

CSC401- Assignment 7

Jay Tang

Due Wednesday, June 1, 11:59pm

Reading

Read **Chapter 7** in Introduction to Computing using Python: An Application Development Focus, Second Edition by Ljubomir Perković.

Logistics

You need to do this assignment on a computer which has Python 3 installed on it.

[Python 3.10 download page can be found here.](#)

You are encouraged to work with your classmates on the assignments. If you do work with someone on the assignments, please include the name of your collaborators at the top of the file you submit. If you worked alone, please indicate that at the top of your submission. **A submission without collaboration information will not receive credit.**

A submission that includes code which does not run will not get any points for the part unless specifically documented reason of the error.

Submission

Submit the assignment using Assignment 7 folder. Save each program to a **separate file** labeled as YourName_assign7_1.py and YourName_assign7_2.py.

This assignment is due Wednesday, June 1, 11:59pm. Submissions after the deadline will be automatically rejected by the system.

Assignment

1. (40pt) Rock, paper, scissors

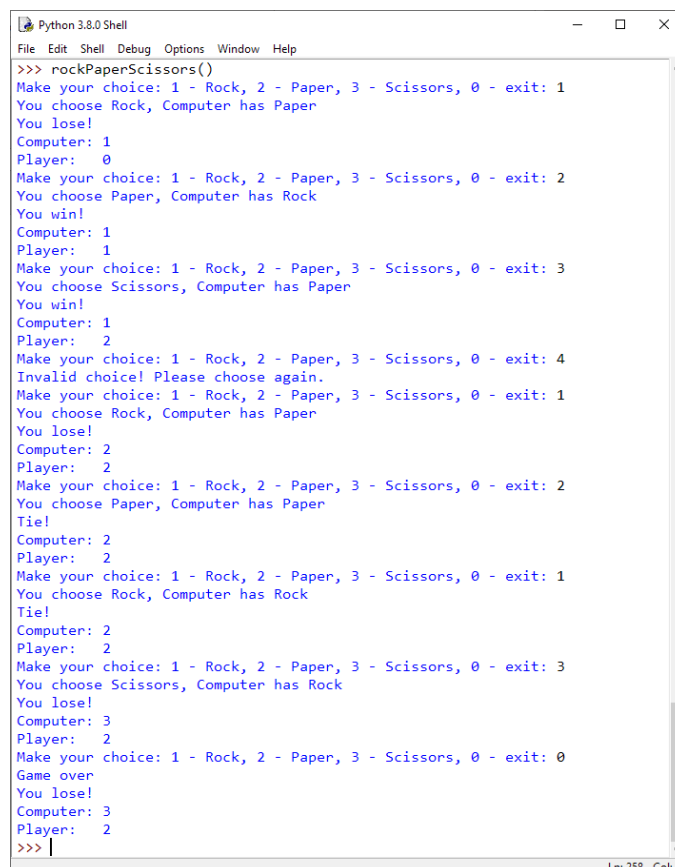
Rock paper scissors is a very simple hand game. If you don't know the rule, refer to the Wikipedia page https://en.wikipedia.org/wiki/Rock_paper_scissors.

For this problem, you will implement the game in the way you like. The player will play against the computer in a fair way. Here are some requirements for the game.

- There should be multiple rounds. You can use a parameter to choose how many rounds or let the player decide when to end.
- Result and score should be displayed at end of each round.
- It must be a fair game. Player cannot know what computer chooses and computer cannot know what player chooses.
- The game should recover gracefully from invalid input typed by the player

Please document your implementation in the docstring so people will know how to play your game when they call `help()` on your function.

Below is a sample run of the game. Your implementation may not be the same, as long as the game is playable and meets the requirements.



```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
>>> rockPaperScissors()
Make your choice: 1 - Rock, 2 - Paper, 3 - Scissors, 0 - exit: 1
You choose Rock, Computer has Paper
You lose!
Computer: 1
Player: 0
Make your choice: 1 - Rock, 2 - Paper, 3 - Scissors, 0 - exit: 2
You choose Paper, Computer has Rock
You win!
Computer: 1
Player: 1
Make your choice: 1 - Rock, 2 - Paper, 3 - Scissors, 0 - exit: 3
You choose Scissors, Computer has Paper
You win!
Computer: 1
Player: 2
Make your choice: 1 - Rock, 2 - Paper, 3 - Scissors, 0 - exit: 4
Invalid choice! Please choose again.
Make your choice: 1 - Rock, 2 - Paper, 3 - Scissors, 0 - exit: 1
You choose Rock, Computer has Paper
You lose!
Computer: 2
Player: 2
Make your choice: 1 - Rock, 2 - Paper, 3 - Scissors, 0 - exit: 2
You choose Paper, Computer has Paper
Tie!
Computer: 2
Player: 2
Make your choice: 1 - Rock, 2 - Paper, 3 - Scissors, 0 - exit: 1
You choose Rock, Computer has Rock
Tie!
Computer: 2
Player: 2
Make your choice: 1 - Rock, 2 - Paper, 3 - Scissors, 0 - exit: 3
You choose Scissors, Computer has Rock
You lose!
Computer: 3
Player: 2
Make your choice: 1 - Rock, 2 - Paper, 3 - Scissors, 0 - exit: 0
Game over
You lose!
Computer: 3
Player: 2
>>>
```

2. (60pt) ATM Machine

You are to simulate a very simple ATM machine that has a full keyboard. An account owner (user) should be able to enter their pin number and select from a menu of transactions: Deposit, Withdraw, Balance, and Quit. You are to assume that the user has only one account on which these transactions can be performed. This account is associated with the user's pin number.

The ATM should properly and regularly communicate with the user. The ATM should get information on current account from a file (accounts.csv). This file is posted with the assignment in the submission folder. Each line of the file contains a 4-digit code (pin), first name, last name, and the account balance. Contents of accounts.csv file follows:

```
9967,Harry,Potter,10.05
4466,Kay,Winthrop,356
2813,Mickey,Mouse,125
2245,Donald,Duck,1235456.5
2315,Daffy,Duck,234
1234,Hermoine,Grainger,23000
6345,Darth,Vader,98745.35
8073,Nicholas ,Nickleby,23750
3381,Tom,Sawyer,12
4455,Jane,Austen,78000
9900,Mary,Poppins,2345000
7465,Ron,Weasley,3579.83
7395,James,Bond,33579
7643,Walter,Mitty,2132106.12
3221,George,Washington,235120.05
8569,Benjamin,Franklin,55000.55
3967,Alexander ,Hamilton,45490.95
```

You are to use the `main()` function code in your program included in `atm.py` file. This function controls the flow of the program. You may not change the names of the functions or the arguments that are passed or modify `main()` in any way. You must use only local variables (i.e. global variables are not allowed). You can find the `main()` function code in `atm.py` file attached.

Steps

1. (10pts) Write a function `startUp(filename)` that takes as parameter a filename that contains the account owner (user) information and current account balance.
 - Initialize the dictionary in the function. All variables in this program must be local variables.

- If the filename is invalid, handle the error, print 'Cannot get to the file' and return None.
 - If the filename is valid, read the file and store each line in the dictionary.
 - i. The key is the 4-digit code (pin), and the value is a list with the user first name, last name and balance. Balance is a float. Do an explicit conversion to float on that list item.
 - ii. The function **returns the dictionary** if the file was successfully read and the dictionary successfully filled with the file data; if the file is not successfully opened/read, **return None**.
2. (5pts) Write a function `verifyPin(d)` that takes as parameter the dictionary.
- Prompt the user to enter a pin number.
 - If the pin is valid, that is, it is 4-digits and is in the dictionary, **return pin and name** where pin is the dictionary key and name is the user's first name. If the pin is invalid, print 'Incorrect pin' and **return (None, None)**.
3. (5pts) Write a function `menu(name)` that takes as parameter the user's first name, displays the user's name and the menu options:
- D: Deposit
 - W: Withdraw
 - B: Balance
 - Q: Quit
- If the user enters a value that is not D, W, B, Q, then print the message 'Valid choices are D, W, B, Q, try again' and display the menu options again. When a valid number is entered, the function should **return the letter of the chosen option**.
4. (10pts) Write a function `getAmount()` that takes no parameters.
- Inside a loop, prompt the user for an amount to be deposited or withdrawn; at this point it does not matter which it is. The amount must be converted to float. If the amount is negative, print 'Negative amount. Please try again'.
 - If user gives an invalid input, handle the error and print 'Invalid amount. Use digits only'.
 - Stay in the loop until a valid number is entered and **return that amount**.
5. (5pts) Write a function `deposit(pin, d)` that takes as parameters the user's pin number and the dictionary. The function calls `getAmount()`, calculates the new balance, updates the dictionary and prints a message of success. The function does not return a value.
6. (10pts) Write a function `withdraw(pin, d)` that takes as parameters the user's pin number and the dictionary. The function calls `getAmount()`. If the amount to be withdrawn is greater than the balance print 'Insufficient

funds to complete the transaction'. The user should be prompted to enter a new amount, until the amount entered is less than or equal to the balance. The function uses the balance in the dictionary to calculate the new balance. The balance is updated in the dictionary. This function does not return a value.

7. (5pts) Write a function `balance(pin, d)` that takes as parameters the user's pin number and dictionary and **returns the user's balance from the dictionary**.
8. (5pts) Write a function `quit(pin, d)` that takes as parameters the user's pin number and dictionary. The user is prompted as to whether or not she wishes to leave the transaction. If she does, then the function **returns the user's first and last name**. If she does not want to leave the application, then **return (None, None)**.