CP1850 – Assignment 3

Handed Out: 27 Nov 2023
Due Date: 10 Dec 2023

Objectives

This problem set will introduce you to error handling mechanisms and proper usage of dictionaries.

Collaboration

You may work with other students. However, each student should write up and hand in his or her assignment separately. Be sure to indicate with whom you have worked in the comments of your submission.

Submission

You will create one python file for each question. You need to ensure that suitable comments are present in the code. Please ensure you submit your assignment to the drop box that is provided on Brightspace **before the due date**.

Late submissions

Penalties for late submissions is as follows:

- Up to 24 hours after the due date: Flat 25% penalty
- Beyond 24 hours up to 48 hours: Flat 50% penalty
- Beyond 48 hours: No points for assignment

If for some reason you are unable to submit the assignment on time, please reach out to me to make alternate arrangements – which will be handled based on the merits of the case. **No extensions will be provided.** Please ensure you stay on top of the requirements and manage your time well.

Problem 01

Add exception handling to a Tip Calculator program.

Console

```
Tip Calculator

INPUT
Cost of meal: ten
Must be valid decimal number. Please try again.
Cost of meal: -10
Must be greater than 0. Please try again.
Cost of meal: 52.31
Tip percent:  17.5
Must be valid integer. Please try again.
Tip percent:  20

OUTPUT
Cost of meal: 52.31
Tip percent:  20%
Tip amount:   10.46
Total amount: 62.77
```

Specifications

- The program should accept decimal entries like 52.31 and 15.5 for the cost of the meal.
- The program should accept integer entries like 15, 20, and 25 for the tip percent.
- The program should validate both user entries. That way, the user can't crash the program by entering invalid data.
- The program should only accept numbers that are greater than 0.
- The program should round results to a maximum of two decimal places.

Problem 02

Create a program that allows you to view the statistics for a player of a game.

Console

```
Game Stats program

ALL PLAYERS:
Elizabeth
Joel
Mike

Enter a player name: elizabeth
Wins:   41
Losses: 3
Ties:   22

Continue? (y/n): y

Enter a player name: john
There is no player named John.

Continue? (y/n): y

Enter a player name: joel
Wins:   32
Losses: 14
Ties:   17

Continue? (y/n): y

Enter a player name: mike
Wins:   8
Losses: 19
Ties:   11

Continue? (y/n): n

Bye!
```

Specifications

- The program should use a dictionary of dictionaries to store the stats (wins, losses, and ties) for each player. You can code this dictionary of dictionaries at the beginning of the program using any names and statistics that you want. Make sure to provide stats for at least three players.
- The program should begin by displaying an alphabetical list of the names of the players.
- The program should allow the user to view the stats for the specified player.

Problem 03

Create a program for birdwatchers that stores a list of birds along with a count of the number of times each bird has been spotted.

Console

```
Bird Counter program

Enter 'x' to exit

Enter name of bird: red-tailed hawk
Enter name of bird: killdeer
Enter name of bird: snowy plover
Enter name of bird: western gull
Enter name of bird: killdeer
Enter name of bird: western gull
Enter name of bird: black-capped chickadee
Enter name of bird: x

Name                        Count
=========================== =====
Black-Capped Chickadee      1
Killdeer                    2
Red-Tailed Hawk             1
Snowy Plover                1
Western Gull                2
```

Specifications

- Use a dictionary to store the list of sighted birds and the count of the number of times each bird was sighted.
- Use the pickle module to read the dictionary from a file when the program starts and to write the dictionary to a file when the program ends. That way, the data that's entered by the user isn't lost.
- After the user finishes entering the birds that have been spotted, sort the bird names in alphabetical order before displaying the names and counts.

Problem 04

Create a program that reads a text file that contains a list of FIFA World Cup champions and determines the country that has won the most championships.

Console

```
FIFA World Cup Winners

Country          Wins  Years
=======          ====  =====
Argentina        2     1978, 1986
Brazil           5     1958, 1962, 1970, 1994, 2002
England          1     1966
France           1     1998
Germany          4     1954, 1974, 1990, 2014
Italy            4     1934, 1938, 1982, 2006
Spain            1     2010
Uruguay          2     1930, 1950
```

Specifications:

1. Use the provided text file named world_cup_champions.txt that contains data like this:

```
Year,Country,Coach,Captain
1930,Uruguay,Alberto Suppici,José Nasazzi
1934,Italy,Vittorio Pozzo,Gianpiero Combi
1938,Italy,Vittorio Pozzo,Giuseppe Meazza
...
...
2002,Brazil,Luiz Felipe Scolari,Cafu
2006,Italy,Marcello Lippi,Fabio Cannavaro
2010,Spain,Vicente del Bosque,Iker Casillas
2014,Germany,Joachim Löw,Philipp Lahm
```

2. When the program starts, it should read the text file and use a dictionary to store the required data using the name of each country that has won the World Cup as the key.
3. The program should compile the data shown above and display the countries alphabetically.

Problem 05

Create a program that allows you to view and edit the sales amounts for each month of the current year.

Console

```
Monthly Sales program

COMMAND MENU
view   - View sales for specified month
edit   - Edit sales for specified month
totals - View sales summary for year
exit   - Exit program

Command: view
Three-letter Month: jan
Sales amount for Jan is 14,317.00.

Command: edit
Three-letter Month: jan
Sales Amount: 15293
Sales amount for Jan is 15,293.00.

Command: totals
Yearly total:       67,855.00
Monthly average:     5,654.58

Command: view
Three-letter Month: july
Invalid three-letter month.

Command: exit
Bye!
```

Specifications:

1. Use the provided text file named monthly_sales.txt that consists of rows that contain three-letter abbreviations for the month and the monthly sales.
2. The program should read the file and store the sales data for each month in a dictionary with the month abbreviation as the key for each item.
3. Whenever the sales data is edited, the program should write the changed data to the text file.