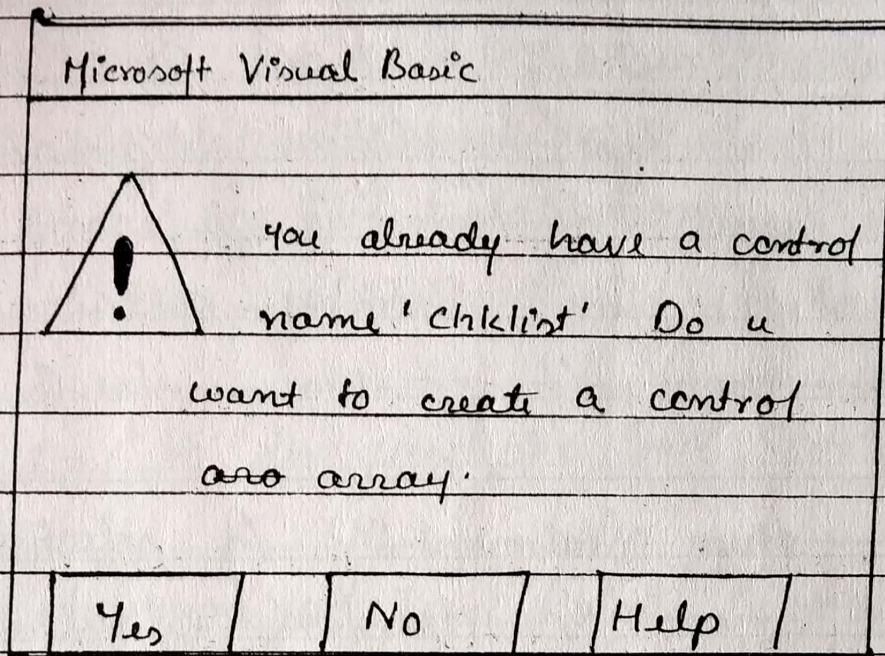
A control array is an array of controls.

For eg:-

A control of five check boxes will have same name. Each check box in array will be uniquely identified by an index value starting from 0 (zero).

To create a control array add a check box control to the form specify by name

'chklist' at the second check box control specify the same name as the first that is 'chklist', we will get a Message as given below.



Click on Yes button to create a control array.

UNIT-4

11/8/12

* Control Structure :-

Control Structure allows us to control the flow of our program structure. If we don't use control structure the program logic will flow through statements from left to right & top to bottom. Most of the power & utility of any programming language lies within decision structure & looping. The control