ALGORITHM OF THE PLAYER AND TEAM REQUIREMENTS:

**Step1:** Start Microsoft Visual Studio.

**Step 2: On the File menu, point to New, and then select Project.**

**Step 3:** Select **Visual C# Projects** under **Project Types**, and then select **Console Application** under **Templates**.

**Step 4:** Add the following code at the beginning of the Program.cs file and player.cs file.

**Step 5:** In player class, I just created three auto implemented property and also create interface. i created onedayteam derived class and also implement the interface. first, we created list in that we add player details then, remove player details using playerid, get the player details using playerid, get the player details using player name and then display the all the player details in the list.

**Step 6:** Add the following code to the Main method:

static void Main(string[] args)

{

string u;

do

{

Player p = new Player();

OneDayTeam oo = new OneDayTeam();

Console.Write("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 y = Convert.ToInt32(Console.ReadLine());

switch (y)

{

case 1:

p = new Player();

count++;

Console.Write("Enter Player Id: ");

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

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

p.PlayerName = Console.ReadLine();

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

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

if (count == 12)

{

Console.WriteLine("you can add only 11 players!!!!!!!!!!!!");

break;

}

else

{

oo.Add(p);

}

break;

case 2:

Console.Write("Enter Player Id to Remove:" );

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

oo.Remove(p.PlayerId);break;

case 3:

Console.Write("Enter Player Id:");

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

Player p11 = oo.GetPlayerById(p.PlayerId);

Console.Write(p11.PlayerId+" "+p11.PlayerName+" "+p11.PlayerAge+"\n");

break;

case 4:

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

p.PlayerName = Console.ReadLine();

Player p1 = oo.GetPlayerByName(p.PlayerName);

Console.Write(p1.PlayerId + " " + p1.PlayerName + " " + p1.PlayerAge+"\n"); break;

case 5: oo.GetAllPlayers(); break;

default:

break;

}

Console.Write("Do You Want to Continue(yes/no)?:");

u = Console.ReadLine();

} while (u=="yes");

PLAYER.CS:

interface ITeam

{

void Add(Player player);

void Remove(int playerId);

Player GetPlayerById(int playerId);

Player GetPlayerByName(string playerName);

List<Player> GetAllPlayers();

}

public class Player

{

public int PlayerId { get; set; }

public string PlayerName { get; set; }

public int PlayerAge { get; set; }

}

public class OneDayTeam : Player,ITeam

{

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

public OneDayTeam()

{

}

public void Add(Player player)

{

oneDayTeam.Add(player);

Console.WriteLine("player is added successfully");

//throw new NotImplementedException();

}

public List<Player> GetAllPlayers()

{

foreach (var item in oneDayTeam)

{

Console.Write(item.PlayerId+" ");

Console.Write(item.PlayerName+" ");

Console.WriteLine(item.PlayerAge);

}

return oneDayTeam;

throw new NotImplementedException();

}

public Player GetPlayerById(int playerId)

{

Player pl = new Player();

foreach (var item in oneDayTeam)

{

if (item.PlayerId == playerId)

{

pl = item;

break;

}

}

return pl;

//throw new NotImplementedException();

}

public Player GetPlayerByName(string playerName)

{

Player pl = new Player();

foreach (var item in oneDayTeam)

{

if (item.PlayerName == playerName)

{

pl = item;

break;

}

}

return pl;

//throw new NotImplementedException();

}

public void Remove(int playerId)

{

var r = oneDayTeam.Find(s => s.PlayerId == playerId);

oneDayTeam.Remove(r);

Console.WriteLine("player is removed successfully");

//throw new NotImplementedException();

}

}

**Step 7:** On the **Debug** menu, select **Start** to compile and to run the application. Press ENTER to close the **Console** window.