Players.cs

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

namespace PhaseEndProject
{
    public class Player
    {
        public int PlayerId {  get; set; }

        public string PlayerName { get; set; }

        public int PlayerAge {  get; set; }
    }

    public interface ITeam
    {
        void Add(Player player);

        void Remove(int playerId);

        Player GetPlayerById(int playerId);

        Player GetPlayerByName(string playername);

        List<Player> GetAllPlayers();
    }

    public class OneDayTeam : ITeam
    {
        public static List<Player> oneDayTeam= new List<Player>();
        public int capacity = 11;

        public void Add(Player player)
        {
            if (oneDayTeam.Count < capacity)
            {
                oneDayTeam.Add(player);
            }
            else
            {
                Console.WriteLine("the team is already at full capacity(11
players)");
            }
        }

        public void Remove(int playerId)
        {
            Player playerToRemove = oneDayTeam.Find(p => p.PlayerId == playerId);
            if (playerToRemove != null)
            {
                oneDayTeam.Remove(playerToRemove);
            }
            else
            {
```

```
                Console.WriteLine("Player with id " + playerId + "not found in
the team");
            }
        }

        public Player GetPlayerById(int playerId)
        {
            return oneDayTeam.Find(p => p.PlayerId==playerId);
        }

        public Player GetPlayerByName(string playerName)
        {
            return oneDayTeam.Find(p =>p.PlayerName==playerName);
        }

        public List<Player> GetAllPlayers()
        {
            return oneDayTeam;
        }
    }
}
```

Program.cs

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

namespace PhaseEndProject
{
    internal class Program
    {
        static void Main(string[] args)
        {
            OneDayTeam cricketTeam = new OneDayTeam();

            while (true)
            {
                Console.WriteLine("Menu");
                Console.WriteLine("1. Add a Player");
                Console.WriteLine("2. Remove a Player by Id");
                Console.WriteLine("3. Get Player by Id");
                Console.WriteLine("4. Get Player by Name");
                Console.WriteLine("5. Get all Players");
                Console.WriteLine("Enter the choice");

                int choice = int.Parse(Console.ReadLine());

                switch (choice)
                {
                    case 1:
                        Console.Write("Enter Player ID: ");
                        int playerId = int.Parse(Console.ReadLine());
                        Console.Write("Enter Player Name: ");
```

```
                        string playerName = Console.ReadLine();
                        Console.Write("Enter Player Age: ");
                        int playerAge = int.Parse(Console.ReadLine());

                        Player newPlayer = new Player { PlayerId = playerId,
PlayerName = playerName, PlayerAge = playerAge };
                        cricketTeam.Add(newPlayer);

                        Console.WriteLine("Player is added successfully");
                        break;

                    case 2:
                        Console.Write("Enter Player ID to remove: ");
                        int playerIdToRemove = int.Parse(Console.ReadLine());
                        cricketTeam.Remove(playerIdToRemove);

                        Console.WriteLine("Player is removed successfully");
                        break;

                    case 3:
                        Console.Write("Enter Player ID to get: ");
                        int playerIdToGet = int.Parse(Console.ReadLine());
                        Player playerById =
cricketTeam.GetPlayerById(playerIdToGet);
                        if (playerById != null)
                        {
                            Console.WriteLine($"Player
found:Id:{playerById.PlayerId},Name: {playerById.PlayerName}, Age:
{playerById.PlayerAge}");
                        }
                        else
                        {
                            Console.WriteLine("Player not found.");
                        }
                        break;

                    case 4:
                        Console.Write("Enter Player Name to get: ");
                        string playerNameToGet = Console.ReadLine();
                        Player playerByName =
cricketTeam.GetPlayerByName(playerNameToGet);
                        if (playerByName != null)
                        {
                            Console.WriteLine($"Player found: ID:
{playerByName.PlayerId}, Age: {playerByName.PlayerAge}");
                        }
                        else
                        {
                            Console.WriteLine("Player not found.");
                        }
                        break;

                    case 5:
                        var allPlayers = cricketTeam.GetAllPlayers();
                        Console.WriteLine("All Players in the Team:");
                        foreach (var player in allPlayers)
                        {
                            Console.WriteLine($"ID: {player.PlayerId}, Name:
{player.PlayerName}, Age: {player.PlayerAge}");
                        }
                        break;

                }
```

```
                Console.WriteLine("do you want to continue yes/no");

                Console.ReadLine();
            }
        }
    }
}
```