Lance Brown

Approach Document

4/30/20

1. The account classes were straightforward. You have a parent Account class that is inherited by Savings and Checking. All the properties are the same, as are some of the methods. They each just had to override the withdraw or deposit methods to account for different restrictions on balances/transfer limits.

I went back and forth on whether to have a customer class, since the customer has very little function in this context. It would share most of its properties with Account since accounts are defined by the customer name and social, but wouldn’t really have any methods. The only way I could justify it was to have the customer store the accounts it created, but it seemed much simpler to have the accounts store the customer instead, since they already have all the customer information. I decided to use a dictionary class variable to do so because each new account can be added as a value to the customer’s SSN as a key which can be used as a unique identifier for lookup. As I’m writing this I’m second guessing myself again, but I stand by my decision to not make Customer a class because it is mostly duplicate code and it does not make sense for Account to inherit Customer to reduce redundancy (every new account would make a new customer).

The complicating factor is the requirement that the total balance for all accounts for a customer can be calculated. Were that not required, there would be no doubt that a Customer class is extraneous, and conversely if there were more than one potential use it would clearly be necessary.

1. I encapsulated data via the property method baked into Python. Data can be set via setters and retrieved via getters. There are no direct calls to data once instantiated.
2. Savings and Checking inherit all properties from Account. Savings overrides both deposit and withdraw methods to account for minimum balance and deposit amount, Checking overrides withdraw to append an overdraft fee on any transaction that brings it into the negative. They both inherit the validateMoney, transfer and checkBalance methods.
3. Savings and Checking both override methods from the parent Account class. Savings overrides both deposit and withdraw methods to account for minimum balance and deposit amount, Checking overrides withdraw to append an overdraft fee on any transaction that brings it into the negative. Account overrides the \_\_str\_\_ method solely for flavor.