Kavita K. Bharti
Department of Computer Science
Durga Mahavidyalaya, Raipur (C.G)

VB.Net-Classes & Objects

When you define a class, you define a blueprint for a data type. This doesn't actually define any data, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object.

Objects are instances of a class. The methods and variables that constitute a class are called members of the class.

Class Definition

A class definition starts with the keyword **Class** followed by the class name; and the class body, ended by the End Class statement. Following is the general form of a class definition –

```
[ <attributelist> ] [ accessmodifier ] [ Shadows ] [ MustInherit |
NotInheritable ] [ Partial ] _
Class name [ ( Of typelist ) ]
   [ Inherits classname ]
   [ Implements interfacenames ]
   [ statements ]
End Class
```

Where,

- ***attribute list***is a list of attributes that apply to the class.Optional.
- ***accessmodifier***defines the access levels of the class, it has values as - Public, Protected,Friend, Protected Friend and Private. Optional.
- ***Shadows*** indicate that the variable re-declares and hides an identically named element, or set of overloaded elements, in a base class. Optional.
- ***MustInherit*** specifies that the class can be used only as a base class and that you cannot create an object directly from it, i.e., an abstract class. Optional.
- ***NotInheritable***specifiesthattheclasscannotbeusedasabaseclass.
- ***Partial***indicatesapartialdefinitionoftheclass.
- ***Inherits***specifiesthebaseclassitisinheritingfrom.
- ***Implements***specifiestheinterfacestheclassisinheritingfrom.

The following example demonstrates a Box class, with three data members, length, breadth andheight −

```
Module mybox
   Class Box
      Public length As Double   ' Length of a box
      Public breadth As Double  ' Breadth of a box
      Public height As Double   ' Height of a box
   End Class
   Sub Main()
      Dim Box1 As Box = New Box()        ' Declare Box1 of type Box
      Dim Box2 As Box = New Box()        ' Declare Box2 of type Box
      Dim volume As Double = 0.0         ' Store the volume of a box here

      ' box 1 specification
      Box1.height = 5.0
      Box1.length = 6.0
      Box1.breadth = 7.0

      ' box 2 specification
      Box2.height = 10.0
      Box2.length = 12.0
      Box2.breadth = 13.0

      'volume of box 1
      volume = Box1.height * Box1.length * Box1.breadth
      Console.WriteLine("Volume of Box1 : {0}", volume)

      'volume of box 2
      volume = Box2.height * Box2.length * Box2.breadth
      Console.WriteLine("Volume of Box2 : {0}", volume)
      Console.ReadKey()
   End Sub
End Module
```

Volume of Box1 : 210
Volume of Box2 : 1560

Member Functions and Encapsulation

A member function of a class is a function that has its definition or its prototype within the class definition like any other variable. It operate so many object of the class of which it is a member and has access to all the members of a class for that object.

Member variables are attributes of an object (from design perspective) and they are kept private to implement encapsulation. These variables can only be accessed using the public member functions. Let us put above concepts to set and get the value of different class members in a class−

```
Module mybox
   Class Box
      Public length As Double    ' Length of a box
      Public breadth As Double   ' Breadth of a box
      Public height As Double    ' Height of a box
      Public Sub setLength(ByVal len As Double)
         length = len
      End Sub

      Public Sub setBreadth(ByVal bre As Double)
         breadth = bre
      End Sub

      Public Sub setHeight(ByVal hei As Double)
         height = hei
      End Sub
```

```
    Public Function getVolume() As Double
       Return length * breadth * height
    End Function
 End Class
 Sub Main()
    Dim Box1 As Box = New Box()         ' Declare Box1 of type Box
    Dim Box2 As Box = New Box()         ' Declare Box2 of type Box
    Dim volume As Double = 0.0          ' Store the volume of a box here

    ' box 1 specification
    Box1.setLength(6.0)
    Box1.setBreadth(7.0)
    Box1.setHeight(5.0)
```

```
      'volume of box 2
      volume = Box2.getVolume()
      Console.WriteLine("Volume of Box2 : {0}", volume)
      Console.ReadKey()
   End Sub
End Module
```

When the above code is compiled and executed, it produces the following result −

VolumeofBox1 :210
VolumeofBox2:156