

## CS 6651 – 03 Intro to Scripting/Python – Assignment 4

In this assignment, you will write a program that performs basic banking transactions using Object Oriented Programming approach.

**Due: April 22, 2020 – 11:59 pm**

### Requirements

1. This program should have three classes
  - a. Parent Class
    - i. BankAccount
  - b. Derived Classes
    - i. Savings
    - ii. Checking
2. When you run this program:
  - a. User should be asked
    - i. Name
    - ii. Account Type
  - b. If account type is savings, object of Savings subclass should be created.
  - c. If account type is checking, object of Checking subclass should be created.
  - d. As soon as object is created, name, account type and account balance (initialized as 0) should be displayed on console.
  - e. After object is created, user should be asked either to deposit or withdraw or exit the program
  - f. When user chooses to deposit:
    - i. If account type is savings:
      1. User should only be allowed to deposit amount more or equal to 500.
    - ii. If account type is checking:
      1. User should be allowed to deposit any amount more than 0.
    - iii. For both account types, deposit amount should be added to current balance.
  - g. When user chooses to withdraw:
    - i. If account type is savings:
      1. User should only be allowed to withdraw amount less than or equal to 500 and amount less than **or equal to** the account balance.
    - ii. If account type is checking:
      1. User should be allowed to withdraw any amount more than 0 but remaining account balance should not be less than 250 after withdrawal.
    - iii. For both account types, withdrawal amount should be subtracted from current balance.
  - h. Both deposit and withdraw amounts should be asked from user and respective error messages should displayed for all invalid inputs and user should be asked to enter amount again.
  - i. After all deposits and withdrawals, name, account type and account balance after each transaction should be displayed on user console and, user should be asked to deposit or withdraw or exit the program.
  - j. When user chooses to exit, name, account type and account balance should be displayed, and program should be terminated with a program termination message.

**Output 1:**

1. Run the program
2. Enter full name of the first user
3. Choose Savings account type (after this step, object of Savings subclass should be created, and account information should be displayed)
4. Choose deposit
5. Enter 300 (this should display an error message and ask user for amount again)
6. Enter 700 (this should update account balance and display account information)
7. Choose to withdraw
8. Enter 600 (this should display an error message and ask user for amount again)
9. Enter 300 (this should update account balance and display account information)
10. Choose to withdraw
11. Enter 450 (this should display an error message and ask user for amount again)
12. Enter 400 (this should update account balance and display account information)
13. Choose exit (this should display account information and terminate the program)
14. Copy all the console output and save it to output1.txt file.

**Output 2:**

1. Run the program
2. Enter full name of the first user
3. Choose Checking account type (after this step, object of Checking subclass should be created, and account information should be displayed)
4. Choose deposit
5. Enter negative 300 (this should display an error message and ask user for amount again)
6. Enter 700 (this should update account balance and display account information)
7. Choose to withdraw
8. Enter 600 (this should display an error message and ask user for amount again)
9. Enter 300 (this should update account balance and display account information)
10. Choose to withdraw
11. Enter 400 (this should display an error message and ask user for amount again)
12. Enter 150 (this should update account balance and display account information)
13. Choose exit (this should display account information and terminate the program)
14. Copy all the console output and save it to output2.txt file.

**Submission:**

You will need to submit a zip file on BlackBoard which contains source code (.py) file and both output text files.