DC-322-L-1 Web Engineering Lab

Lab 02 BSSE F20 Nov 03, 2022

The objective of this lab is to

- Learn file handling in python (reading, writing & appending data to a file).
- Learn exception handling & also create your custom exceptions as per the requirement of your custom program.

Instructions!

- Keep your student identity cards with you.
- This is an individual lab, you are strictly **NOT** allowed to discuss your solutions with your fellow colleagues, even not allowed to ask how is he/she is doing, it may result in a negative marking.
- You can **ONLY** discuss this with TAs or Ma'am.
- Save your work frequently. Make a habit of pressing CTRL+S after every line of code you write.
- This is a **GRADED** lab, so, at the end of the lab session, you should have your complete work ready for evaluation.
- Follow proper coding conventions and write comments.
- You're getting your lab as a hard-copy, no interaction with internet is required, so using internet during lab is **PROHIBITED**, just use you locally setup **IDE.**
- ***Total Time for this Lab is 90 minutes.***

Task 01 [20 Marks - 90 minutes]

ATM System in python

In this task, you'll be designing a python menu based ATM system. Take a real life ATM, for that you first register or setup a pin, then while transacting, you give the already registered card and pin/password for processing a transaction.

For this program, use the same approach, as this lab is based on file and exception handling, so you need to write data (after registration) and read data (while login or checking if the user already exists). As you have now studied the Exception Handling too, there'll be several cases you can utilize it for, like while taking user input, reading or writing a file etc.

The grading and focus of this program will be based on *how you used the exceptions, files and the control and flow of the program.*

Below are mentioned the details of each major component.

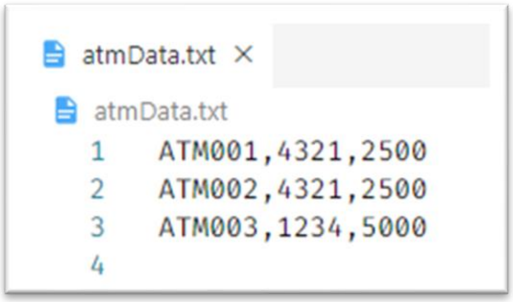
1. **Exceptions**
   - Create a custom exception class ***ATMException,*** which handles:
     - ***AccountNotFound*** (while login, if user enters wrong account number).
     - ***InsufficientBalance*** (While withdrawing money more than balance).

        - ***InvalidPin*** *(While setting up pin, it should be exactly 4 digit numerical value, other than that, raise this exception).*
- *For all other exceptions like negative numbers, invalid format of pin etc. you can use python's built-in exceptions.*

2. Register an account

- A User can register his/her account, with *account number and password.*
    - **account numbers** will be automated (assigned by your program), think of like a sequence with some prefix like ATM001, ATM002 etc.)
    - **password** should be exactly 4 digits numerical number.
    - **Account balance** keep the default opening balance 100 (no need to take it from user).
    If you want to add any other properties like username etc. You can do it.

- After successful registration, the data should be **written in a file,**
  input file format should be:
  <account number>, <password>, <balance>



```
atmData.txt
ATM001,4321,2500
ATM002,4321,2500
ATM003,1234,5000
```

-

3. Login

- Once a user is registered, one can login with his/her account number and password. After login, user can perform 3 functionalities
    - **Check his/her balance.**
        - Just print the balance in one's account.
    - **Withdraw**
        - Take the amount from the user.
        - Handle exception if the user enters negative amount or amount more than his/her balance.
    - **Deposit**
        - Take the amount user wants to deposit.
        - Handle exception if the user enters negative amount.

NOTE: Your Program shouldn't struck while taking user input no matter what a user enters, file not found error, user already exists etc., all these exceptions and errors must be handled carefully.