Name: SGT Schoenwald-Oberbeck, Jesse
Date: 11MAR2017
Current Module: Object Oriented Programming
Project Name: Bank of Nerds

Project Goals:

Build a simulated bank user interface.

Considerations:

o Most elements of this project's design would be best build as objects.

Initial Design:

bank_of_nerds_lib.py will contain all code except the main function/launcher, which will be in bank_of_nerds.py

Data Flow:

Various menus will provide user interaction, guiding the user through customer creation, creation of various accounts, as well as deposits and withdrawals. The bank object will be a container for customer objects. Customer objects will contain name, age, and the customer's accounts. Each created customer object will also create and contain its own ID number, for easy reference by the customer. The Account class is a parent class, never used on its own. It serves as a base for the different account types, and has a variable for money, as well as the base versions of the deposit and withdraw functions. Each account subtype adds it's own "type" variable to be checked in various places, primarily for withdrawal restrictions. Some of the subtypes of Account overwrite the withdraw method to add further checks.

Communication Protocol:

No networking was required, and none were implemented or used.

Potential Pitfalls:

o Construction should be relatively straightforward.

Test Plan:

User Test:

Multiple runs of the program, using every variation/combination of options and inputs the player can think of.

Test Cases:

All test cases completed with correct output.

Conclusion:

The design of the bank interface, in order to be built most effectively for the present and for potential future adjustment, was planned out using class objects, each containing a simple set of properties needed for the object type. Once the objects were built, all that remained was program flow, menus, and functions for actions such as withdrawal and deposit that provide human interaction, before calling the actual deposit and withdraw methods from the classes that adjust the money the objects contain.