**Sub Procedure**

A complete instruction with a meaning is called a statement. In VB.Net, a group of statements which together performs a task when it is called is known as a procedure. Similarly, a sub procedure in VB.Net is a group statements enclosed by the Sub and End Sub statements.

Sub Procedures returns control to the calling code after performing the task. Sub procedure do not return a value to the calling code. Every Visual Basic program must consist of a sub procedure called Main. Every programs in VB starts within the main procedure and end within the main().

Programmer can create new sub procedures in programs which should be called within the main procedure to work.

**Example Program**

Module Module1

Sub main ( )

custom\_sub\_procedure ( )

End Sub

Sub custom\_sub\_procedure ( )

console.WriteLine("Welcome to sub procedures")

End Sub

End Module

In the above example, the programs starts and ends within the main ( ) procedure, the custom\_sub\_procedure ( ) will be called within the main sub procedure. Here, after executing the custom sub procedure, the control is transferred back to the main sub procedure.

The programmer can pass data through the parenthesis of sub procedures, these data are called arguments.

**Example Program**

Module Module1

Sub Main ( )

----

End Sub

Sub diaply\_msg (ByVal strText As String)

----

End Sub

End Module

There are two types of argument passing they described below.

ByVal : Here the arguments passed by value. Which means the actual variable does not passes, but a copy of the variable is passed through the parenthesis. Any changes made to the variable does not affect the actual variable.

ByRef : In this case, the argument will be passed by reference. Here the location of the actual variable is passed, which means there is a direct access to the variable and so, any changes that made to the variable will be permanent.

**Functions in Visual Basic .Net**

In Visual Basic .Net, any sub procedure which returns a value to the calling code is known as a function.

The functions are used in VB using a keyword Function.

Every function in Visual Basic .Net are sub procedure whereas every sub procedures are not functions.

Syntax

[modifier] Function function\_name ( [parameter list]) As return type

[statements]

End Function

Example

Public Function Sum (int a, int b) As integer

s = a + b

return s

End Function

**Example Program**

Module Module1

Sub Main()

Dim a As Integer = 5

Dim b As Integer = 10

Dim max As Integer = Findmax(a,b)

End Sub

Function Findmax(ByVal num1 As Integer, ByVal num2 As Integer) As Integer

Dim Result As Integer

If (num1>num2) Then

Result = num1

Else

Result = num2

End If

Return Result 'returning the result value

End Function

End Module

There are two methods to return a value from the function.

By using the return statement.

By assigning the value to the function name.

In the above example we used the first method, and if we use the second methods the statement Return Result is replaced with FindMax = Result.

**Recursive Function**

A recursive Function is one that calls itself.

Module Module1

Function Factorial (ByVal num As Integer)

Dim result As Integer

If num = 1 Then

Return 1

Else

result = Factorial(num - 1) \* num

Return result

End If

End Function

Sub Main()

Console.WriteLine("Factorial = {0} ", Factorial(6))

Console.ReadKey()

End Sub

End Module