# Object Oriented Programming

# Continuous Assessment: Bank Management System

**by Ayan Abedin(D19125792)**

**Raghav Bansal(D20123625)**

**Description**: We developed an application to manage bank services, A Bank in general, its Account classes and function for services.

Program is not a loop program that show menu again and again, User must run the program again after creating or login into any account. On the entire code, **IBAN** is used as unique identifying value.

Let us go in details one by one.

**Project Classes and Function**

Our code starts with the **class Bank** that create and manage the main account, We have a function in a class that is \_\_init\_\_(), we used this function to assign values to object properties, or other operations that are necessary to do when the object is being created. We write self, that is bound the default instance as it being made. Then we needed two account Saving account and checking account, as a subclasses, so we create classes, like **class Saving(), and Checking(), which inherit property from parent class.**

**Functions: -** In each class there are **functions.** It starts with **function register** that will be called when customer create an account, it contains all the validation like IBAN should be 10 digit number, PIN should be between 4 to 6, and then age that is most important for registration, because Saving account can be opened by teenager i.e. 14 years old, and Checking account can be opened by legal age i.e. 18 years old. Then we create a **function login**, when Customer wants to login after registering a valid account. When user register or create an account there will be separate file stored, in login function we are opening and reading the file information and that should be match. For that we used if and else statement for matching **client\_details\_list** with file stored.

Then there is **function add\_cash() (or deposit)** our default amount in all the account type is **€100** that we did in staring of code by **self.cash,** here when amount is greater than 0, **self.cash += amount** that will increment the current amount with deposited amount. And it will be modified in the stored file account as well. Then for withdrawing cash, we just decrement the amount by **self.cash -= amount**.

Next function is **Transfer\_cash(),** we all this function when customer want totransfer amount, first it will ask for amount and then name of account and IBAN, if account exists, amount will be transferred and by the file handling write(w) it will be written in account file as well.

After that **function pin\_change()** if pin is between 4 and 6 main account customers can edit it later after logging in.

Following that, Last function to call is **delete\_account()**, to remove a file we used **‘remove’ function** from the **os** module (*import os*), that why we are importing **os.**

In the main function, in the menu all the function called. We start with initialize Classes to variable, and then creating menu. Then the user can do operations as they want.

**Creativity idea**

Our creativity in this is to create a proper bank system with a security of IBAN and pin, and some make it look good and very easy to understand the bank, we want to make it user-friendly program, if Customer is visiting in our bank, they should be guided properly, so created a menu that shows all the create bank account option, after that all other bank services are provide in those bank account. We made file for each customer information, and then only login option, when customer want to login, they should enter correct IBAN and pin. And customers want to change pin they can change.

**User manual**

This project file contains a python script (CA2.py). This is a simple python console-based program which is very easy to understand and use. Code starts with branding name that will bring our menu option, that includes create and login into main account, create and login Saving account, create and login Checking account. Before using an account service, Customer must create that account. Then customer can login into the required account and use services. Customer can enjoy all the services in main account, after logging into main account, customer can **Add amount**, **Check Main account balance**, **Withdraw an Amount**, **transfer an amount to any account (Saving or Checking),** **Edit account information**, **Close an Account**, and then simply **logout**. For the main account, there is no restriction on adding an amount or withdrawing or transferring into other account. **Saving Account & Checking Account customer can only transfer money to only savings account and checking accounts respectively**. Before transferring amount, Customer must **write exactly same name as customer create other bank account name**, for example, there is main account named by John, and then John creates a Saving account, in **file handling** that file will be John\_Savings.txt, so for transferring user must enter full name with account type**. In transfer, Customer can also see transaction history, Customer can see amount deducted and left balance in the account and amount transferred as well.** In this code. If customer create any account**, a .txt file will be created**, that will contain all the information of Customer, and it **get updated** when customer use bank service. Saving account can be opened by teenager from 14 years old, on the other hand checking account can be opened by only customers who are 18 years old. In the **checking account Customer** can have a **negative credit limit** as well. For a unique ID, all customers have to login with IBAN & PIN, that should be same when it is created, Main account **customer can change their pin**. When Customer use all services, he/she can choose option for logout. All the operations performed in the account is reflected in their particular file. Please refer to <.txt> files to notice the change.

**Team Members Individual Contribution**

This program was too hard to write but easy to think, we both first start with the code structure, we wrote all the requirement we need, we divided thing in the code, this code is actually written by both of us, we use repl.it for editing the code, we both figure out with class Bank simultaneously, then Ayan created Saving Account and Raghav created Checking account. For the functions in code, Ayan figured out about the login and registering of account and withdrawing an amount, Raghav wrote functions for Deletion account, and add cash(or deposit money) for main account. Tranfer\_Cash function is done by both of us, Raghav worked on file handling, Ayan worked on the code, Main menu half written by Raghav and Ayan.

For our creativity, we came up with many ideas, but it was implemented some of them, Ayan got idea for menu look and he implemented in it, Raghav wrote code for other services like add, withdraw, transfer, logout. We both worked together most of the code, it’s very hard to define every single line of code in contribution. If Raghav get stuck Ayan helped with his error, Vice-a-versa.

**Difficulties and more challenging parts**

* **Subclasses and Inheritance**: It was easy to think to understand the concept of subclasses, but we find it too hard to apply it, firstly we made Bank as our parent class, we wrote the sub account information as a function, but after completing the whole code, we tried to put Saving and Checking into subclass of Bank.

We find it too hard, then we changed the values and data as well.

* **File handling**: As per project requirements, bank information should be stored in the three external files: customers.txt, accounts.txt, accountsTransaction.txt. But we could not figure out that part of code, instead of leaving it, we wrote something that open a file with all information of customer with name, IBAN, pin, age, and Amount. So now, when customer create an account, there is separate file stored in folder, that contain all information, it was not easy, but we did it and it is looking good with the code.
* **One Withdraw/ One transfer per month for savings Account**: - In our program we made a function that withdraw an account and that transfer an amount as well, but we find it challenging to put 1 month limit just on savings account.