SEHIR BANK

In this mini project, you are going to develop a text-based simple banking application. **The primary goal of this project is to make you practice functions, loops, conditional execution, and getting user input through keyboard**. The detailed explanations about the game that you are going to develop are provided below (Note: example console output from the application is written in **BLUE**.).

Project Logic:

The application's idea is for the customer to be able to interact with his bank account in the easiest way possible. The main features you will develop in this mini project are as follow: withdraw and deposit money, Transfer Money to other accounts, and display the users' account Information.

Text-Based Bank app:

Your bank will only have 2 customers:

First user's User name should be Ahmet and his password should be 1234

Second user's User name should be Zeynep and her password should be 4321

When your program first starts, the users will be asked the service they want. Meaning, they will have 2 options. That is, either to log in or exit the application. If the chose login, your application should ask the users to enter their usernames and passwords. If a user enters a wrong username or password, then your application should warn the user and ask for a valid username/password again, until the user provides the correct ones. Below is a sample of how your welcome page should look like:

```
--- Welcome to SEHIR Bank V.0.1 ---

1. Login

2. Exit
```

If the user chooses to log in, your app should display the following:

```
1. Login

2. Exit

>>> 1

User Name: Ahmet

Password: 1234
```

Your program should keep asking till the user gives the correct credentials. Once your application gets the correct credentials, your program should again welcome the user but this time with his name clearly displayed. Your application should also display the list of the available services. Below is a sample of how this should look like:

```
Welcome  Ahmet!

 Please enter the number of the service:

1. Withdraw Money

2. Deposit Money

3. Transfer Money

4. My Account Information

5. Logout
```

Your user will have 4 services and a logout option. Your program should print the menu numbered and the users should be able to enter the number corresponding to the service they want to use.

If option 1 is picked, your program should ask the user to provide an amount of money to withdraw from their account as follows:

```
Please enter the amount you want to withdraw: 90
```

However, your program *should check whether or not the user's account has enough money to do that transaction*. If not, your program should warn the user that he doesn't have enough money and go back to the main menu.

```
You don't have 90 TL in your account

Going back to main menu...
```

If the user has enough money to do the transaction required, then your program should inform the user that the amount he requested has been withdrawn from his account as follow:

```
Please enter the amount you want to withdraw: 700

700 TL withdrawn from your account

Going back to main menu...
```

If option 2 is picked, your program should ask the user to provide an amount of money to deposit to their account. After the deposit is successful, your program should inform the user that the amount he entered has been deposited into his account. Your program should go back to the main menu after the deposit is complete.

```
Please enter the amount you want to drop: 80

80 TL added to your account

Going back to main menu...
```

If option 3 is picked, your program should ask the user to enter an amount of money to be transferred as follows:

```
Please enter the amount you want to transfer:
```

You application should check if the user has enough money in his bank account to transfer the amount he entered.
If he doesn't have enough money, your application should warn the user as follows:

```
Please enter the amount you want to transfer: 40

Sorry! You don't have enough money to complete this transaction
```

Your program should ask the user what he wants to do. He can either can back to main menu or try to enter another amount to be transferred. A sample of your code running should be as follows:

```
1. Go back to main menu

2. Transfer again

>>>
```

If option 4 is picked, My Account Information, your application should display the name of the bank, the current date and time. The user's name, password, and how much money he has in his account. A sample of your code running should be as follows:

```
------- SEHIR Bank -------

----2019-03-03 13:43:18----

----------------------------

Your Name:  Ahmet

Your Password:  1234

Your Amount (TL):  0
```

If option 5 is picked, your program should logout but keep the users' account balance saved so if he logs in again, his account balance is correct. *For example, if the user logs in, deposits 60 TL and then logs out. When he logs in again and checks his account information, his account balance should display 60 TL (Providing that he did not withdraw or deposited any other money.)*

Implementation Notes:

- You may check this link to get further information about Time module. (https://docs.python.org/3/library/time.html)
- **You MUST use functions**. You are strongly advised to use a function for each service. However, you have to use at ***least 3 functions. Your grade will be submitted to a 10% reduction if you do not create 3 functions.***
- Your code should be efficient, easy to follow, and track.

Warnings:

- If you don't follow the implementation notes, you are very likely to lose points.
- **Do not** talk to your classmates on project topics when you are implementing your projects (This is serious). **Do not** show or email your code to others (This is even more serious). If you need help, talk to your TAs or the instructor, not to your classmates. If somebody asks you for help, explain them the lecture slides, but do not explain any project related topic or solution. Any similarity in your source codes will have **serious** consequences for both parties.
- Carefully read the project document, and pay special attention to sentences that involve "**should**", "**should not**", "**do not**", and other underlined/bold font statements.
- If you use code from a resource (web site, book, etc.), make sure that you reference those resource at the top of your source code file in the form of comments. You should

give details of which part of your code is from what resource. Failing to do so **may result in** plagiarism investigation.

- Even if you work as a group of two students, each member of the team should know every line of the code well. Hence, it is **important** to understand all the details in your submitted code.

How and when do I submit my project? :

- Projects may be done individually or as a small group of two students (doing it individually is recommended for best learning experience). If you are doing it as a group, only **one** of the members should submit the project. File name will tell us group members (Please see the next item for details).
- Submit your own code in a **single** Python file. Name your code file with your and your partner's first and last names (see below for naming):
    - If your team members are Deniz Barış and Ahmet Çalışkan, then name your code file as deniz_baris_ahmet_caliskan.py (Do **not** use any Turkish characters in file name). If you are doing the project alone, then name it with your name and last name similar to the above naming scheme. Those who **do not** follow the above naming conventions **will get 5 pts off** of their grade.
- Submit it online on LMS by **5 pm on Friday, Dec. 22, 2017**

Late Submission Policy:

- -10%: Submissions between 17:01 – 18:00 on the due date
- -20%: Submissions between 18:01 – midnight (00:00) on the due date
- -30%: Submissions which are 24 hour late.
- -50%: Submissions which are 48 hours late.
- Submission more than 48 hours late will not be accepted

Grading Criteria? :

CODE ORGANIZATION			FUNCTIONALLITY				
Meaningful variable names (4 pts)	Proper use of functions, compact code with no unnecessary repetitions (4 pts)	Sufficient commenting (2 pts)	LOGIN FUNCTION (15 pts)	DEPOSITING AND WITHDRAWING MONEY (25)	TRANSFERING (10 pts)	LOGOUT (15)	BANK BALANCE KEEPING (25)

Have further questions?:

Please contact your TAs if you have further questions. If you need help with anything, please use the office hours of your TAs and the instructor to get help. Do not walk in randomly (especially on the last day) into your TAs' or the instructor's offices. Make an appointment first. This is important. Your TAs have other responsibilities. Please respect their personal schedules!

Good Luck!