BYU IDAHO CSE 110 | Programming Building Blocks

10 Teach: Team Activity

Multiple Lists

Instructions

Face-to-face students will complete this activity in class.

Online students should arrange for a one hour synchronous meeting to work through the activity together.

Overview

Sometimes you have two lists that you want to keep in sync with each other. For example, in this assignment, you will track bank accounts and the balances in each one.

> **Hint from Instructor:**
>
> In future courses you will learn how to create a new custom data type (so you aren't limited to ints, floats, and strings). In that case, you can create a data type for bank accounts that stores both the name of the account and it's balance. Then, you only have one list: a list of bank accounts. This is called Programming with Classes or "Object-oriented Programming."

Assignment

I. CORE REQUIREMENTS

Work through these core requirements step-by-step to complete the program. Please don't skip ahead and do the whole thing at once, because others on your team may benefit from building the program up slowly.

01. Create two lists, one for the names of the bank accounts, and one for the balances.

    Ask the user for the name of the bank account and then for it's current balance. Keep looping until the user types "quit" for the name of an account. For each one, add the name and the balance to the appropriate list.

02. Loop through the lists using an index and display the name of the account with the balance next to it.

03. Compute and display the total balance in all of the accounts and the average balance.

The following shows the expected output:

```
Enter the names and balances of bank accounts (type: quit when done)
What is the name of this account? checking
What is the balance? 102.57
What is the name of this account? savings
What is the balance? 82.32
What is the name of this account? emergency fund
What is the balance? 200.00
What is the name of this account? quit

Account Information:
checking - $102.57
savings - $82.32
emergency fund - $200.0

Total: $384.89
Average: $128.30
```

II. STRETCH CHALLENGE

01. Determine which bank account has the highest balance and display the name and balance of that account.

02. Change your display so that it shows the index of the account next to the name.

    After displaying the list, ask the user if they want to update an account. If they respond with **yes**, ask for the index of the account, and the new balance.

03. Change the last step into a loop, so that the user can keep updating accounts until they say no. After each update, display the new list of balances.

The following shows the expected output after completing the stretch challenges:

```
Enter the names and balances of bank accounts (type: quit when done)
What is the name of this account? checking
What is the balance? 238.12
What is the name of this account? savings
What is the balance? 392.99
What is the name of this account? quit

Account Information:
0. checking - $238.12
1. savings - $392.99

Total: $631.11
Average: $315.56
Highest balance: savings - $392.99

Do you want to update an account? yes
What account index do you want to update? 0
What is the new amount? 200

Account Information:
0. checking - $200.00
1. savings - $392.99

Do you want to update an account? yes
What account index do you want to update? 1
What is the new amount? 425.50

Account Information:
0. checking - $200.00
1. savings - $425.50

Do you want to update an account? no

Account Information:
0. checking - $200.00
1. savings - $425.50
```

Sample Solution

When your program is finished, please view the sample solution for this program to compare it to your approach.

You should work to complete this team activity for the one hour period first, without looking at the sample solution. However, if you have worked on it for at least an hour and are still having problems, you may feel free to use the sample solution to help you finish your program.

- Sample solution (Core requirements): teach10_sample.py
- Sample solution (Stretch challenges): teach10_stretch_sample.py

Submission

When complete, please report your progress in the associated I-Learn quiz.

If you decided to do additional work on the program after your team activity, either by yourself or with others, feel free to include that additional work when you report on your progress in I-Learn.