**Assignment 1.**

**Objectives:**

1. To understand object oriented concepts using c#.
2. Please follow naming conventions and best practices as discussed in the class.

**Problem Description:**

**Part 1:**

1. Create an abstract class Quadrilateral which has a data member length & abstract method area.
2. Create two derived classes Square and Rectangle.
3. Square class shall have no data member. Give implementation of area method.
4. Rectangle class shall have data member called Breadth. Give implementation of area method.
5. In the main program create Rectangle object and Square object.
6. Make use of only constructors to cm reate the objects & initializing length of Square to 4 & that of rectangle set length to 5 & Breadth to 6.
7. Compare the two areas & display appropriate message.

(E.g.: Square area is greater than Rectangle area)

**Part 2:**

1. Create a property **IsSquare** in rectangle class which returns a true when the given Rectangle is a Square.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Assignment 2**

**Module: C#**

**Objective: To understand Multilevel inheritance in C#**

**CREATE A CONSOLE APPLICATION GAMEINHERITENCE**

**CLASS GAME**

Create a class Game - Base Class

with Attributes no.of players & country

Create Public Properties of Attributes

Create Two Constructors

1) Default constructors

2) Parameterized Constructor with all attributes

create a method - WorldCup() which will show

output as 'World Cup' + Current Year

(Use DateTime.Now Properties)

**CLASS CRICKET**

Create a class Cricket - which will be derive from Base Class Game

Add Attributes - coachname - string

Create Public Properties of Attributes

Create Two Constructors

1) Default Constructors

2) Parameterized Constructor with noofplayers,country,coachname

\*(Make use of base keyword to call the attributes of Base Class)

**Class FootBall**

create a class FootBall - which will be derive from Base Class Game

Add Attributes - Leaguename - string

Create Public Properties of Attributes

Create Two Constructors

1) Default Constructors

2) Parameterized Constructor with noofplayers,country,LeagueName

\*(Make use of base keyword to call the attributes of Base Class)

Class ShowGameDetails - which is not a Derive Class

Create two Methods:

1) ShowCricketDetails - which will show All information about Cricket

2) ShowFootBallDetails - Which will show All information about Football

**Class Program**

Create Two Objects of Cricket as India & Australia Respectively

Create Two Objects of FootBall as Spain & England Respctively.

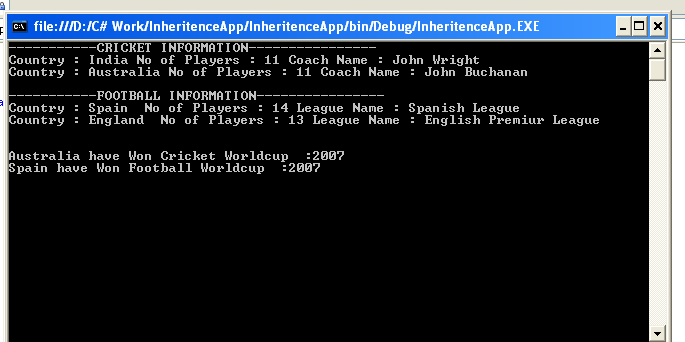
Call the methods of Class ShowGameDetails.

Create an Object of Game Class.

Call the Method Worldcup with Object Australia for Cricket

Call the Method Worldcup with Object Spain for Football

Show the Output as Below.



**ASSIGNMENT 3**

Create an enumeration of VehType with values as Car, Truck, Bus

Create an enumeration FuelType as Diesel and Petrol and CNG

Create a class library with class name Vehicle in namespace Sample

Data Members of the Vehicle class are as below

name string

color string

vehType // use enum VehType declared above

NoOfWheels int

fuelType // use enum

make string

Methods of Vehicle class are

Start()

display Car has started or Truck has started

make use of VehType to display Car or Truck

Stop()

display Car has stopped or Truck has stopped

make use of VehType to display Car or Truck in the string

Constructors

1.params: all members need to be declared here

2.params - pass only name, color, VehType, NoOfWheels and use in the code block default values to set other parameters

default values:

FuelType as Petrol

Do not use this here to call any other constructor

3.params - pass only name and color

make use of this to call No. 1 constructor

pass default values thru this() :

VehType as Car

NoOfWheels 4

FuelType as Petrol

Declare properties for each (including property VehicleType for enum VehType) and every datamember

Make use VehicleType property inside the methods Start() and Stop() to display the vehicle (eg. Car should be displayed when VehType.Car is set as VehType for the object)

Use the class library in Console Application called VehicleApp

Make use of proper naming conventions

In VehicleApp create the following objects

Vehicle car = new Vehicle("Cielo","red");

Vehicle truck = new Vehicle("Tata","blue",VehType.Truck,6,FuelType.Diesel);

Vehicle smallcar = new Vehicle("Indica","silver",VehTyep.Car,4);

See that all constructors are made use of in this program.

Then call methods of all the objects.

Then write Console.WriteLine(....) for all objects here using properties display information of each object.

Add break points and see how the program runs