

CODE:

Program.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Numerics;
using System.Text;
using System.Threading.Tasks;
using ClassLibrary1;
using static ClassLibrary1.Class1;

namespace PhaseEndProject
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Begin:
            Console.WriteLine(" 1.To Add Player \n 2.To Remove player by ID \n
3.Get Player by ID \n 4.Get Player by Name\n 5.Get All Players");
            int I = Convert.ToInt32(Console.ReadLine());
            switch (I)
            {
                case 1:
                    there:
                    OneDayTeam E = new OneDayTeam();
                    if (OneDayTeam.Oneday.Count != OneDayTeam.Oneday.Capacity)
                    {
                        Console.Write("Enter the Player Id:");
                        E.PlayerId = Convert.ToInt32(Console.ReadLine());
                        Console.Write("Enter the Player Name:");
                        E.PlayerName = Console.ReadLine();
                        Console.Write("Enter the Player Age:");
                        E.PlayerAge = Convert.ToInt32(Console.ReadLine());

                        E.Add(E);
                    }
                    else
                    {
                        Console.WriteLine(" You cannot add players capacity is
full");
                        goto Begin;
                    }
                    Console.WriteLine("Do you want to continue (yes/no?):please
type yes");
                    string s = Console.ReadLine();
                    if (s == "yes")
                    {
                        goto there;
                    }
                    break;
                case 2:
                    OneDayTeam V = new OneDayTeam();
                    Console.Write("Enter the Player Id you want to remove:");
                    int i = Convert.ToInt32(Console.ReadLine());
                    V.Remove(i);
                    break;
                case 3:
                    OneDayTeam E1 = new OneDayTeam();
                    Console.Write("Enter the Player Id you want to search:");
```

```

        int Y = Convert.ToInt32(Console.ReadLine());
        Player p = E1.GetPlayerById(Y);
        Console.WriteLine($"{p.PlayerId} {p.PlayerName}
{p.PlayerAge}");
        break;
    case 4:
        OneDayTeam E2 = new OneDayTeam();
        Console.WriteLine("Enter Player Name to search:");
        string i2 = Console.ReadLine();
        Player p1 = E2.GetPlayerByName(i2);
        Console.WriteLine($"{p1.PlayerId} {p1.PlayerName}
{p1.PlayerAge}");
        break;
    case 5:
        Console.WriteLine("List all players");
        List<Player> PP = new List<Player>();
        OneDayTeam W = new OneDayTeam();
        PP = W.GetAllPlayers();
        foreach (var item in PP)
        {
            Console.WriteLine($"{item.PlayerId} {item.PlayerName}
{item.PlayerAge}");
        }
        break;
    }
    Console.WriteLine("Do you want to continue(yes/no)? :type yes");
    string ji = Console.ReadLine();
    if (ji == "yes")
    {
        goto Begin;
    }
    Console.ReadLine();
}
}
}

```

## ClassLibrary1.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ClassLibrary1
{
    public class Class1
    {
        public interface ITeam
        {
            void Add(Player player);
            void Remove(int playerid);
            Player GetPlayerById(int playerid);
            Player GetPlayerByName(string playername);
            List<Player> GetAllPlayers();
        }
        public class Player

```

```

{
    private int _playerid;
    private string _playername;
    private int _playerage;
    public int PlayerId
    {
        get { return _playerid; }
        set { _playerid = value; }
    }
    public string PlayerName
    {
        get { return _playername; }
        set { _playername = value; }
    }
    public int PlayerAge
    {
        get { return _playerage; }
        set { _playerage = value; }
    }
}
public class OneDayTeam : Player, ITeam
{
    public static List<Player> Oneday = new List<Player>();
    public OneDayTeam()
    {
        Oneday.Capacity = 11;
    }
    public void Add(Player player)
    {
        Oneday.Add(player);
    }
    public List<Player> GetAllPlayers()
    {
        return Oneday;
    }
    public Player GetPlayerById(int playerid)
    {
        Player player = null;
        foreach (var item in Oneday)
        {
            if (item.PlayerId == playerid)
            {
                player = item;
                break;
            }
        }
        return player;
    }
    public Player GetPlayerByName(string playername)
    {
        Player player = null;
        foreach (var item in Oneday)
        {
            if (item.PlayerName == playername)

```

```

        {
            player = item;
            break;
        }
    }
    return player;
}

public void Remove(int playerid)
{
    Player player = null;

    foreach (var item in Oneday)
    {
        if (item.PlayerId == playerid)
        {
            Console.WriteLine("Player Removed successfully");
            player = item;
        }
    }
    Oneday.Remove(player);
}
}
}
}
}

```

GIT Link : <https://github.com/sundar2568223/Phase1EndProject-1.git>