1. Create a class called Team that has properties to hold the team name, number of wins, number of draws, number of losses, number of games. All these properties have private setters.
2. Add a property called Points which calculates the points based on the values for Wins, Draws and Losses. 3 Points for a win, 1 for a draw and 0 for a loss. You need to use the long hand means of writing a property here as there is a calculation in the getter.
3. Create a constructor where the team name is set. It should take in one parameter. In the program.cs file create **Team** objects for five teams such as the following:

Team SligoRovers = new Team("Sligo Rovers");

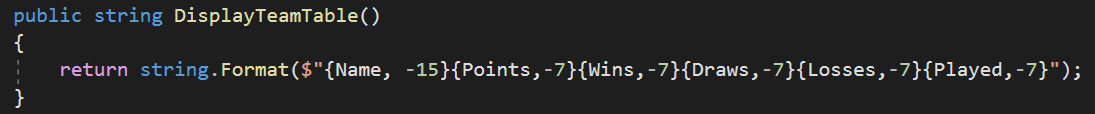
Team FinnHarps = new Team("Finn Harps");

Team GalwayUnited = new Team("Galway United");

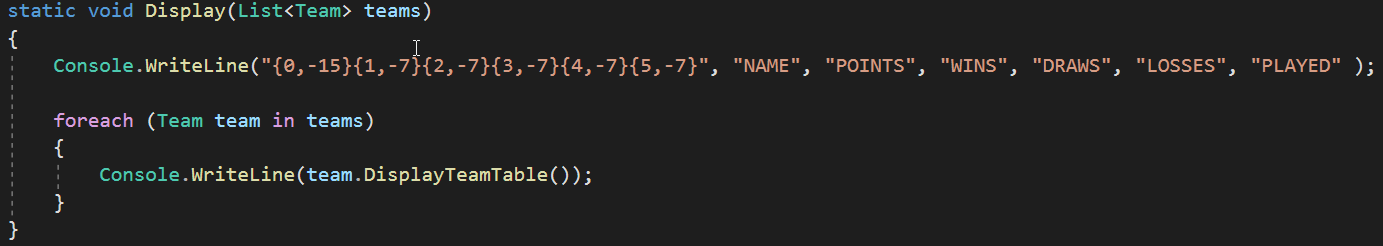
Team DerryCity = new Team("Derry City");

Team Dundalk = new Team("Dundalk");

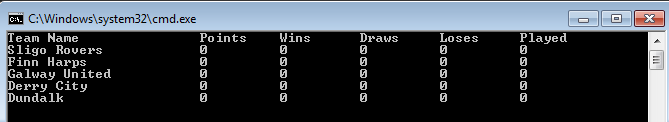
1. In the program.cs file create an List of type Team to hold all the Team objects and add the Team objects created to the list.
2. Create a method in the Team class called DisplayTeamTable. This method which is shown below returns a formatted string with team details.



1. Create a method in program.cs called DisplayTeams which is passed a List of Teams and displays the details by calling the DisplayTeamTable.



1. Use this method to display the league table as shown below.



1. Create an enum called Result that has 3 elements {Win, Draw, Lose}. **(We have not covered this, some research needed).**
2. Create a method called AddResult in the Team class that takes in one parameter of type Result and depending on the result update the read only properties.
3. For each Team object call the AddResult method and supply a Result such as below:

SligoRovers.AddResult(Result.Win);

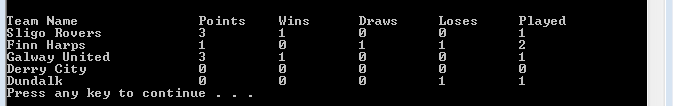
FinnHarps.AddResult(Result.Loss);

FinnHarps.AddResult(Result.Draw);

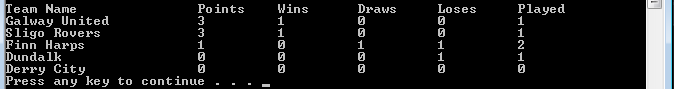
GalwayUnited.AddResult(Result.Win);

Dundalk.AddResult(Result.Loss);

1. Display the league table as shown below using the Display method previously written.



1. Implement the **IComparable** interface to allow the **Team** objects be compared based on points.
2. Sort the array and then reverse it so the team with the highest points is listed first.



1. Add another class called Player. This is a small class that has player name and position as properties. Add a property to Team which is a List of Players. Create some players and add them to the Team. In program.cs display the players on one team.

