CSC401- Assignment 8

Jay Tang

Due Wednesday, June 8, 11:59pm

Reading

Read **Chapter 8** 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 8 folder. Save each program to a **separate file** labeled as YourName_assign8_1.py and YourName_assign8_2.py.

This assignment is due Wednesday, June 8, 11:59pm. Submissions after the deadline will be automatically rejected by the system.

Assignment

1. ATM Machine 2.0 (50pts)

In the previous assignment, you implemented a simulation of a simple ATM machine. You used a list to store each customers' account information – this is not a good practice. Now we have learned how to define our own class, it's time to give our ATM machine an upgrade.

Develop a class `Account` to represent an account at the bank. The account needs to support the following methods:

- `Account(fname, lname, balance)`: Constructor that initialize name and balance of the account
- `getFName()`: Returns the account owner's first name
- `getLName()`: Returns the account owner's last name
- `getBalance()`: Returns the account's balance
- `deposit(amount)`: deposit amount into account
- `withdraw(amount)`: withdraw amount from account

You are tasked to upgrade the programming to use the `Account` class instead of the list without affecting its functionality. The program should behave exactly the same as before.

If you had trouble finishing Assignment 7, you can use the solution as the start point of this assignment.

2. Blackjack (50pts)

Before starting your work on this problem, please read Case Study CS.6 Game of Chance in the textbook. It goes through the rules of blackjack and also the implementation of the code. You can find the complete code in blackjack.py in the submission folder.

As you can see, the implementation does not follow object-oriented programming principles (because this case study is for chapter 6, prior to the introduction of OOP). So we are going to do some improvement similar to the first problem.

Develop two classes: `Card` and `Deck`. You will need to figure out what methods are needed in these two classes. You can use the `Card` and `Deck` classes we discussed in this week's lecture as a start point.

Once you are done with the two classes, you need to modify the rest part of the program so it will work with the new implementation. Take the errors as your friend, not your enemy, and resolve those errors one by one. Remember debugging your code is a huge part of the development.

There are lots of code examples of blackjack you can find online. **DO NOT ATTEMPT TO USE THEM**. You should only modify some small parts of the program after you finish the two classes. Any resemblance to the solutions online will result in a 0 for this assignment.