ECE 122: Introduction to Programming for ECE - Spring 2024

Project 2: Building a Soccer Player Database

Using Classes and OOP to Create a Database

Due Date/Deadline: see website, class policy and Moodle for submission

Description

The goal of the project is to build a simple system to evaluate points for a fantasy soccer team. We will limit your team to 10 players from the English Premier League, La Liga, and Bundesliga. Each student will develop their own unique way of determining points. You do not need to have much background knowledge about soccer for this assignment. You will use the following table of players and data and will end up combining data values to form a total point value for each player on your team.

In this project you will create a database implemented in a list that includes objects of type **Player**.

Each player is identified by the following details:

1. Player's name
2. Player's league
3. Goals so far this season
4. Assists so far this season
5. Passes so far this season

Our fantasy team will have the following 10 players with stats accurate as of 3/10/24 for the 2023/24 seasons.

ID	Name	League	Goals	Assists	Passes
1	Erling Haaland	EPL	18	5	270
2	Robert Lewandowski	La Liga	12	6	414
3	Harry Kane	Bundesliga	30	6	472
4	Son Heung-Min	EPL	14	8	660
5	Julian Alvarez	EPL	8	8	915
6	Leroy Sane	Bundesliga	8	11	903
7	Mohamed Salah	EPL	15	9	686
8	Jude Bellingham	La Liga	16	3	1164
9	Toni Kroos	La Liga	1	7	1908
10	Sehrou Guirassy	Bundesliga	21	1	494

Table 1: Ten players and the related info

The project include multiple files:

1. **Player.py**: This file contains the user-defined class for player information and all of the functions needed to manipulate the associated data.

2. **FantasyCalculator.py**: This file includes the main program. The program should include menu display functions, a list that includes all the players, and various data manipulation functions.

Submission/Grading Proposal

Only **one zip file** must be submitted on moodle. The zip file should contain your **Player.py** and **FantasyCalculator.py** files. Make sure you know how to create and submit a zip file before the submission date (and that your zip file is not empty once submitted). Submit a zip file often as you make progress, you can always update it before the deadline (this way you get an extra copy on the cloud if you accidentally erase your files). This project will be graded out of 100 points:

1. Your program should implement all basic functionality/Tasks and run correctly (90 pts).
2. Overall programming style with comments (including function header doc-string) (10 pts).

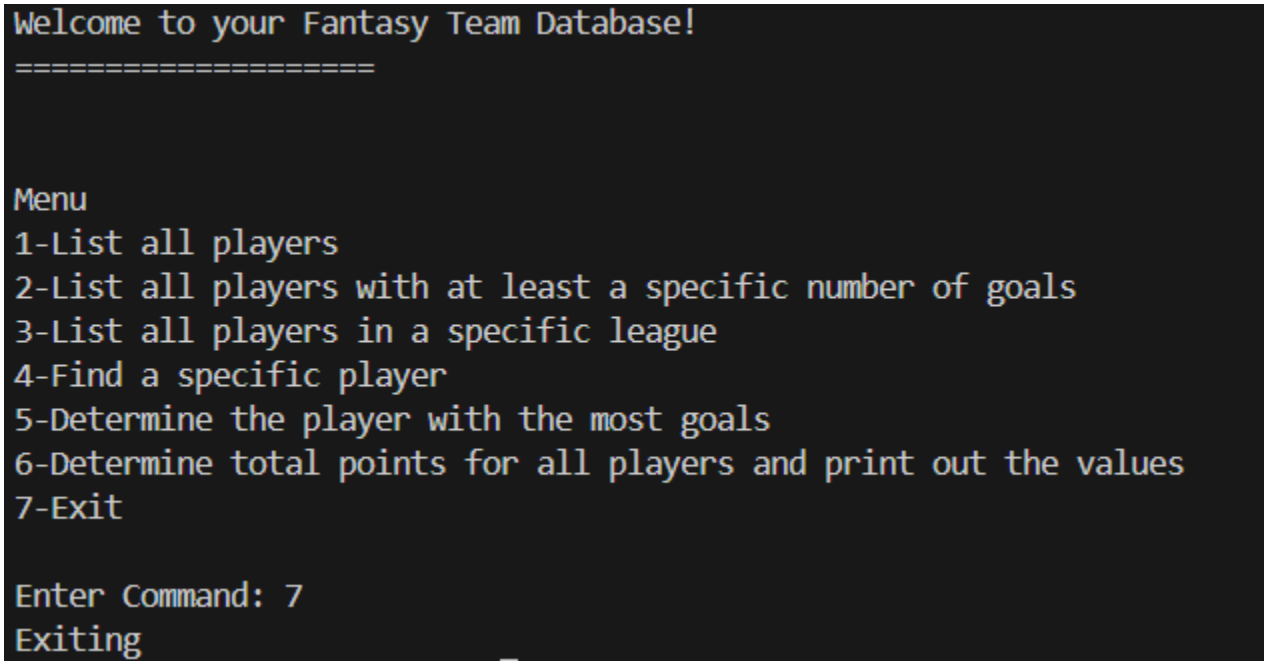
Overview of The Main Program Functionality

Task0

When you execute your code, it should print a menu with the following options:

1. List all players
2. List all the players with at least a specific number of goals
3. List all players in a specific league
4. Find a specific player
5. Determine the player with the most goals
6. Determine total points for all players and print out the values
7. Exit

Below the menu there should be a prompt asking the user for input. The output when you first run your program and exit should appear as follows:



How to proceed:

1. In **FantasyCalculator.py**, your code should print a welcome message.
2. Write a function **display_menu** in FantasyCalculator.py to print the menu shown in the above screenshot.
3. Consider using a while loop to repetitively print the menu and perform a selected operation until the user selects option 7 (Exit).
4. If the user selects option 7, your code will print the message “Exiting” and terminate the loop.

Task1

Implement the code that is executed when a user selects option 1. If the user selects this option when prompted to enter a command, your code should display the details of all players currently present in the database. Note that you will need to “hard code” this information in FantasyCalculator.py when you create and initialize the Players objects. This list should be followed by the menu and prompt, giving the user an opportunity to enter another command. The resulting output should look like the following:

```
Welcome to your Fantasy Team Database!
==================


Menu
1-List all players
2-List all players with at least a specific number of goals
3-List all players in a specific league
4-Find a specific player
5-Determine the player with the most goals
6-Determine total points for all players and print out the values
7-Exit

Enter Command: 1
Name                        League       Goals      Assists      Passes
Erling Haaland                 EPL          18            5         270
Robert Lewandowski          La Liga         12            6         414
Harry Kane                Bundesliga        30            6         472
Son Heung-Min                  EPL          14            8         660
Julian Alvarez                 EPL           8            8         915
Leroy Sane                Bundesliga         8           11         903
Mohamed Salah                  EPL          15            9         686
Jude Bellingham             La Liga         16            3        1164
Toni Kroos                  La Liga          1            7        1908
Sehrou Guirassy           Bundesliga        21            1         494

Menu
1-List all players
2-List all players with at least a specific number of goals
3-List all players in a specific league
4-Find a specific player
5-Determine the player with the most goals
6-Determine total points for all players and print out the values
7-Exit

Enter Command:
```

How to proceed:

1. Create a class Player in Player.py that will contain attributes for each Player (e.g. name, league, goals,assists, and passes). You should include a constructor in your class to allow for attribute initialization.
2. Write a function called **initialize** in FantasyCalculator.py to create Player objects, initialize their attributes, and store them in a list called players. Using Python code, add objects to the list for the 10 players from the table above.
3. Create a function **print_player** in Player.py that prints out the five attributes for a player, as shown in the screenshot above. Formatting the output neatly in columns can be a challenge. Hint: Examine this web site for suggestions on how to nicely format the column output: https://stackoverflow.com/questions/19103052/string-formatting-columns-in-line
4. Create a function **display_players** in FantasyCalculator.py that prints out information for all players in the list players using calls to print_player. Try to make sure the formatting looks aligned. The name of a player can sometimes be very long. Try to have formatting consistent regardless of name size.
5. The function display_players should be called when the user selects option 1 from the menu.

6. The menu and prompt should be shown again once the code for the task is finished.

We suggest that you implement and test the code for Tasks 0 and 1 and confirm they are working correctly before moving on to Task 2

Task2

For this task, you will implement code that is executed if a user selects option 2. This option enables users to see all players in the database that scored a certain number of goals.

```
Enter Command: 2
Please indicate number of goals: 10
Name                        League       Goals      Assists       Passes
Erling Haaland                 EPL          18            5          270
Robert Lewandowski          La Liga         12            6          414
Harry Kane                Bundesliga        30            6          472
Son Heung-Min                  EPL          14            8          660
Mohamed Salah                  EPL          15            9          686
Jude Bellingham             La Liga         16            3         1164
Sehrou Guirassy           Bundesliga        21            1          494
```

After selecting option 2 from the main menu, the user is prompted for the number of goals. The program then prints out all the relevant players. If no players with the appropriate count are found, the message “No players found” is printed. The main menu then reappears prompting the user for another selection.

How to proceed:

1. Write a function in FantasyCalculator.py called **display_goals**. This function will take the list of players, and the number of goals. Consider using the print_player functions you created earlier.
2. After the information is printed, the menu and prompt is shown once again.

Task3

Now try implementing option 3. In this option you will print out all players in a specific league.

```
Enter Command: 3
Please indicate the league: EPL
Name                          League        Goals      Assists       Passes
Erling Haaland                   EPL           18            5          270
Son Heung-Min                    EPL           14            8          660
Julian Alvarez                   EPL            8            8          915
Mohamed Salah                    EPL           15            9          686
```

When option 3 is selected, the user is prompted to enter a league. Follow a similar approach as option 2, create a function called **display_league** to print out all players in the specified league. Otherwise print out “No players found”.

Task4

Now try implementing option 4. In this option you will try to search for a player using their name. When option 4 is selected, the user is prompted to enter a player's name. The players list will be searched for the player with the matching name. If a match is found, the player’s details will be displayed on screen. If no match is found, “Player not found” will be displayed. You should proceed using an approach similar to the steps used to implement Tasks 2 and 3. Use a minimum amount of code in the new **display_player** function.

```
Enter Command: 4
Please indicate name of player: Robert Lewandowski
Robert Lewandowski           La Liga           12            6           414
```

Task5

Now try implementing option 5. When option 5 is selected, the function **display_most_goals** should be called and the list will be searched through and the player with the most goals will be printed.

```
Enter Command: 5
Harry Kane has the most goals with 30
```

Task6

For the last part of the project, you will determine the number of "fantasy points" for each player by determining a weighted sum of goals, assists, and passes.. Be creative and select your own formula to create a weighted sum of the three values for each player. Print out the point values to two decimal points.

```
Enter Command: 6
Erling Haaland has 69.40 points.
Robert Lewandowski has 56.28 points.
Harry Kane has 111.44 points.
Son Heung-Min has 71.20 points.
Julian Alvarez has 58.30 points.
Leroy Sane has 64.06 points.
Mohamed Salah has 76.72 points.
Jude Bellingham has 77.28 points.
Toni Kroos has 55.16 points.
Sehrou Guirassy has 74.88 points.
```

How to proceed:

1. To implement option 6, create a function **calculate_total_points.**
2. The player's point total is determined by a formula a * goals + b * assists + c *passes. You can set the values of a, b, and c to your choosing.
3. Print out the points for each player.

Suggestions

1. Get started early. Unless you already have programming experience, it will be difficult to finish this assignment in one day.
2. Work on getting Tasks 0 and 1 working properly first before moving on to implementing the remaining tasks.
3. Build up your program in small steps. Add lots of comments to inform the grader of what you are trying to do.