Implementing OOP concepts in C#

1. Write a program to display the name and age of a person. Use a default constructor to assign values to the name and age variables. Use a parameterized constructor to pass the values of name and age. Use a single method to display the values from both the constructors.

Solution:

```
using System;

class Employee
{
   private string name;
   private int age;

   // Default Constructor
   public Employee()
   {
      name="Mark";
      age=25;
   }

   // Parameterized Constructor
   public Employee (string varName, int varAge)
   {
      name = varName;
      age = varAge;
   }

   public void ShowData()
   {
      Console.WriteLine("Name = " + name);
      Console.WriteLine("Age = " + age);
   }

   static void Main()
   {
      Employee objEmpOne = new Employee ();
      Employee objEmpTwo = new Employee ("Allen",30);
      objEmpOne.ShowData();
      Console.WriteLine();
      objEmpTwo.ShowData();
   }
}
```

2. Write a program that calculates the square of an integer, say 3 and a double, say 4.2. Use method overloading to calculate the square of the integer and double values.

Solution:

```
using System;
class Maths
{
   public void DoOverLoad()
   {
      int intX = 3;
      double dblY = 4.2;
      Console.WriteLine("Square of int value is : "+Square(intX)
+ "\n" +  "Square of double value is : " + Square(dblY));
   }

   public int Square(int intY)
   {
      return intY*intY;
   }

   public double Square(double dblY)
   {
      return dblY*dblY;
   }
}

public class OverLoad
{
   public static void Main()
   {
      Maths objMaths = new Maths();
      objMaths.DoOverLoad();
   }
}
```

3. Write a program to demonstrate Inheritance. Define a base class **`Vehicle`** having properties like type, color, speed, brand and methods **`Run()`** and **`Display()`**. The **`Run()`** method should display a message "I am running" and the type of the vehicle. The **`Display()`** method should display the various properties of the vehicle. Derive a class **`Car`** and initialized the derived attributes of base class **`Vehicle`**. Finally, in the `Main()` method execute **`Run()`** and **`Display()`** using an object of the derived class **`Car`**.

Solution:

```
using System;

class Vehicle
{
   public string strType;
   public string strColor;
   public double dblSpeed;
   public string strBrand;
```

```
   public void Run()
   {
      Console.WriteLine(strType + "   : I am running");
   }
   public void Display()
   {
      Console.WriteLine("Type  : " + strType);
      Console.WriteLine("Color : " + strColor);
      Console.WriteLine("Speed : " + dblSpeed);
      Console.WriteLine("Brand : " + strBrand);
   }
}

class Car : Vehicle
{
   public Car(Vehicle objVehicle)
   {
      strType = objVehicle.strType;
      strColor = objVehicle.strColor;
      dblSpeed = objVehicle.dblSpeed;
      strBrand = objVehicle.strBrand;
   }
}

class Inherit
{
   static void Main()
   {
      Vehicle objVehicle = new Vehicle();
      objVehicle.strType = "Car";
      objVehicle.strColor = "Red";
      objVehicle.dblSpeed = 100.2;
      objVehicle.strBrand = "BMW";
      Car objCar = new Car(objVehicle);
      objCar.Run();
      objCar.Display();
   }
}
```

4. Consider the previous question. Override the **`Run()`** method in the derived class to display a message "The CAR is running".

Solution:

```
using System;

class Vehicle
{
   public string strType;
   public string strColor;
```

```
   public double dblSpeed;
   public string strBrand;

   public virtual void Run()
   {
      Console.WriteLine(strType + "   : I am running");
   }
   public void Display()
   {
      Console.WriteLine("Type  : " + strType);
      Console.WriteLine("Color : " + strColor);
      Console.WriteLine("Speed : " + dblSpeed);
      Console.WriteLine("Brand : " + strBrand);
   }
}

class Car : Vehicle
{
   public Car(Vehicle objVehicle)
   {
      strType = objVehicle.strType;
      strColor = objVehicle.strColor;
      dblSpeed = objVehicle.dblSpeed;
      strBrand = objVehicle.strBrand;
   }
   public override void Run()
   {
      Console.WriteLine("The CAR is running");
   }
}

class Inherit
{
   static void Main()
   {
      Vehicle objVehicle = new Vehicle();
      objVehicle.strType = "Car";
      objVehicle.strColor = "Red";
      objVehicle.dblSpeed = 100.2;
      objVehicle.strBrand = "BMW";
      Car objCar = new Car(objVehicle);
      objCar.Run();
      objCar.Display();
   }
}
```