Chapter - 6
Procedures, Functions and Modules

Procedure:
It's a named unit of a group of program statement that performs a well defined task. This unit can be called from the calling program.

Role of Procedure:
Increase Reusability – Using same code again and again.
Modularization – Means to divide a Big program into small modules.

Types of General Procedure:

1) Sub - A procedure performs a task and does not return a value.

```
Private/Public Sub Procedure-Name( parameter list)
    Statements
End Sub
```

Eg: Write a VB Procedure that receives a number and check that it is even or odd.
Called Procedure:-

```
Sub checkEvenOdd (x as Integer)
    If x mod 2 = 0 then
        Print "Even No."
    Else
        Print "Odd No."
    End if
End Sub
```

Calling Program:-

```
Private sub command1_click( )
    Dim  A as Integer
    A = val(Inputbox("Enter a No"))
    Call checkEvenOdd (A)
End Sub
```

2) Function - A procedure performs a task and returns a value.

```
Private/Public Function Function-Name( parameter list)
    Statements
End Function
```

Eg: Write a VB Function that receives a number and Returns True if it is even otherwise False.

Called Function:

```
Function checkEvenOdd (x as Integer) As Boolean
    If x mod 2 = 0 then
        checkEvenOdd = TRUE
    Else
        checkEvenOdd = FALSE
    End if
End Function
```

Calling Program:

```
Private sub command1_click( )
        Dim x as Boolean
        Dim  A as Integer
        A = val (Inputbox (“Enter a No”) )
         x = checkEvenOdd (A)
        If x = TRUE then
                Msgbox “Even No.”
        Else
                Msgbox “Odd No.”
        End If
End Sub
```

Call byVal and Call byRef:-

The call byVal method copies the values of actual parameter into the formal parameters, ie. The procedure creates its own copy of argument values and then uses them. Only a copy of a variable is passed to the called procedure and if the procedure changes the value, the changes affects only the copy and not reflected back to the original variable itself.

```
Sub DemoByValue( ByVal x as Intyege )
        x = x + 10
        Print x
End Sub

Private sub mainprg( )
Dim A as Integer
A = 10
Print A
Call DemoByValue( A)
Print A
End Sub
```

The O/P is:

10
20
10

The call byRef method does not creates its own copy of original values, rather it refers to the original values only by different names called reference, and thus the called procedure works with the original data and any changes in the values gets reflected to the data.

```
Sub DemoByReference ( ByRef x as Integer )
        x = x + 10
        Print x
End Sub
```

```
Private sub mainprg( )
        Dim A as Integer
        A = 10
        Print A
        Call DemoByReference ( A)
        Print A
End Sub
```

The O/P is:

10
20
20

Code Modules:-

A module is a container in VB, that contain some variables, procedures and definition.

Three types of Modules are:

1) Form Module - Stores all the procedures and declarations pertaining to single form. *Form modules are stored with **.FRM** extension.*

2) Standard Module - Store general purpose code of the application, ie code and declaration that are not specific to a form. *Standard modules are stored with **.BAS** extension.*

3) Class Module - It stores the blueprint for user created custom object. *Class modules are stored with **.CLS** extension.*

Variable Scope: -

The part of a program within which a variable is accessible, is known as its scope.

Three Variable Scopes are:

1) Private Scope/ Local Scope - Variables declared within a procedure are in local scope.

2) Module Scope - Variables available for all the procedures within that module.

3) Public Scope / Global Scope - Variables declared with public statement and available to the application.

Static Variables: Local and Static variable differ by their life span. Static variables retain its value even after the procedure has finished executing.

Which variable is static in the following function?

```
Static function MyFunction ( )
        Static X1 as Integer
        Dim Y1 as Integer
End Function
```

***Ans: -** Both X1 and Y1 are static variables because a Static function makes all its local variables Static.*

*** * ***