

CHRISTOPHER NEWPORT UNIVERSITY
DEPARTMENT OF PHYSICS, COMPUTER SCIENCE & ENGINEERING
CPSC 360 –PROGRAMMING LANGUAGES CONCEPTS

*** Assignment 1: C#: FIFA World Cup ***

Instructor: Dr. Roberto A. Flores

Goal

Use C# for problem solving.

Description

In anticipation for the 2022 FIFA World Cup, you want to write a program to calculate rankings and statistics of teams during the first stage (group round).

In soccer, the result of a match between 2 teams is given by goals scored: the team that scores more goals wins, the team that scores less goals loses, and both tie if they score the same number of goals.

A team earns 3 points for a win, 1 point for a tie and 0 points for a loss.

Teams are ranked using the following incremental rules (i.e., next rule applies when there are ties):

1. Most points earned
2. Most wins
3. Most goal difference (i.e., goals scored - goals against)
4. Most goals scored
5. Less matches played
6. Alphabetical order (i.e., ascending)

Implementation

Write a method “getRankings” (below) that receives match results and returns the rankings of teams.

```
public static string[] getRankings(string[] matches)
```

The parameter “matches” is an array of strings where the first string (index 0) indicates the name of the group (e.g., “World Cup 2015: Group G”), and where each subsequent string has a result of the form:

```
"<TeamA-name>#<TeamA-goals>@<TeamB-goals>#<TeamB-name>"
```

where <TeamA-name> and <TeamB-name> are the names of opposing teams, and <TeamA-goals> and <TeamB-goals> are the goals scored by these teams respectively. For example, the string “The Netherlands#1@3# Spain” shows that Spain won the match against The Netherlands by a score of 3 to 1.

Results are returned as an array of strings where the first string (index 0) is the name of the group (as given in the input), and where subsequent strings indicates the ranking of each one of the teams. These strings will be of the form

```
"<a> <Team-name> <b>p, <c>g (<d>-<e>-<f>), <g>gd (<h>-<i>)"
```

where <a> is a number with the team’s rank, is total points, <c> is games played, <d> is games won, <e> is games tied, <f> is games lost, <g> is goal difference (i.e., scored-against), <h> is goals scored, and <i> is goals against. For example, given the array input below:

```
{ "World Cup 2010: Group H",  
  "Honduras#0@1#Chile",  
  "Spain#0@1#Switzerland",
```

```

        "Chile#1@0#Switzerland",
        "Spain#2@0#Honduras",
        "Chile#1@2#Spain",
        "Switzerland#0@0#Honduras"
    }

```

the method returns the following string array:

```

    { "World Cup 2010: Group H",
      "1) Spain 6p, 3g (2-0-1), 2gd (4-2)",
      "2) Chile 6p, 3g (2-0-1), 1gd (3-2)",
      "3) Switzerland 4p, 3g (1-1-1), 0gd (1-1)",
      "4) Honduras 1p, 3g (0-1-2), -3gd (0-3)"
    }

```

Initial Files

You are given the initial files *Program.cs* (to write your solution) and *UnitTest.cs* (unit test cases, which you must not modify). Your implementation must begin on the method defined in *Program.cs*. You can add other methods or classes if you want.

Visual C#

You'll need Microsoft Visual Studio (for Mac or Windows) to implement this assignment. Visual Studio can be downloaded for free from Microsoft's website (link shown in lecture). When installing it (using Visual Studio Installer), make sure to include (in addition to the Visual Studio core editor) .NET desktop development (including the .NET framework development tools indicated under Installation details).

Project setup (on Windows): In Visual Studio choose menu options "File >> New >> Project... >> Visual C# >> Get Started >> Console App (.NET Core)" to create a new console-based project. In the "Solution Explorer" view, delete file "Program.cs" (created by Visual Studio) from the project; then right-click on the name of the project (2nd item), choose "Add >> Existing Item" and browse to import the "Program.cs" file given to you; then right-click on the name of the solution (1st item) and choose "Add >> New Project >> Visual C# >> Test >> MSTest Test Project (.NET Core)" to create a unit test project. Delete the file "UnitTest1.cs" created by Visual Studio. Right-click on this new project and choose "Add >> Existing Item" to import the unit file given to you. Lastly, create a reference from the unit test project to the console project by right-clicking on the unit test project and selecting "Add >> Project Reference" and checking on the name of your console application.

At this point you can build the solution and run tests using "Test >> Run All Tests" from the menu (of course, all tests fail right now but there should not be any compile errors).

Get acquainted with the debugger. It'll be your main tool to fine-tune your program.

Groups

This assignment is done individually or in teams of 2 people as assigned by the instructor. Each group should *submit one copy* of the assignment with a comment at the top of the source file(s) indicating the **percent effort** that each student spent on the project.

Submission

Source code (i.e., "*.cs" files only) must be submitted through Scholar by its due date.

Grading

Your program will be tested according to the functionality specified in this description and will be graded according to the number of test cases it passes. Answers must not be hard-coded.

Each of the tests is worth an equal number of points, up to a maximum of 100.

Be reminded that no code should be disclosed to nor taken from others (including online sources) as described in the “Honor Code” section in the syllabus.

...