



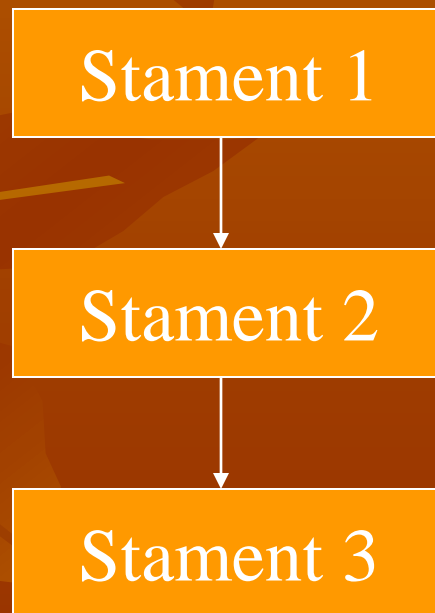
Control Structures

Control Flow

- In a program, statements may be executed sequentially, selectively or iteratively. Every programming language provides constructs to support sequence, selection or iteration. So there are three types of programming constructs :
- Sequential Constructs
- Selection Constructs
- Iterative Constructs

Sequential Construct

- The sequential construct means the statements are being executed sequentially. This represents the default flow of statements.

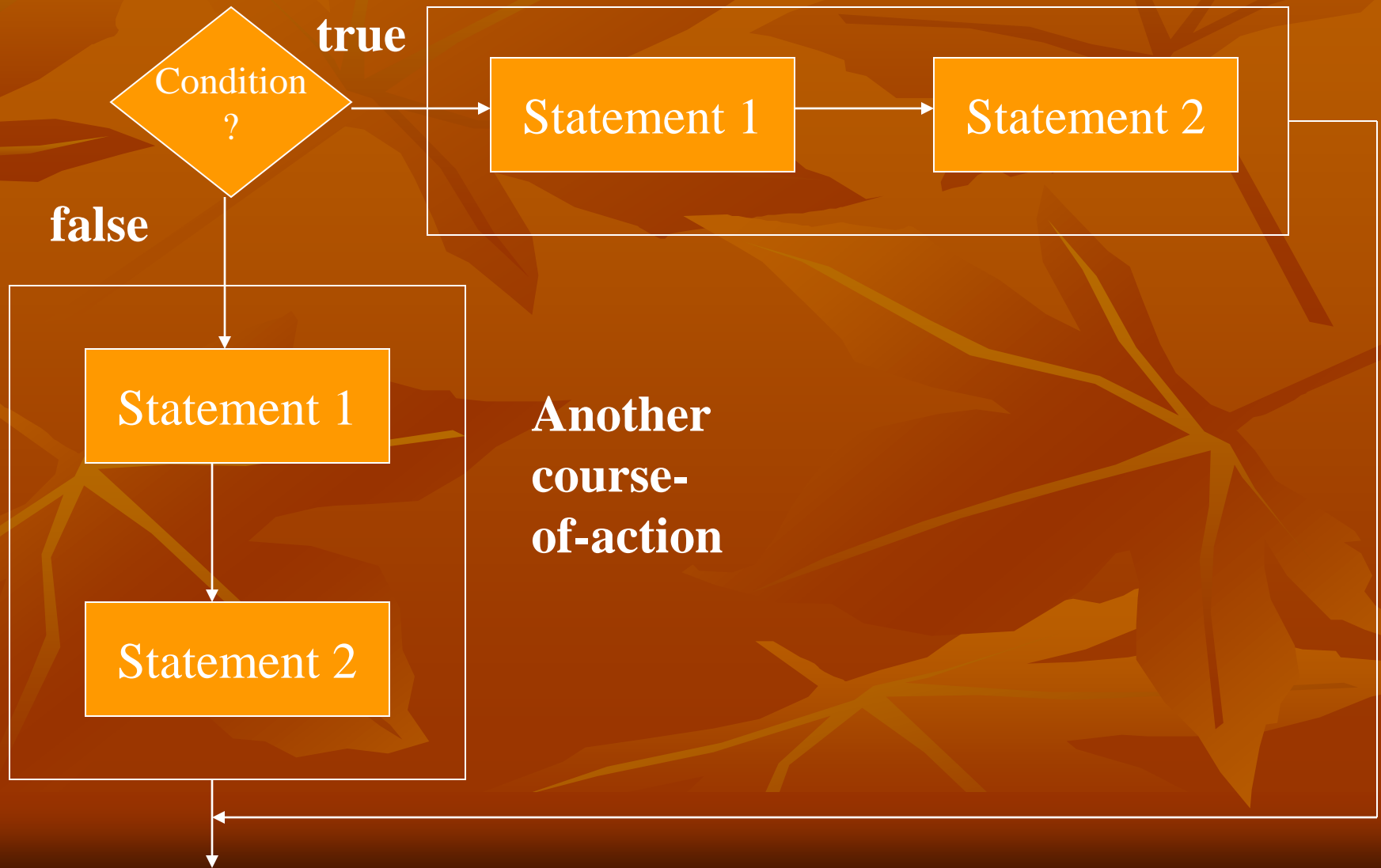


Selection Construct

- The selection construct means the execution of statement(s) depending upon the condition-test. If a condition evaluates to true, a course-of-action (a set of statements) is followed otherwise another course-of-action is followed. This construct is also called decision construct as it helps in decision making.

Selection Construct

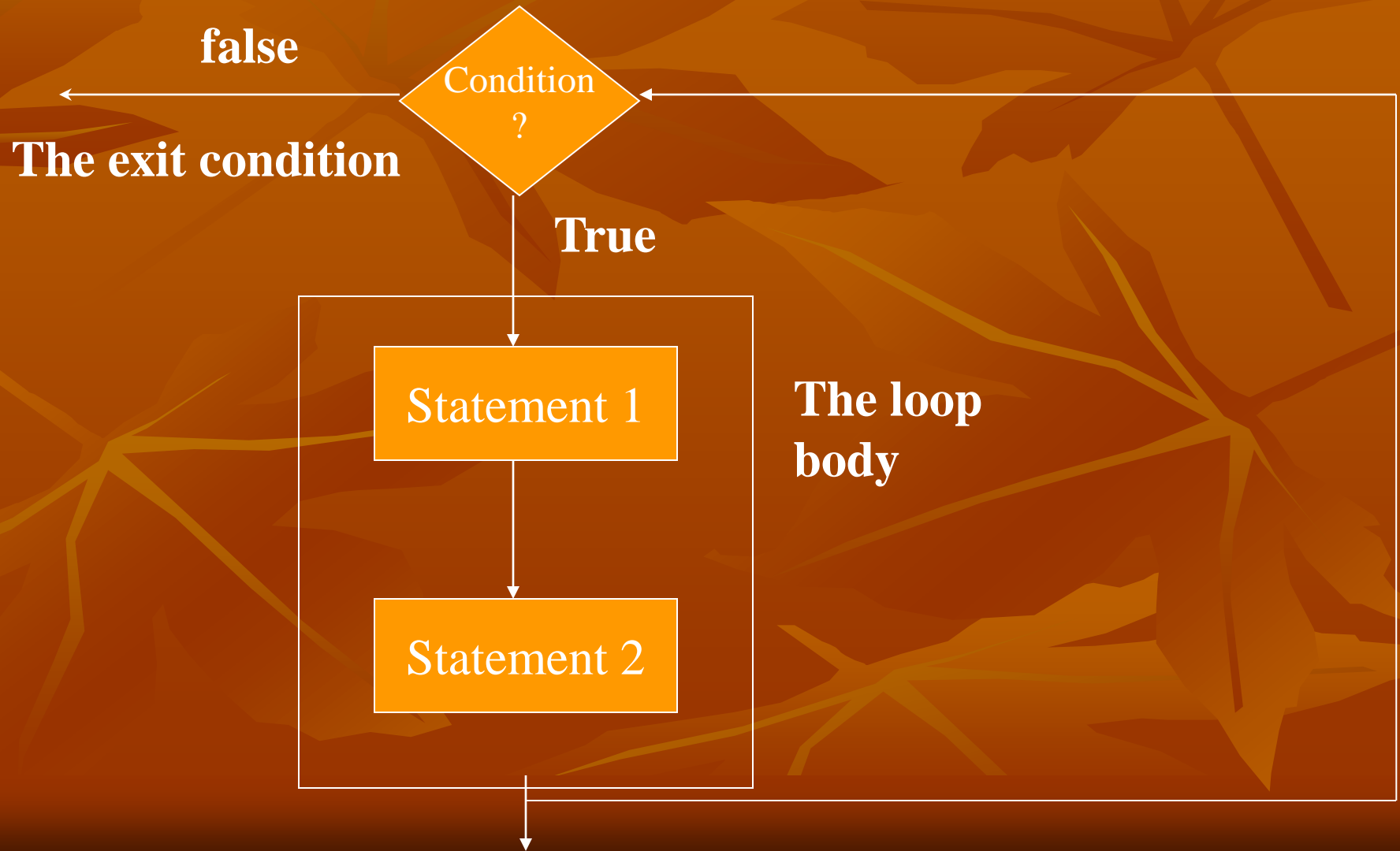
One course-of-action



Iterative Constructs

- The iterative or repetitive constructs means repetition of a set-of-statements depending upon a condition-test. A set-of-statements are repeated again and again till the condition or Boolean Expression evaluates to true. The iteration constructs are also called as looping constructs.

Iterative Construct



Selection Constructs

- VB provides two types of selection construct :
 - ❖ 1) **If** statement
 - ❖ 2) Select **Case** statement
- The If Statement : If statement of VB comes in various forms & are given below:
 - 1) **If..Then** Statement
 - 2) **If..Then..Else** Statement
 - 3) **If..Then..ElseIf** Statement
 - 4) Nested **Ifs**

If..Then Statement

- Def. : An If..Then statement tests a particular condition; if the condition evaluates to true, a course-of-action is followed otherwise it is ignored.

- Syntax :

If (boolean expression) Then
statements

End If

If..Then Statement

- Example 1. :

If (Num>0) Then

Print “It is a positive number”

End if

- Example 2 :

If txtAge.Text>=18 Then

Print “You are eligible to vote”

End if

If..Then..Else Statement

- If..Then..Else statement provides an alternate choice to the user i.e. if the condition is true then a set of statements are executed otherwise another set of statements are executed.

- Syntax :

If (boolean Expression) Then

VB Statement(s)

Else

VB Statement(s)

End If

Examples of If..Then..Else

- Example 1 :

If `txtAge.Text >= 18` Then

Print “You are eligible to vote”

Else

Print “Sorry, You are not eligible to vote”

End If

- Example 2 :

If `Num Mod 2 = 0` Then

Print “It is an Even Number”

Else

Print “It is an Odd Number”

End If

If..Then..ElseIf Statement

- If..Then..ElseIf statement is used to test a number of mutually exclusive cases and only executes one set of statements for the case that is true first.

- Syntax :

If (Boolean Expression 1) Then

Statement(s)

ElseIf (Boolean Expression 2) Then

Statement(s)

ElseIf (Boolean Expression 3) Then

Statement(s)

:

[Else

Statement(s)

End If

Example of If..Then..ElseIf

If (Age<=4) Then

Print “Your rate is free.”

ElseIf (Age<=12) Then

Print “You qualify for the children’s rate.”

ElseIf (Age<65) Then

Print “You must pay full rate”

Else

Print “You qualify for the seniors’ rate.”

End If

Nested Ifs

- A nested If is an if that has another If in its if's body or in its else's body. The nested if can have one of the following three forms :

1. **If** (expression 1) **Then**

If (expression 2) **Then**

 Statement 1

 [**Else**

 Statement 2

End If

Else

 body-of-else]

End If

Nested Ifs

2. If (expression 1) Then
body-of-if

Else

:

If (expression 2) Then
Statement-1

[Else

Statement-2]

End If

Nested If's

```
3) If (expression 1) Then
    :
    If (expression 2) Then
        Statement-1
    [Else
        Statement-2]
    End If
Else
    If (expression 3) Then
        Statement-3
    [Else
        Statement-4]
    :
    End If
End If
```

Example of Nested If's

If Num>0 Then

Print "It is a positive number"

Else

If Num<0 Then

Print "It is a negative number"

Else

Print "The number is equal to zero"

End If

End If

Select-Case Statement

- Select-Case is a multiple branching statement and is used to executed a set of statements depending upon the value of the expression. It is better to use Select-Case statement in comparison to If..Then..ElseIf Statement when the number of checks are more. There are 3 different forms of using Select-Case statements and are given below :

Different forms of Select-Case

1. Select Case : Simplest Form [Exact match]

Select Case Expression

Case Value

'one or more visual basic statements

Case Value

'one or more visual basic statements

Case Else :

'one or more visual basic statements

End Select

Example of Form 1

Select Case byMonth

Case 1,3,5,7,8,10,12

number_of_days=31

Case 2

number_of_days=28

Case 4,6,9,11

number_of_days=30

End Select

Example of Form 2

Select Case marks

Case Is < 50

Result = "Fail"

Case Is < 60

Result = "Grade B"

Case Is < 75

Result = "Grade A"

Case Else

Result = "Grade A+"

End Select

Example of Form 3

Select Case Age

Case 2 to 4 : Print “PreNursery”

Case 4 to 6 : Print “Kindergarden”

Case 6 to 10 : Print “Primary”

Case Else : Print “Others”

End Select

Home Assignment 1

- Write a program in VB to compare three no's and print the smallest number among these no's.
- Write a program in VB to check the eligibilty of a person to vote. Display a message "You are eligible to vote" if the age of the person is greater than or equal to 18 otherwise print "Sorry! You are not eligible to vote"
- Write a program to compare two no's and then print the square and cube of the larger number among these no's.

Home Assignment

- Write a program to display the grade obtained by the child according to the marks obtained by him/her. Criteria for assigning the grades is given below : If marks are
 - ≥ 90 - Grade is A
 - < 90 and ≥ 80 – Grade is B
 - < 80 and ≥ 70 – Grade is C
 - < 70 and ≥ 60 – Grade is D
 - < 60 – Grade is E

Looping Structures

- VB offers broadly following three types of looping structures :
 1. For..Next
 2. Do Loop
 - a) Do While..Loop
 - b) Do..Loop While
 - c) Do Until..Loop
 - d) Do..Loop Until
 3. While..Wend

For..Next Statement

- This type of statement is used when the user knows in advance how many times the loop is going to be executed.

- Syntax :

For <counter Variable>=<start_val> To <end_val> Step
<increment/Decrement Value>

‘ One or more VB Statements

Next <counter Variable>

Examples

❖ Example 1 : Generate natural no's from 1 to 100

```
For I = 1 To 100
```

```
    Print I
```

```
Next I
```

❖ Example 2 : Generate first 20 even no's.

```
For E = 2 to 40 Step 2
```

```
    Print E
```

```
Next E
```

More Examples

- Example 3 : Generate odd no's from 100 to 30 in a list box.

For O = 99 to 31 Step -2

ListO.AddItem(O)

Next O

- Example 4 : Generate table of any number N.

For T = 1 To N

Print N; "*", T; "=", N*T

Next T

More Examples

- Example 5 : Find factorial of a given number N.

:

Fact=1

For I= 1 to N

 Fact = Fact * I

Next I

Print “Factorial of ”; N; “=”; Fact

:

Home Assignment 2

- Write Programs for the following problems :
 - 1) To Generate all the natural no's from 200 to 500.
 - 2) To Generate the table of 10.
 - 3) To Generate all the even no's from 200 to 400 in reverse order.
 - 4) To Print first 20 multiples of the number input by the user.

Nested Loops

- A loop within another loop is called as Nested Loop.

- Example :

For I = 1 to 5

For J = 1 To 3

Print J

Inner Loop

Outer Loop

Next J

Print

Next I

Working of Nested Loops

- In nested loops the inner loop is executed completely for one time execution of the outer loop. In the Previous example the Inner Loop will be executed three times for every execution of the outer loop.
- Nested loops are very useful when there is a requirement to generate different kind of patterns as output.

Examples

- Example : Program to generate the output given below :

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

- Sol :

```
For I = 1 To 5
```

```
    For J = 1 To I
```

```
        Print I;
```

```
    Next J
```

```
Print
```

```
Next I
```

Home Assignment 3 – Nested Loops

- Write programs to generate the following patterns :

1) 1 2) 1 2 3 4 5
1 2 1 2 3 4
1 2 3 1 2 3
1 2 3 4 1 2
1 2 3 4 5 1

3) A
A B
A B C
A B C D

Home Assignment 4 – Output Related Questions

- Specify the output of the following output related questions, assume that variables are declared :

```
1) sum=0
For I = 1 To 4
    For J = 1 To 3
        sum = sum + J
    Next J
Print "Sum = ";sum
Next I
```

```
2) sum = 0
p=2 : q=4
Do While p<=10
    If p Mod 2 = 0 Then
        sum = sum + p
    End If
    p=p+1
Loop
Print sum
```

Output related questions

3) R=5 : S=3

Do

Print R, S

R = R + 1

If R = 7 Then

Exit Do

End If

S = S + R

Loop While R <=10

4) M=2 : N=4

Do Until M>12

N = N + M

If M Mod 3 = 0 Then

N = N - M

Else

N = N + M

End If

M=M+1

Loop

Print M, N

Home Assignment - 5

- Programming Questions : Develop programs for the followings :
- To generate the mirror image of the number entered by the user.
- To generate the prime numbers between 100 and 700.
- To Check whether the number entered by the user is a prime number or not.
- To print first 30 multiples of the number entered by the user.
- To generate the factorial of the number input by the user.

Home Assignment - 5

- To generate the Fibonacci series upto first 15 terms
i.e. 0 1 1 2 3 5 8.....
- To find the sum of all the numbers divisible by 7
from 100 to 200.
- To check whether the year entered by the user is leap
year or not.
- To find and print the sum of digits of the number
entered by the user.
- To find and print the product of digits of the number
accepted from the user.