General Procedures in Visual Basic .NET

There are two types of general procedures in visual basic. One is called a “Sub” procedure; which is primarily used to perform an action, and the other is called a “Function” procedure; which is used to perform a calculation and pass an item back.

A “Function” procedure should be used when the procedure is needed to calculate a single item only and pass this information back. A “Sub” procedure will be used when the procedure needs to pass back more than one item of information or when it will perform an action where nothing needs to be passed back.

Both procedures may or may not have parameters. Parameters are used to pass information to procedure and to pass information back from procedures. Parameters tell compiler the following:

1) The number of parameters procedure has.
2) The data type of each parameter.
3) Their position in the parameter list.
4) What kind of parameter are they, ie. Value (input) parameters or Reference (output) parameters.

Sub Procedures – syntax, declaring sub that has only two parameters in Formal Parameter List

```vbnet
Sub ProcedureName(ByVal Parameter1 As DataType, ByRef Parameter2 As DataType)
    procedure statements
End Sub
```

Sub Procedures – syntax, calling sub that has only two parameters in Actual Parameter List

```vbnet
ProcedureName(Parameter1, Parameter2)
```

Function Procedure – syntax, declaring function that has only one parameter in Formal Parameter List

```vbnet
Function FunctionName(ByVal Parameter As DataType) As DataType
    Function statements
End Function
```

Function Procedure – syntax, calling function that has only one parameter in Actual Parameter List

```vbnet
Variable = FunctionName(Parameter)
```

Parameter List Correspondence Rules between Formal and Actual Parameters.

1) Actual parameter list must be the same size and order as the formal parameter list.
2) Actual parameter that corresponds to a VALUE formal parameter:
   - may be a variable, expression, or constant
3) Actual parameter that corresponds to a REFERENCE formal parameter
   - should be of the same data type as the formal parameter
   - must be a variable itself, not an expression or constant