**Aritha Mindula Jayaratne (s4683873)**

**Victoria University  SUBJECT: INT5150**

**REPORT: Banking SYSTEM (object oriented programming).**

**Modeling and Analysis of the Banking system.**

A simple banking system has been developed using object-oriented programming principles. The code has several OOP concepts that has been used to compile the banking system altogether. This will be further discussed in the later part of the report how these concepts have been used to compile the code. The code has one super class as (‘Account’) and from the super class there are two sub classes as ‘Currentaccount’ & ‘DepositAccount’. Two more accounts has been created using ‘Currentaccount’ as ‘RestrictedAccount’ & ‘AccountWithOverdraft’. Another class was created as ‘Bank’ to manage the multiple accounts in the banking system and this class will perform various functions as shown below.

**Super class (‘Account’):**

Serves as the base or the super class for all the account types mentioned. This has attributes like ‘balance’ and ‘account\_number’. This includes two functions as ‘withdraw’ and ‘deposit’ to withdraw and deposit money into the account. These two functions do the basic operations to ensure the validity of the transactions.

**Sub classes (‘Specialized Accounts’):**

* ‘CurrentAccount’: This class inherits from the ‘Account’ class and all of its attributes and also adds a new attribute as ‘cheque\_count’ to represent and validate the available cheques in the account.
* ‘DepositAccount’: This class also inherits from the ‘Account’ class and all its attributes. New attributes are added to perform certain attributes as ‘withdrawal\_limit’ and ‘interest\_rate’ to calculate an interest fee for the deposited money.
* ‘RestrictedAccount’: This class inherits certain features from the ‘CurrentAccount’ and also adds a new attribute ‘max\_withdrawal’ to limit the amount of money that could be withdrawn from the account. It acts as barrier limit the user from emptying the restricted account.
* ‘AccountWithOverdraft’: This class inherits from ‘CurrentAccount’ and has all its attributes and adds a new attribute as ‘overdraft\_limit’ which allows the user to access overdrafts to a certain limit in the account.

**Bank class (‘Bank’):**

This class will be the main user interface for the users, and this will be managing all the other mentioned accounts and it will be consisting of several methods which will help the banking system function.

‘add\_account’: This function will add an account to the banks list of accounts.

‘withdraw\_from\_account’: This function will withdraw money from a defined account from the four specialized accounts.

‘deposit\_to\_account’: This function will allow the user to deposit cash into any of the specialized account.

‘save\_accounts\_to\_file’: This function will save all the entered account information to the file.

‘load\_account\_from\_file’: This function will help to load any of the saved account information to a certain file.

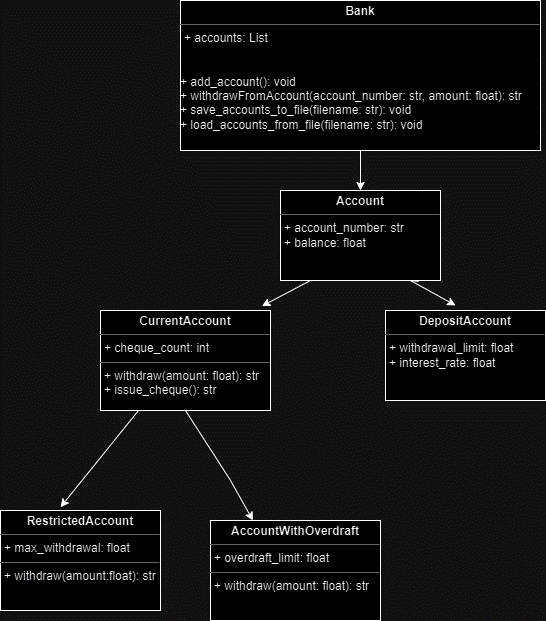
The model of the banking system showcases some of the object oriented programming concepts as mentioned below.

**Inheritance: inheritance is used when creating the specialized account types. These accounts inherited attributes from the super class. Due to inheritance it allowed to reuse the code and create different objects with specialized functions.**

**Polymorphism: In this code polymorphism can be seen by method overriding concept. Each account has different ways how the withdraw method has been used based on their type. For example, current account and deposit account has different withdraw functions.**

**Encapsulation: attributes such as balance, account\_number, cheque\_count, withdrawal\_limit are encapsulate in their main classes and these are accessed using methods. This will help to maintain data integrity and control the use of class attributes.**

**UML Diagram.**



**Testing the Program.**

Test case scenario:

Withdrawing from the current account an amount of 7000. Current account has only 6000. The output of the scenario should be displayed as ‘’insufficient funds”.

Outcome

A computer screen shot of a code

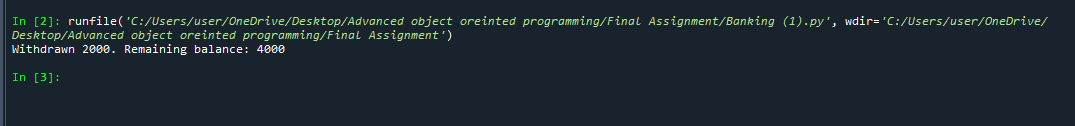
Description automatically generated

Status: Pass

Test case scenario:

Withdrawing 2000 from the current account. The output of the scenario will be displayed as “Withdrawn 2000, Remaining balance: 4000”.

Outcome

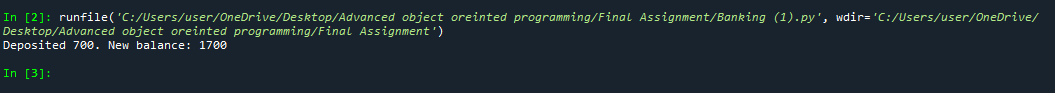


Status: Pass

Test case scenario:

Depositing 700 to the current account. The output of the scenario will be displayed as “Deposited 700, New balance: 1700“.

Outcome



Status: Pass

Test case scenario:

Depositing 0 to the current account. The output of the scenario will be displayed as “Invalid deposit amount”.

Outcome

A screen shot of a computer

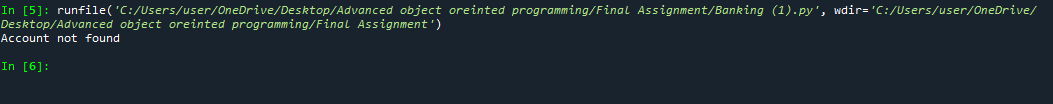
Description automatically generated

Status: Pass

Test case scenario:

Deposit 1000 to the current account but the account ID is wrong. The output of the scenario will be displayed as “Account not found”.

Outcome

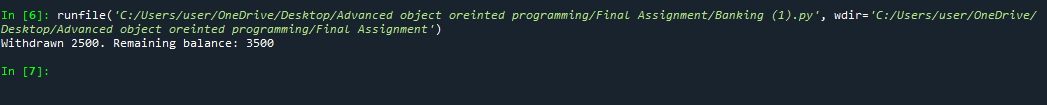


Status: Pass

Test case scenario:

Withdrawing 2500 from the Deposit account. The output of the scenario will be displayed as “Withdrawn 2500, Remaining balance: 3500”.

Outcome

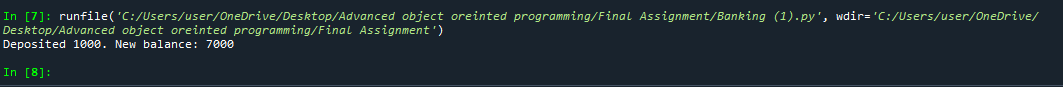


Status: Pass

Test case scenario:

Deposit 1000 in the deposit account. The output of the scenario will be displayed as “Deposited 1000. New balance: 7000’’.

Outcome

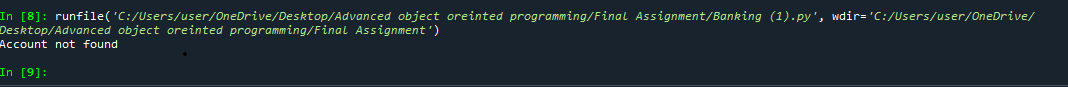


Status: Pass

Test case scenario:

Deposit 1000 into the deposit account but the given account ID is wrong. The output of the scenario will be displayed as “Account not found”.

Outcome

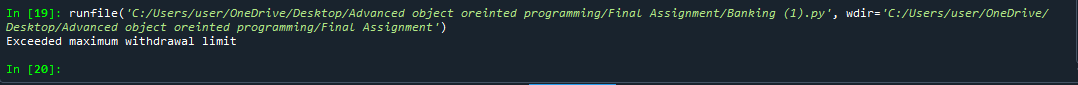


Status: Pass

Test case scenario:

Withdraw 5000 from the restricted account which has a maximum withdrawal limit of 500. The output of the scenario will be displayed as “Exceeded maximum withdrawal limit”.

Outcome

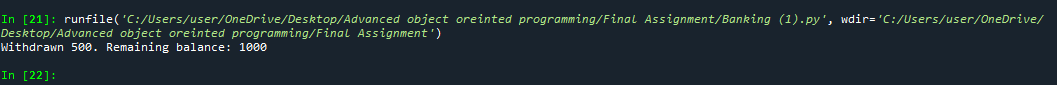


Status: Pass

Test case scenario:

Withdraw 5000 from the restricted account which has a maximum withdrawal limit of 500. The output of the scenario will be displayed as “Withdrawn 500. Remaining balance: 1000”.

Outcome



Status: Pass

Test case scenario:

Deposit 2000 to the Restricted account. The output of the scenario will be displayed as “Deposited 2000. New balance: 3500”.

Outcome

A blue background with yellow text

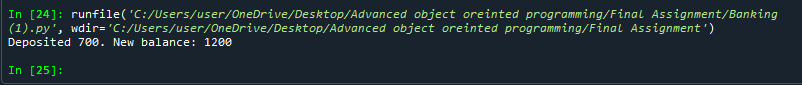
Description automatically generated

Status: Pass

Test case scenario:

Deposit 700 in the Overdraft account and the output will be displayed as “Deposited 700. New balance: 1200”.

Outcome

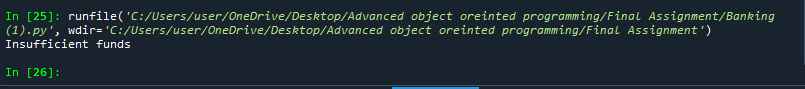


Status: Pass

Test case scenario:

Withdraw 800 from the overdraft account and the output of this scenario will be “Insufficient funds”.

Outcome



Status: Pass

**References.**

* Singh, N., Chouhan, S.S. and Verma, K. (2021) ‘Object Oriented Programming: Concepts, Limitations and Application Trends’, *2021 5th International Conference on Information Systems and Computer Networks (ISCON), Information Systems and Computer Networks (ISCON), 2021 5th International Conference on*, pp. 1–4. doi:10.1109/ISCON52037.2021.9702463.
* Wu, Z., Liu, J. and Shi, S. (2022) ‘Research on Programming of 3D Simulation System under the Control of Computer Python Language’, *2022 IEEE 5th International Conference on Information Systems and Computer Aided Education (ICISCAE), Information Systems and Computer Aided Education (ICISCAE), 2022 IEEE 5th International Conference on*, pp. 918–922. doi:10.1109/ICISCAE55891.2022.9927507.
* Fabrizio Romano, Benjamin Baka and Dusty Phillips (2019) *Getting Started with Python : Understand Key Data Structures and Use Python in Object-oriented Programming*. Birmingham, UK: Packt Publishing. Available at: https://research.ebsco.com/linkprocessor/plink?id=f48ed05a-d753-3518-9d90-b8140fd1ea8c (Accessed: 18 September 2023).