VISUAL BASIC PROGRAMMING

UNIT II: Branching and Looping: If...Then, If-Then-Else, Else If, Nested If, Selected-Case, For-Next, Do-Loop, While-Wend, Nested Loops, Stop Statement. VB Controls: Textbox – Checkbox – List Box –Combo Box –Label – Command Button – Directory List –Box –Drive List Box. Assigning Values to Forms and Controls – Naming Controls and Forms – Executing Commands – Displaying O/P Data –Assigning Properties Collectively Using With Block

BRANCHING AND LOOPING

Selection is choosing to execute one of several different blocks of statements, depending on the value of an expression.

Looping is a group of instructions executed repeatedly until some particular condition has been satisfied.

conditional looping is looping action will continue indefinitely until some logical condition has been satisfied.

Unconditional looping is number of passes through the loop will be known in advance.

RELATIONAL OPERATORS/COMPARISON OPERATORS

- In order to carry out branching operations in visual basic, we express conditions in equality and inequality
- These operators are used to compare numeric quantities(constants, numeric variables or numeric expressions) or strings, thus forming logical expressions that are either true or false.
- The operands within a logical expression must be of same type.(i.e both must be numeric or both must be strings)

Equal	=
Not equal	<>
Less than	<
Less than or equal to	<=
Greater than	>
Greater than or equal to	

Example

a. Numeric

X=27

C<b

Flag<>0

Profit>(a+b)

b. String

student="Radha priya"

char<>"a"

Target<city

LOGICAL OPERATOR

AND	Both the condition must be true
OR	Anyone of the condition must be true
Xor	True if one of the expression is true and other is false
Not	Used to reverse the value of a logical expression
Imp	Always result in a true condition unless the first expression is true and the second is false

Ex:
X=27 and student=“radha”
X>0 and student <=“radha”
c<d or c<a
Not(student=“radha”) and (account=“savings”)

HIERARCHY OF OPERATORS

OPERATORN	OPERATOR
Exponentiation	^
Negation(Preceding a numeric quantity with a minus sign)	-
Multiplication and division	* /
Integer Division	\
Integer Reminder	mod
Addition and Subtraction	+ -
Relationals	= <> < <= > >=
Logical Not	Not
Logical And	And
Logical Or	Or
Logical Xor	Xor
Logical Eqv	Eqv
Logical Imp	Imp

BRANCHING WITH THE if-then BLOCK

- ✓ If then block is used to execute a single statement or a block of statements on a conditional basis.
- ✓ Two forms of if-then statement
- ✓ Simplest is single-line, single statement, if-then

Syntax

if logical expression then executable statement

- ✓ The executable statement will be executed only if the logical expression is true.
- ✓ Otherwise, the statement following if-then will be executed next.

Ex:

If x<0 then x=0

Sum=sum+x

GENERAL FORM OF IF-THEN BLOCK

The general form of an if-then block:

```
If logical expression then
    executable statements
End if
```

The block of statements included between if-then and end if will be executed if the logical expression is true. Otherwise the block of statements will be bypassed and the statement following End if will be executed next.

Ex:

```
If a>b then
   print a
End if
If income <=40000 then
    tax=0.2*pay
    net=pay-tax
End if
```

BRANCHING WITH IF-THEN-ELSE BLOCKS

If-then-else block permits one of two different groups of executable statements to be executed depending on the outcome of a logical test.

Syntax

```
If logical expression then
    executable statements
Else
    executable statements
End if
```

If the logical expression is true, the first group of executable statements will be executed. Otherwise second group of executable statements will be executed.

ex:

```
If a>b then
   print a
Else
   print b
End if
```

GENERAL FORM OF if-then-else block

Syntax:

```
If logical expression1 then
    executable statements
Elseif logical expression2 then
    executable statements
Elseif logical expression 3 then
    executable statements
Repeated Elseif clauses
Else
   executable statements
End if
```

Example

```
If val(text1.text)<10 then
   msgbox “it is a single digit number”
Elseif val(text1.text)<100 then
    msgbox “it is a double digit number”
Elseif val(text1.text)<1000 then
    msgbox “it is a three digit number”
Else
     msgbox “it is a number”
Endif
```

```
If (status=“single”) then
     msgbox “not married”
Elseif (status=“double”) then
      msgbox “married”
Elseif (status=“three”) then
       msgbox(“married and have childern”)
Endif
```

SELECTION: Select Case-Syntax

```
Select Case expression
Case value1
        executable statements
Case value 2
        executable statements
……….
Case Else
       executable statements
End select
```

- ✓ When the select case structure is executed, the value of the expression is compared successively with value1, value2, etc., until a match is found.

- ✓ The group of executable statements following the matching case statement is then executed, and control is passed to the first statement following end Select. If a match cannot be found among the available values(i.e, value1, value2 etc) then the executable statements following Case Else are executed.

- ✓ The Select Case structure is particularly convenient when used in conjunction with a menu entry. In such situations the selection is based upon the menu item that is chosen.

EXAMPLE

```
‘ Tax rate based upon marital status
Dim status as String, taxrate as single
Status=Ucase(status)
Select case status
Case “single”
    taxrate=0.20
Case “married”
    taxrate=0.14
Case “retired”
    taxrate=0.12
Case else
    msgbox(“error-please try again”)
End select
```

```
Dim Flag as integer, label as string
Select case flag
Case 1,3,5
    label=“odd digit between 1 and 5”
Case 2,4,5
    label=“even digit between 2 and 6”
Case 7 to 9
    label =“any digit between 7 and 9”
Case Is>=10
    label = “Too big”
Case Else
    label=“nonpositive number”
End Select
```

LOOPINT WITH For-Next

- ✓ For-Next structure is a block of statements that is used to carry out a looping operation.
- ✓ Execute a sequence of statements some predetermined number of times
- ✓ The structure beings with a For-To statement and ends with a Next statement
- ✓ For-To statement specifies the number of passes through the loop
- ✓ Index is a variable whose value begins with value1, increases by 1 each time the loop is executed until it reaches value2.
- ✓ The next statement identifies the end of the loop. It consists of the keyword next followed by the index.

Syntax

```
For index=value1 to value2
    executable statements
Next index
```

EXAMPLE

```
Sum=0
For i=1 to 10
    sum=sum+i
Next i
```

The more general form of the for-next structure can be written as

Syntax:

```
For index =value1 to value2 Step value 3
    executable statements
Next index
```

- ✓ Within the for-to statement, value 3 determines the amount by which value1 changes from one pass to next.
- ✓ This quantity need not be restricted to an integer, and it can be either positive or negative.
- ✓ If value 1 is negative, then value1 must be greater than value 2.

EXAMPLE

Sum=0

For count=2.5 to -1 step -0.5

sum=sum + count

Next count

- Count take values 2.5,2.0,1.5…..0.0,-0.5,-1.0

Rules

- The index value can appear within a statement inside the loop but its value cannot be altered
- The value 1 and value2 are equal and value3 is nonzero, the loop will be executed once.
- The loop will not be executed at all under any of the following conditions:
  a. value1 and value2 are equal and value3 is zero
  b. value1 is greater than value2, and value 3 is positive
  c. value1 is less than value2, and value 3 is negative
- Control can be transferred out of a loop, but not in.

Example:

```
sum=0
For i=1 to 10
    sum=sum + i
    if sum>=10 then
       Exit for
Next i
```

LOOPING WITH Do-Loop

- ✓ A Do-Loop structure always begins with a Do statement and ends with a Loop statement
- ✓ 4 different ways to write a do-loop structure
- ✓ In two of the forms require that a logical expression appear in the do statement. The other two forms require that the logical expression appear in the loop statement at the end of the block.

- ✓ A Do-Loop structure always begins with a Do statement and ends with a Loop statement
- ✓ 4 different ways to write a do-loop structure
- ✓ In two of the forms require that a logical expression appear in the do statement. The other two forms require that the logical expression appear in the loop statement at the end of the block.

First form **Do while logical expression** **executable statements** **Loop**	**Second form** **Do until logical expression** **executable statements** **loop**
Third form Do executable statements Loop while logical expression	Fourth form Do executable statements Loop until logical expression

Sum=0 **Count=1** **Do while count <=10** **sum=sum + count** **count=count+1** **Loop**	**Sum=0** **Count=1** **Do until count>10** **sum=sum + count** **count=count+1** **Loop**
Sum<=10m=0 Count=1 Do sum=sum + count count=count+1 Loop while count	Sum=0 Count=1 Do sum=sum + count count=count+1 Loop until count>10

LOOPING WITH while-wend

Visual basic supports While-wend structures in addition to Do-Loop structure. This structure also permits conditional looping. The structure begins with the while statement.

Syntax

While logical expression

executable statements

Wend

- ✓ The while-wend structure continues to execute as long as the logical expression is true.

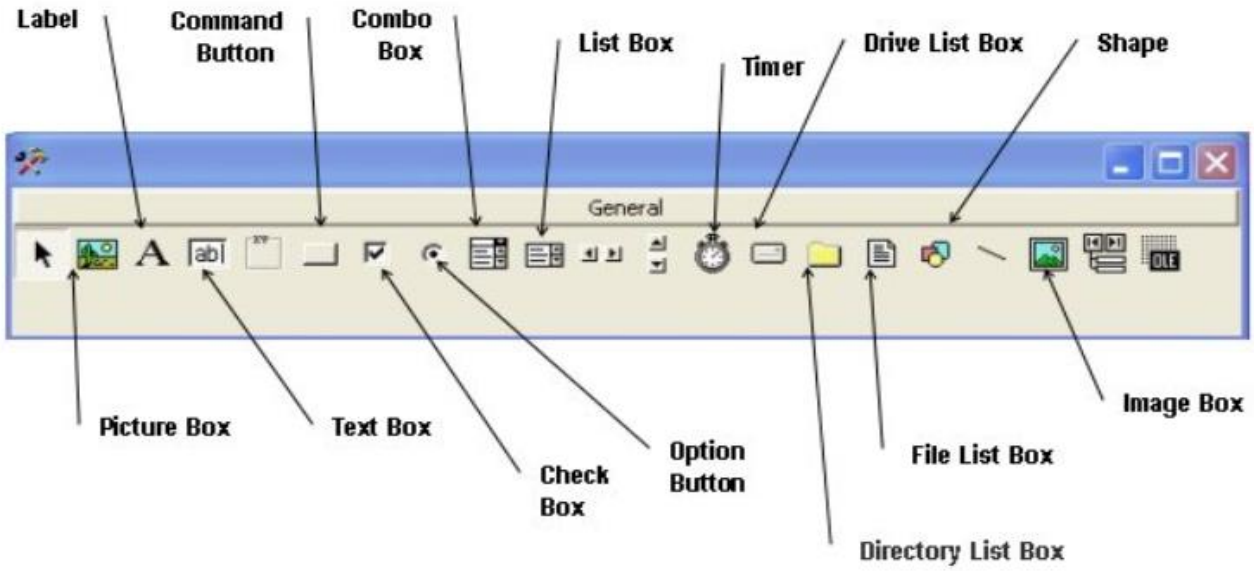
Ex:

```
Sum=0
Count=1
While count <=10
    sum=sum + count
    count=count+1
Wend
```

STOP STATEMENT

- ✓ The stop statement is used to terminate the execution at any point in the program
- ✓ This statement may appear anywhere in a visual basic program except at the end

VISUAL BASIC CONTROL FUNDAMENTALS



Check box	Provides a means of specifying a yes/no response. Within a group of check boxes any number of boxes can be checked
Combo box	Combines the capabilities of a text box and a list box
Command button	Provides a means of initiating an event action by the user clicking on the button
Data	Provides a means of displaying information from an existing database

Directory list box	Provides a means of selecting among existing drives
File list box	Provides a means of selecting files within the current directory
Frame	Provides a container for other controls. contain group of option buttons, check boxes or graphical shapes.
Horizontal scroll bar	Allows a horizontal scroll bat to be added to a control
Image box	Used to display graphical objects and to initiate event actions
Label	Used to display text on a form. The text cannot be reassigned during program execution.
Line	Used to draw a straight line segments within forms