Lab2OOP

April 5, 2017

1 COMP 10020 Introduction to Programming 2

1.1 Lab 2 - OO Rugby Tournaments

1.2 SOLUTIONS!!!!

In this lab you will be tasked with exercising your key Python programming skills. The **Pick & Go Test Match Results Database** (http://www.lassen.co.nz/pickandgo.php) contains the results of every international rugby match played since 1875. The following code block reads data from a data file (*'RugbyResultsData.csv'*) scraped from Pick & Go and stores this in a list of dictionary objects, each of which contains the detaisl of a match. The details stored about each match are stored in a dictionary object with the following keys:

- **Date**: The date on which the match was played
- **Day**: The day of the week on which the match was played
- **Year**: The year in which the match was played
- **Team_1**: The home team (three letter country code, e.g. IRL = Ireland, NZL = New Zealand)
- **Team_2**: The away team (three letter country code, e.g. IRL = Ireland, NZL = New Zealand)
- **Team_1_Score**: The score achieved by the home team.

- **Team_2_Score**: The score achieved by the away team.

- **Team_1_Tries**: The number of tries scored by the home team.
- **Team_2_Tries**: The number of tries scored by the away team.
- **Neutral**: Was the match played at a neutral venue?

1.2.1 Question 1

Can you write a Python class, called **Match**, to store the details of a rugby game?

```
In [1]: # Write code here
        class Match:

            # A constructor called when an object of the class is instantiated.
            def __init__(self, date, day, year, team_1, team_2, team_1_score, team_

                self.date = date
                self.day = day
                self.year = year

                self.team_1 = team_1
                self.team_2 = team_2

                self.team_1_score = team_1_score
                self.team_2_score = team_2_score
                self.team_1_tries = team_1_tries
                self.team_2_tries = team_2_tries

            def show(self):
                print(self.date + ": " + self.team_1 + " " + str(self.team_1_score)
```

1.2.2 Question 2

Can you adjust the code written in the last lab to read the data from *'RugbyResultsData.csv'* into a list of **Match** objects?

```
In [2]: # Adjust this code to create a list of Match objects

        matches = list()
        count = 0
        with open('RugbyResultsData.csv') as f:
            for line in f:
                words = line.split(',')

                match = Match(words[0],words[1],words[2],words[3],words[4],int(word
                matches.append(match)
                count = count + 1

        print(str(count) + " matches loaded")
```

```
3234 matches loaded
```

1.2.3 Question 3

Add a *print* method to your **Match** class to print the details of a match (make it look nice!). Iterate through the list of matchs and print the details of each.

```
In [3]: # Write to iterate through the list of match objects and print the details
        for m in matches[-10:-1]:
            m.show()
```

```
08 Oct 2016: ARG 21 - 33 AUS
22 Oct 2016: NZL 37 - 10 AUS
05 Nov 2016: NZL 29 - 40 IRE
05 Nov 2016: WAL 8 - 32 AUS
12 Nov 2016: ITA 10 - 68 NZL
```

```
12 Nov 2016: SCO 22 - 23 AUS
12 Nov 2016: FRA 52 - 8 SAM
12 Nov 2016: ENG 37 - 21 SAF
12 Nov 2016: IRE 52 - 21 CAN
```

1.2.4 Question 4

Can you write a **Tournament** class to represent the 6 nations rugby tournament. This class should be able to do three things:

- Store the details of all the matchs in the tournament
- Add a match to the tournament
- Calculate the points achieved by each team in the tournament
- Print a table showing the standings for each team after the games in the tournament. To calculate the points each team receives 3 points for a win, 1 point for a draw, and no points for a loss.

```
In [11]: class TeamStanding:

             def __init__(self, team):
                 self.team = team
                 self.points = 0
                 self.won = 0
                 self.lost = 0
                 self.drawn = 0
                 self.points_for = 0
                 self.points_against = 0
                 self.points_diff = 0
                 self.tries_for = 0
                 self.tries_against = 0
                 self.tries_diff = 0

             def addMatch(self, match, points, result):

                 if self.team != match.team_1 and self.team != match.team_2:
                     return

                 self.points += points

                 if result == "win":
                     self.won += 1
                 elif result == "loss":
                     self.lost += 1
                 elif result == "draw":
                     self.drawn += 1

                 if self.team == match.team_1:
                     self.points_for += match.team_1_score
                     self.points_against += match.team_2_score

                     self.tries_for += match.team_1_tries
                     self.tries_against += match.team_2_tries

                 elif self.team == match.team_2:

                     self.points_for += match.team_2_score
                     self.points_against += match.team_1_score

                     self.tries_for += match.team_2_tries
                     self.tries_against += match.team_1_tries

                 self.points_diff = self.points_for - self.points_against
                 self.tries_diff = self.tries_for - self.tries_against


             def show(self):
                 print(self.team  + "\t" + str(self.points)  + "\t" + str(self.won)

In [43]: class Tournament:

             def __init__(self):
                 self.matches = list()
                 self.standings = dict()

             def addMatch(self, match):

                 self.matches.append(match)

                 if match.team_1_score > match.team_2_score:
                     team_1_result = "win"
                     team_2_result = "loss"
                     team_1_points = 2
                     team_2_points = 0
                 elif match.team_2_score > match.team_1_score:
                     team_1_result = "loss"
                     team_2_result = "win"
                     team_1_points = 0
                     team_2_points = 2
                 else:
                     team_1_result = "draw"
                     team_2_result = "draw"
                     team_1_points = 1
                     team_2_points = 1
```

```
            if match.team_1 not in self.standings:
                self.standings[match.team_1] = TeamStanding(match.team_1)

            self.standings[match.team_1].addMatch(match, team_1_points, team_1

            if match.team_2 not in self.standings:
                self.standings[match.team_2] = TeamStanding(match.team_2)

            self.standings[match.team_2].addMatch(match, team_2_points, team_2

        def show(self):
            print("Team"  + "\t" + "PTS"  + "\t" + "W"  + "\t" + "L"  + "\t" +
            for s in self.standings:
                self.standings[s].show()
```

1.2.5 Question 5

The file *2016_6Nations_Results.csv* contains the results of each match from the 2016 Rugby 6 Nations tournament. Load the data from this file, create a **Tournament** object that stores all of the matches.

```
In [44]: six_nations = Tournament()

         count = 0
         with open('2016_6Nations_Results.csv') as f:
             for line in f:
                 words = line.split(',')

                 match = Match(words[0],words[1],words[2],words[3],words[4],int(wor

                 six_nations.addMatch(match)

                 count = count + 1

         print(str(count) + " matches loaded")
```

```
15 matches loaded
```

1.2.6 Question 6

Print the final standings table for the tournament.

```
In [45]: # Write code here
         six_nations.show()
```

```
Team        PTS         W         L         D         DIFF         TDIFF
FRA         4         2         3         0         -27         -2
```

```
WAL          7          3          1          1          62          10
ENG          10          5          0          0          62          9
SCO          4          2          3          0          7          -2
ITA          0          0          5          0          -145          -21
IRE          5          2          2          1          41          6
```

```
In [ ]:
```