➢ Decision Statements

Decision making structures require that the programmer specify one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

VB.Net provides the following types of decision making statements. Click the following links to check their details.

Statement	Description
If ... Then	**If...Then statement** consists of a boolean expression followed by one or more statements.
If...Then...Else	An **If...Then statement** can be followed by an optional **Else statement**, which executes when the boolean expression is false.
nested If	You can use one **If** or **Else if** statement inside another **If** or **Else if** statement(s).
Select Case	A **Select Case** statement allows a variable to be tested for equality against a list of values.

1. If … Then Statement

It is the simplest form of control statement, frequently used in decision making and changing the control flow of the program execution. Syntax for if-then statement is –

```
If condition Then
    [Statement(s)]
End If
```

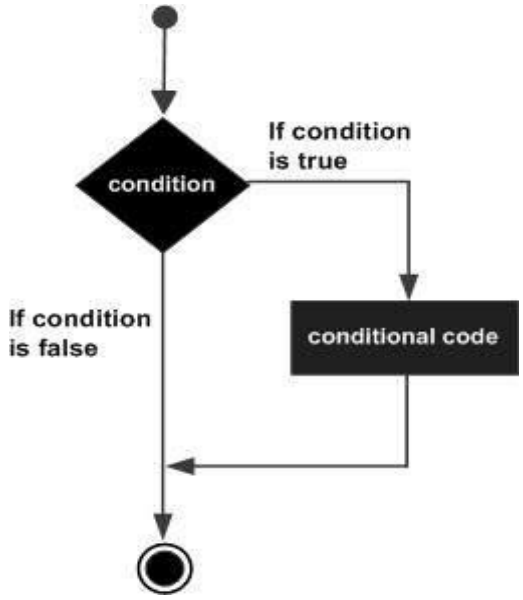
Where,

Condition is a Boolean or relational condition and *Statement*(s) is a simple or compound statement.

Example :

```
If (a <= 20) Then
    c = c+1
End If
```

If the condition evaluates to true, then the block of code inside the If statement will be executed. If condition evaluates to false, then the first set of code after the end of the If statement (after the closing End If) will be executed.

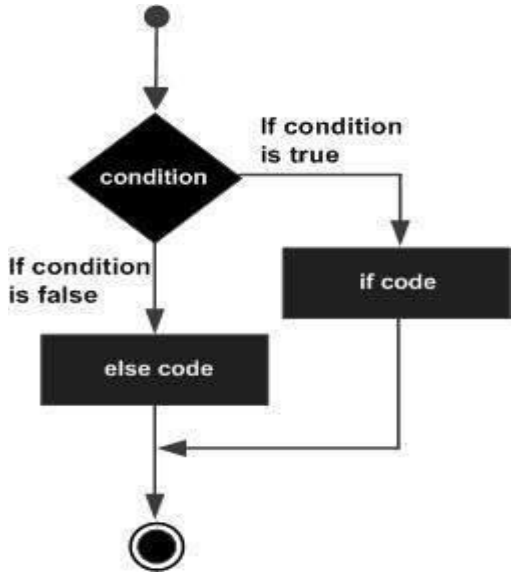


2. If … Then … Else Statement

An **If** statement can be followed by an optional **Else** statement, which executes when the Boolean expression is false. Syntax for if-then-else statement is –

```
If (boolean_ expression) Then
    Statement (s) will execute if then boolean expression is true
Else
    Statement (s) will execute if then boolean expression is false
End If
```

If the Boolean expression evaluates to **true**, then the If block of code will be executed, otherwise Else block of code will be executed.



3. Nested IF

It is always legal in VB.Net to nest If-Then-Else statements, which means you can use one If or ElseIf statement inside another If ElseIf statement(s). The syntax for a nested If statement is as follows −

```
If ( boolean_expression 1)Then
    'Executes when the boolean expression 1 is true
   If ( boolean_expression 2)Then
       'Executes when the boolean expression 2 is true
   End If
End If
```

*** You can nest ElseIf...Else in the similar way as you have nested If statement.

Example: The following code, tests the score which is input by the user and display appropriate message according to the score.

```
Private Sub btnTestGrade_ Click( )
    Dim score As Single
    Dim result As String
    score = InputBox ("Enter Score .. ?")
    If score < 50 Then
       result = "Failed"
    Else if score < 70 then
       result = "Pass"
     Else if score < 80 then
       result = "Good"
     Else if score < 90 then
       result = "V.Good"
```

```
        Else
           result = "Excellent"
        End If
        MsgBox (result)
    End Sub
```

4. Select Case Statement

A **Select Case** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each select case. The syntax for a Select Case statement is as follows –

```
Select [ Case ] expression
    [ Case expressionlist
            [ statements ] ]
    [ Case Else
            [ elsestatements ] ]
End Select
```

Where,

- ***expression*** − is an expression that must evaluate to any of the elementary data type in VB.Net, i.e., Boolean, Byte, Char, Date, Double, Decimal, Integer, Long, Short, Single, String.
- ***expressionlist*** − List of expression clauses representing match values for *expression*. Multiple expression clauses are separated by commas.
- ***statements*** − statements following Case that run if the select expression matches any clause in *expressionlist*.
- ***elsestatements*** − statements following Case Else that run if the select expression does not match any clause in the *expressionlist* of any of the Case statements.

H.W.: Rewrite the above segment of code related to If statement using Select statement.

5. IIf ( ) Function

It is a built in function accepts as an argument an expression and two values, evaluates the expression, and returns the first value if the expression is **true** or the second value if the expression is **false**. The IIf has the following syntax:

```
Iff (expression, True Part, False Part)
```

Iff function is compact notation for simple If statement, and you can use it to shorten If .. Then .. Else statement.

Example: Let's say you want to display one of the strings "CLOSE" or "FAR" depending on the value of distance variable. Instead of a multiple if statement, you can call IIf ( ) Function as follows.

```
Private Sub Button1_Click( )
        Dim distance As Single
        Dim result As String
        distance = InputBox( "Enter Distance   ?")
        result = IIf ( distance >1000, "FAR", "CLOSE")
        MsgBox (result)
End Sub
```

H.W: Read statement grade using an InputBox, test the grade then print "Pass" of "Fail" by using a message box.

Note: Tools that used with selection statements are CheckBox and RadioButtons.

H.W: Design and write a program to read length and width in Centimeter then convert them meter or inch or both. Use appropriate controls in your design.