CS 1160 – Introduction To Computer Programming

Lab 7 – Lists and Exceptions

Learning Objectives

- Learn how to work with lists in Python
- Learn how to work with exceptions in Python

Overview

A local YMCA needs a program that is going to keep track of all of its members and their data. To start, a program is required that will be used to input all the member's data and calculate the total amount of money earned from the active members. The data that must be stored for each member is as follows in this order:

1. First Name (string)
2. Last Name (string)
3. Age (int)
4. Account is Active (boolean)
5. Monthly Bill (float)

The program will ask for each piece of data for each member, and store the member's data into a list. Once the employee is finished entering the members into the program, the program must print out a tabulated output of all the members and their data as well as the total amount of bills collected from active users only, the average revenue from active users only, the total amount of members, and the total amount of active members.

This program must be designed with ease of use in mind. The program must utilize exception handling anywhere in the program where an exception may occur. No exceptions may occur in your program, regardless what the user types in for the member's data. If an exception occurs while testing your program, you will not get full points.

The Program

The program must loop and take various inputs for each member to be added to the program, first name, last name, age, account is active, and the monthly bill. Exception handling is required to make sure that even if the user types in an incorrect data type that the program does not crash. If the user types an incorrect value, you can discard the previously entered data for that member only and restart the loop. Finally, when the user is done inputting member's data, print out every member's data in tabular format, the total revenue from active members, average revenue from active members, the total number of members (active and inactive), and the total active members. You are not required to use functions to implement your program but it is strongly recommended.

Hint: to restart a while loop after an exception has occurred you can use the keyword "continue" to go back to the beginning of the loop.

On the following page is a sample execution including exception handling:

```
Welcome to the YMCA Database
First Name: Bob
Last Name: Smith
Age: 50
User's Account is Active: True
Monthly Bill: $banana
Error: Bill was invalid, please enter member’s data again
First Name: Bob
Last Name: Smith
Age: 50
User's Account is Active: True
Monthly Bill: $50.00
Would you like to enter another YMCA member? (y/n) y
First Name: Chuck
Last Name: Smith
Age: 40
User's Account is Active: False
Monthly Bill: $10.00
Would you like to enter another YMCA member? (y/n) y
First Name: Mark
Last Name: Smith
Age: 22
User's Account is Active: yeah
Error: Active was invalid, please enter member’s data again
First Name: Mark
Last Name: Smith
Age: 22
User's Account is Active: True
Monthly Bill: $100.00
Would you like to enter another YMCA member? (y/n) y
First Name: Marge
Last Name: Simpson
Age: 45
User's Account is Active: True
Monthly Bill: $25.00
Would you like to enter another YMCA member? (y/n) y
First Name: Homer
Last Name: Simpson
Age: fifty
Error: Age was invalid, please enter member’s data again
First Name: Homer
Last Name: Simpson
Age: 50
User's Account is Active: False
Monthly Bill: $5.00
Would you like to enter another YMCA member? (y/n) n

-------------Current YMCA Membership Data--------------
|Name:              |Age:  |Active:     |Bill:
-----------------------------------------------------
|Smith, Bob         |50    |True        |$50.00
|Smith, Chuck       |40    |False       |$10.00
|Smith, Mark        |22    |True        |$100.00
|Simpson, Marge     |45    |True        |$25.00
|Simpson, Homer     |50    |False       |$5.00

----------------------Statistics-----------------------
Total Revenue from Active Members: $175.00
Average Revenue from Active Members: $58.33
Total of Members: 5
Total of Active Members: 3
```

How to Submit
Save your .py Python program with your code and submit it to the drop box in Pilot.

Grading
This lab is worth 3.000 points, distributed as follows:

Task	Points
Successfully recorded each member's data in list: first name, last name, age, active, bill	0.750
Successfully avoided all exceptions while program is running	0.750
Successfully printed tabulated output of members	0.750
Successfully calculated and printed revenue and member statistics	0.750
Total	3.000