**Case Study:**

1. FastPace Cricket Academy has decided to create a solution to maintain information about the teams’ players for one day game with the below functionalities:
2. User will be able to add a player to the team with details Player Id, Name, and Age.
3. User will be able to remove a player from the team by passing the player’s Id.
4. User will be able to get player details by passing the player’s Id.
5. User will be able to get player details by passing the player’s name.
6. User will be able to get all player details.
7. User will not be able to add more than 11 players to the team.

**Program:**

**Program.cs**

using System;

using System.Collections.Generic;

namespace ProjectPhase2

{

public class Program

{

static void Main(string[] args)

{

repeat:

Console.WriteLine("Enter 1:To Add Player 2:To Remove Player by Id 3.Get Player By Id 4.Get Player by Name 5.Get All Players:");

int ch = Convert.ToInt32(Console.ReadLine());

switch (ch)

{

case 1:

OneDayTeam o = new OneDayTeam();

if (OneDayTeam.oneDayTeam.Count <= OneDayTeam.oneDayTeam.Capacity)

{

Console.WriteLine("Enter Player ID");

o.PlayerId = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter Player Name:");

o.PlayerName = Console.ReadLine();

Console.WriteLine("Enter Player Age:");

o.PlayerAge = Convert.ToInt32(Console.ReadLine());

o.Add(o);

}

else

{

Console.WriteLine("You cannot add more than 11 Players.");

}

break;

case 2:

OneDayTeam o1 = new OneDayTeam();

Console.WriteLine("Enter Player ID to Remove:");

int id = Convert.ToInt32(Console.ReadLine());

o1.Remove(id);

break;

case 3:

OneDayTeam o2 = new OneDayTeam();

Console.WriteLine("Enter Player ID:");

int id1 = Convert.ToInt32(Console.ReadLine());

Player p = o2.GetPlayerById(id1);

Console.WriteLine("Player ID: " + p.PlayerId);

Console.WriteLine("Player Name: " + p.PlayerName);

Console.WriteLine("Player Age: " + p.PlayerAge);

break;

case 4:

OneDayTeam o3 = new OneDayTeam();

Console.WriteLine("Enter Player Name:");

string name = Console.ReadLine();

Player p1 = o3.GetPlayerByName(name);

Console.WriteLine("Player ID: " + p1.PlayerId);

Console.WriteLine("Player Name: " + p1.PlayerName);

Console.WriteLine("Player Age: " + p1.PlayerAge);

break;

case 5:

Console.WriteLine("All Players details:");

List<Player> all = new List<Player>();

OneDayTeam o4 = new OneDayTeam();

all = o4.GetAllPlayers();

foreach (var i in all)

{

Console.WriteLine("Player ID: " + i.PlayerId);

Console.WriteLine("Player Name: " + i.PlayerName);

Console.WriteLine("Player Age: " + i.PlayerAge);

}

break;

default:

Environment.Exit(0);

break;

}

Console.WriteLine("Do you want to continue (yes/no)?");

string ch1 = Console.ReadLine();

if (ch1 == "yes")

{

goto repeat;

}

else if (ch1 == "no")

{

Environment.Exit(0);

}

Console.ReadLine();

}

}

}

**Player.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ProjectPhase2

{

public class Player

{

private int \_playerid;

public int PlayerId

{

get { return \_playerid; }

set { \_playerid = value; }

}

private string \_playername;

public string PlayerName

{

get { return \_playername; }

set { \_playername = value; }

}

private int \_playerage;

public int PlayerAge

{

get { return \_playerage; }

set { \_playerage = value; }

}

}

}

**ITeam.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ProjectPhase2

{

interface ITeam

{

void Add(Player player);

void Remove(int playerId);

Player GetPlayerById(int playerId);

Player GetPlayerByName(string playerName);

List<Player> GetAllPlayers();

}

}

**OneDayTeam.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ProjectPhase2

{

class OneDayTeam: Player, ITeam

{

public static List<Player> oneDayTeam = new List<Player>();

public OneDayTeam()

{

oneDayTeam.Capacity = 11;

}

public void Add(Player player)

{

oneDayTeam.Add(player);

}

public void Remove(int playerId)

{

Player p = null;

foreach (var i in oneDayTeam)

{

if (i.PlayerId == playerId)

{

Console.WriteLine("Player{0} details is removed successfully", i.PlayerId);

p = i;

}

}

oneDayTeam.Remove(p);

}

public Player GetPlayerById(int playerId)

{

Player p = null;

foreach (var i in oneDayTeam)

{

if (i.PlayerId == playerId)

{

p = i;

break;

}

}

return p;

}

public Player GetPlayerByName(string playerName)

{

Player p = null;

foreach (var i in oneDayTeam)

{

if (i.PlayerName == playerName)

{

p = i;

break;

}

}

return p;

}

public List<Player> GetAllPlayers()

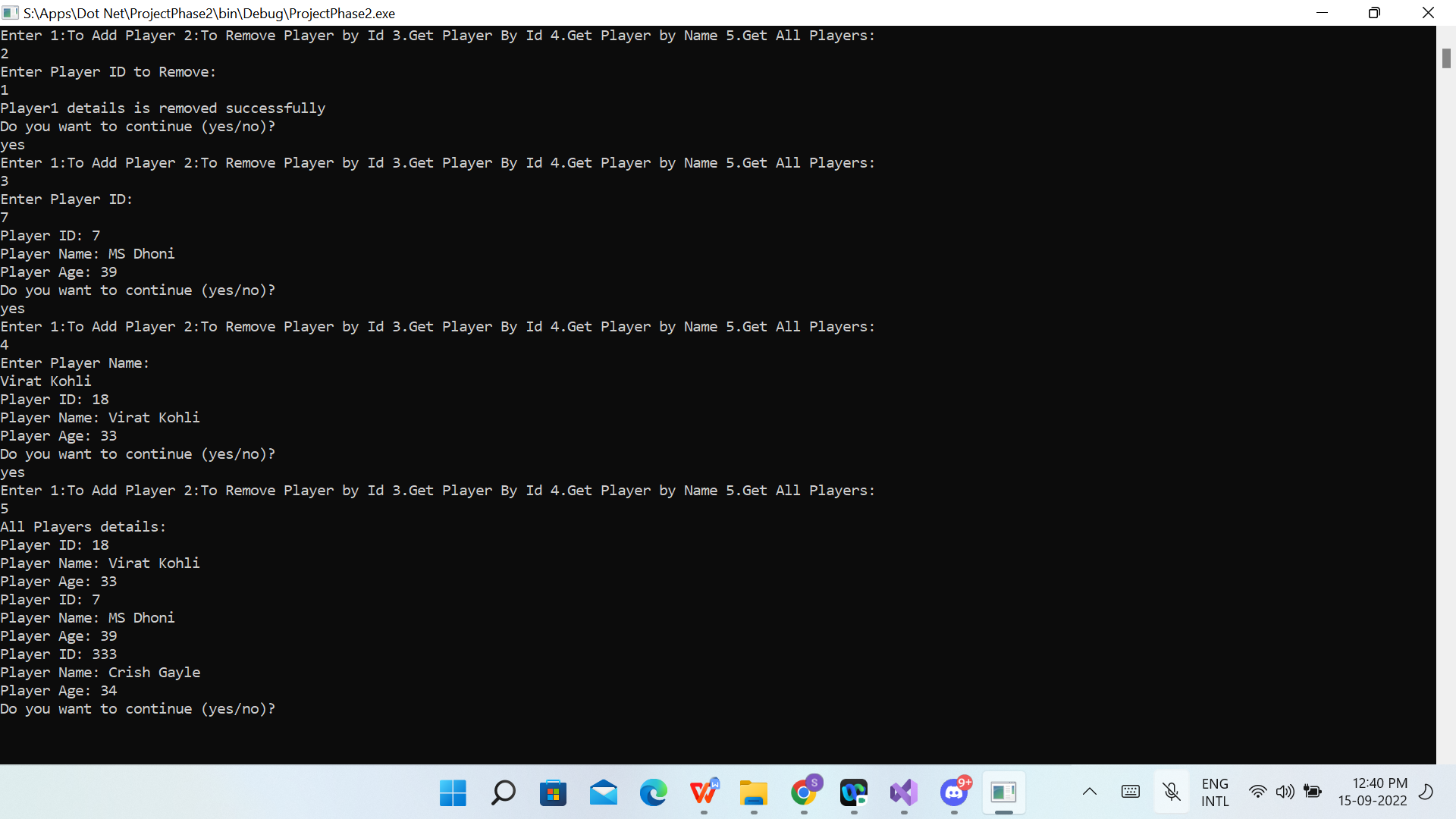
{

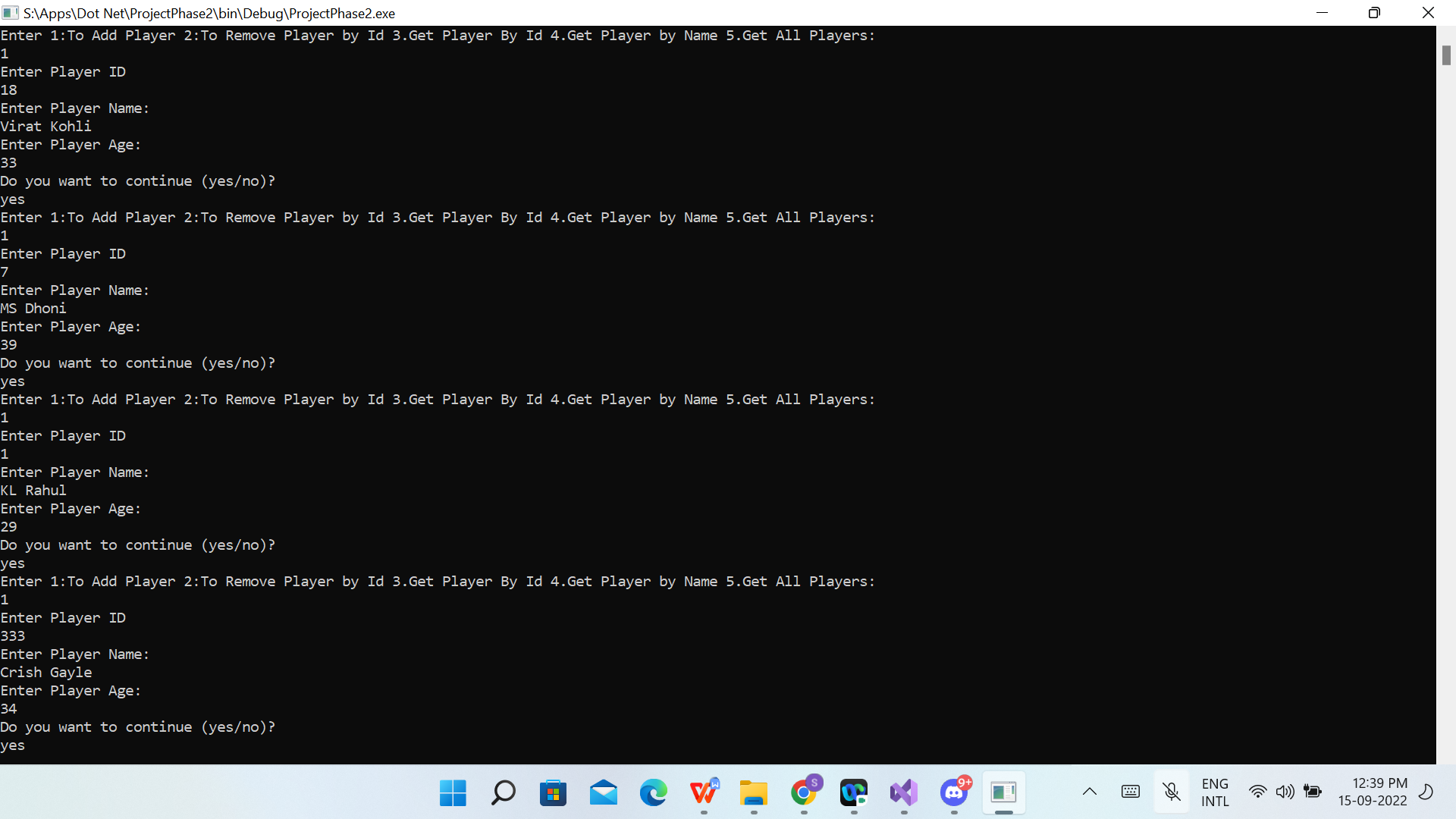
return oneDayTeam;

}

}

}

**Output:**

****