You will write a small application that will allow football teams to store information about matches in which they have taken part. A match is between 2 teams. **The home team is playing at their own ground, the away team is playing away from home. A match is called a home win if the home team wins and an away win if the away team wins. Otherwise the match is called a draw. The goals scored by the home team are called home goals and those by the away team are called away goals. A team gains 3 points if it wins, 1 point if it's a draw and 0 points otherwise.**

The application will collect, store and report information about matches that a

team has played.

There will be two Java classes in the application, one for a match, called **FootMatch**, and one for a team, called **Team.**

You will also write a test class.

The **Team** class will calculate and update results from each match that the team has played in.

The FootMatch class

Fields:

There will be 2 fields with data type String, homeTeamName and awayTeamName and 2 integer fields called homeGoals and awayGoals.

Methods:

In the FootMatch class the result of a match will be set by a **setResult** method **after** the match has taken place.

There will be **accessor methods** for all fields.

There will also be a **toString** method that will return the current values of all the fields with suitable labels.

The Team class

Fields:

A String field called **teamName** and
an **ArrayList** called **results** storing **FootMatch** objects.(see extension notes)

Three integer fields named **totalGoalsFor, totalGoalsAgainst** and **totalPoints**.
An integer array field called stats.

Methods:

getResults(FootMatch aMatch)

A **getMatchResult** method will reference a football match through a FootMatch object parameter. It will have to work out if the team was the home team or the away team. If neither of these is true it will print a message saying the team was not involved in the match.

If the team was playing it will calculate which goals were scored by the team and which goals were scored against the team. It will also add each match to the list.

It will then pass this information to a
private **updateStatistics** method, **updateStatistics (int ourGoals, int theirGoals)**

The **updateStstistics** method will calculate the totalGoalsFor, totalGoalsAgainst and the totalPoints for all matches and return this information.

So as before, if the team name is Everton and the result is Leeds 2 Everton 4 it will add this match to the list.

Extension

To gain the higher marks you will need to add other methods that provide a variety of ways to view the match records of a team. For example to list matches against a particular team or of matches that are draws.

Test class
As you develop your solution test it by adding to your test class appropriate tests.

OOPJ: Football Results

Foot Match:

```
public class FootMatch
{
	private String homeTeamName;
	private String awayTeamName;
	private int homeGoals;
	private int awayGoals;
	private String result;

	/**
	 *constructor (set up before match takes place
	 */
	public FootMatch(String hteam, String ateam)
	{
		homeTeamName = hteam;
		awayTeamName = ateam;
		homeGoals = 0;
		awayGoals = 0;
	}
	/**
	 * After match is played set the result
	 */
	public void setResult(int hgoals, int agoals)
	{
		homeGoals = hgoals;
		awayGoals = agoals;
		if (hgoals > agoals) {
			result = homeTeamName + "(homeTeam)won the match";
		} else if (agoals > hgoals) {
			result = awayTeamName + "(awayTeam) won the match";
		} else if (agoals == hgoals) {
			result = "This match    is a draw";
		}
		System.out.println(result);
	}
```

```
	/**
	 * get methods for the team object to invoke
	 */
	public String getHomeTeamName()
	{
		return homeTeamName;
	}

	public String getAwayTeamName()
	{
		return awayTeamName;
	}

	public int getHomeGoals()
	{
		return homeGoals;
	}

	public int getAwayGoals()
	{
		return awayGoals;
	}

	public void printResult()
	{
		if (homeGoals > awayGoals) {
			result = "home win";
		} else if (awayGoals > homeGoals) {
			result = "away win";
		} else if (awayGoals == homeGoals) {
			result = "draw";
		}
	}

	public String toString()
	{
		return "Result: " + homeTeamName + " " + homeGoals + " : " +
awayTeamName + " " + awayGoals;
```

```
	}

}
```

Team:

```
import java.util.ArrayList;

public class Team
{
	// goals for and against are set at zero and then can be increased by accessing a match involving the team
	private String teamName;
	private ArrayList<FootMatch> results;
	private int totalGoalsFor;
	private int totalGoalsAgainst;
	private int totalPoints;

	//constructor function sets name of team
	public	Team(String tname)
	{
		teamName = tname;
		results = new ArrayList<FootMatch>();
		totalGoalsFor = 0;
		totalGoalsAgainst = 0;
		totalPoints = 0;
	}

	/*
	 * this method checks if the match involves this team
	 * then gets the goals for and against this team
	 * adding them to the cumulative record.
	 */

	public void getMatchResult(FootMatch match) {
		int ourGoals = 0;
		int theirGoals = 0 ;
		if (match.getHomeTeamName().equals(teamName)) {
			ourGoals = match.getHomeGoals();
```

```
            theirGoals = match.getAwayGoals();
            totalGoalsFor += ourGoals;
            totalGoalsAgainst += theirGoals;
            if (ourGoals > theirGoals) {
                totalPoints +=3;
            } else if (ourGoals == theirGoals) {
                totalPoints +=1;
            }
        } else if (match.getAwayTeamName().equals(teamName)) {
            ourGoals = match.getAwayGoals();
            theirGoals = match.getHomeGoals();
            totalGoalsFor += ourGoals;
            totalGoalsAgainst += theirGoals;
            if (ourGoals > theirGoals) {
                totalPoints += 3;
            } else if (ourGoals == theirGoals) {
                totalPoints += 1;
            }
        } else {
            System.out.println("This match does not involve " + teamName);
        }
        results.add(match);
}

private void upDateStatistics()
{
        totalPoints = 0;
        for (FootMatch match : results) {
            if (match.getHomeTeamName().equals(teamName)) {
                if (match.getHomeGoals() > match.getAwayGoals()) {
                    totalPoints += 3;
                } else if (match.getHomeGoals() == match.getAwayGoals()) {
                    totalPoints += 1;
                }
            } else if (match.getAwayTeamName().equals(teamName)) {
                if (match.getAwayGoals() > match.getHomeGoals()) {
                    totalPoints += 3;
                } else if (match.getAwayGoals() == match.getHomeGoals()) {
```

```
                    totalPoints += 1;
                }
            }
        }
    }

    public String getTeamName()
    {
        return teamName;
    }

    public void listHomeMatches()
    {
        System.out.println("Home Matches:");
        for (FootMatch match : results) {
            if (match.getHomeTeamName().equals(teamName)) {
                System.out.println(match);
            }
        }
    }

    public void listAwayMatches()
    {
        System.out.println("Away Matches:");
        for (FootMatch match : results) {
            if (match.getAwayTeamName().equals(teamName)) {
                System.out.println(match);
            }
        }
    }

    public void listAllMatches()
    {
        System.out.println("All Matches:");
        for (FootMatch match : results) {
            System.out.println(match);
        }
    }
```

```
    public void listMatchesByOpponentsName(String aName)
    {
        System.out.println("Matches Against " + aName + ":");
        for (FootMatch match : results) {
            if (match.getHomeTeamName().equals(teamName) &&
match.getAwayTeamName().equals(aName)) {
                System.out.println(match);
            } else if (match.getHomeTeamName().equals(aName) &&
match.getAwayTeamName().equals(teamName)) {
                System.out.println(match);
            }
        }
    }

    public void listMatchesWon()
    {
        System.out.println("Matches Won: ");
        for (FootMatch match : results) {
            if (match.getHomeTeamName().equals(teamName) &&
match.getHomeGoals() > match.getAwayGoals()) {
                System.out.println(match);
            } else if (match.getAwayTeamName().equals(teamName) &&
match.getAwayGoals() > match.getHomeGoals()) {
                System.out.println(match);
            }
        }
    }

    public void listMatchesLost()
    {
        System.out.println("Matches Lost: ");
        for (FootMatch match : results) {
            if (match.getHomeTeamName().equals(teamName) &&
match.getHomeGoals() < match.getAwayGoals()) {
                System.out.println(match);
            } else if (match.getAwayTeamName().equals(teamName) &&
match.getAwayGoals() < match.getHomeGoals()) {
```

```
                    System.out.println(match);
                }
            }
        }

        public String toString()
        {
            return "Total Goals For:" + totalGoalsFor + "\\nTotal Goals Against:" +
totalGoalsAgainst + "\\nTotal Points:" + totalPoints;
        }
}
```

test method: TestAss2

```
public class TestAss2
{
        private Team team1;
        private Team team2;
        private Team team3;
        private Team team4;
        private FootMatch match12;
        private FootMatch match31;
        private FootMatch match23;
        private FootMatch match42;
        public TestAss2()
        {
            team1 = new Team("Brighton");
            team2 = new Team("Manchester United");
            team3 = new Team("Liverpool FC");
            team4 = new Team("Manchester City");

            match12 = new FootMatch("Brighton", "Manchester United");
            match31 = new FootMatch("Liverpool FC", "Brighton");
            match23 = new FootMatch("Manchester United", "Liverpool FC");
            match42 = new FootMatch("Manchester City", "Manchester United");
        }

        public void testTeam()
        {
```

```
            team1.getMatchResult(match12);
            team2.getMatchResult(match12);
            team3.getMatchResult(match12);

            team3.getMatchResult(match31);
            team1.getMatchResult(match31);

            team2.getMatchResult(match23);
            team3.getMatchResult(match23);

            team4.getMatchResult(match42);
            team2.getMatchResult(match42);

            System.out.println(team1);
            System.out.println(team2);
            System.out.println(team3);
            System.out.println(team4);
        }

        public void testMatches()
        {
            match12.setResult(4, 1);
            System.out.println(match12);
            match31.setResult(2, 2);
            System.out.println(match31);
            match23.setResult(0, 0);
            System.out.println(match23);
            match42.setResult(0, 2);
            System.out.println(match42);
        }
}
```

Results:

```
testAss21.testMatches();
Brighton(homeTeam)won the match
Result: Brighton 4 : Manchester United 1
This match    is a draw
Result: Liverpool FC 2 : Brighton 2
This match    is a draw
```

Result: Manchester United 0 : Liverpool FC 0
Manchester United(awayTeam) won the match
Result: Manchester City 0 : Manchester United 2

testAss21.testTeam();
This match does not involve Liverpool FC
Total Goals For:6\nTotal Goals Against:3\nTotal Points:4
Total Goals For:3\nTotal Goals Against:4\nTotal Points:4
Total Goals For:2\nTotal Goals Against:2\nTotal Points:2
Total Goals For:0\nTotal Goals Against:2\nTotal Points:0

Submission

1 Mar at 17:20
Grade: 37 (50 pts possible)