LAB GOALS

To learn how to create new classes. Encapsulation, Public vs. Private, Inheritance.

Open up Microsoft Visual Basic .NET.

Step 1: Create a new Console project named “User_Defined_Classes”. Don’t forget..

Option Explicit On
Option Strict On

Step 2: Creating a new Class called CAR:

Right below the Option Strict line and above the Module line, enter the following class definition. This user defined class will create a new data type called CAR. Once the class is created, we may start to use this data type in our program. The class below demonstrates how one can encapsulate (protect) the data items such as model, make, year and color by declaring them as PRIVATE and only allow access to them via some PUBLIC methods.

```
' Use of Private and Public Attributes and Methods.
'--------------------------------------------------------------
Public Class CAR
' Private Data
Private make As String
Private model As String
Private year As Integer
Private color As String

'-------------------------------
' Public Methods
Public Sub PrintColor() 'return the color attribute
Console.WriteLine("The Color = {0}", color)
End Sub

'-------------------------------
Public Sub SetColor(ByVal clr As String)
color = clr
End Sub
End Class

Step3: Using the CAR class in our program:

Module Module1
Sub Main()
Dim MyCar As CAR ' MyCar is a Reference Variable
MyCar = New CAR ' The NEW operator will actually allocate the memory needed for the car object.

MyCar.SetColor("BLUE")
MyCar.PrintColor()

'Console.WriteLine(MyCar.color) ' Note that this is a error, since we don't have direct access to the Color attribute.
End Sub
End Module

Step4: Refining the CAR class by adding additional functionality to it:
Add the following procedures and functions to the CAR class.

Public Function GetColor() As String 'return the color attribute
Public Sub SetMake()
Public Sub SetModel()
Public Sub SetYear()
Step 5: Inheritance:

Create a new Class called SUPER_CAR, by first inheriting the existing class CAR and extending its property and functionality.

```
Class SUPER_CAR
    Inherits CAR               ' Inherits from the CAR class
    Public SuperAttribute As String
    '-------------------------------
    Public Function GetSuperAttribute() As String
        Return SuperAttribute
    End Function
    '-------------------------------
    Public Sub SetSuperAttribute(ByVal SuperAttr As String)
        SuperAttribute = SuperAttr
    End Sub
End Class
```

```
Module Module1
    Sub Main()
        Dim MyCar As CAR       ' MyCar is a Reference Variable
        MyCar = New CAR       ' The NEW operator will actually allocate the memory needed for the car object.
        MyCar.SetColor("BLUE")
        MyCar.PrintColor()
        ' Console.WriteLine(MyCar.color)  ' Note that this is a error, since we don't have direct access to the Color attribute.
        '-------------------------------
        Dim BatMobil As New SUPER_CAR
        BatMobil.SetColor("BLACK")
        BatMobil.SetSuperAttribute("Goes Real Fast!!")
        BatMobil.PrintColor()
        Console.WriteLine(BatMobil.SuperAttribute)   ' Note that I can access SuperAttribute because it is public.
        '-------------------------------
        Console.ReadLine()
    End Sub
End Module
```