**Aman Song – C20325896**

**Object oriented programming assignment**

**Bank Management System Report**

**THE CODE**

Within my code, I have a total of 3 classes, 1 main class called ‘BankAccount’ and 2 subclasses, ‘CheckingsAccount’ and ‘SavingsAccount’ which inherits the main class. The main class provides all the basic methods such as withdrawals and deposits while the subclasses lay out specific conditions such as savings accounts may only withdraw once a month and checkings account allowing a negative balance up to a specified limit.  
The methods within my classes are simple \_\_init\_\_ initialises the data taken in from the account file, withdraw subtracts the amount from the funds if allowed so, transfer just takes away transferred amount from funds and \_\_str\_\_ just returns the accounts details if called.

I have also many functions within my code, most of them are used to perform a specific task such as view account, withdraw/deposit, view transactions etc and they are mainly used to write to and update the text files.

When the user first loads up the program, my ‘main function’ tells the user would they like to login or create an account, if they choose login, they must input their account ID and their 4-digit pin and then it opens the account text file and using a for loop, checks for any accounts that matches the credentials inputted by using ‘in line’. If an account has been found, the code takes in the details as a list and creates an instance using the first element of the list and the rest of the elements are put into the instance.

If the user chooses to create and account, the function ‘createAccount’ is called and the user is asked if they want to open a checking or savings account. Then the user is asked to input their details such as their name and to create a 4-digit pin. The code checks the age and what account they wanted to open and if they’re under 18 and chose a checking account the code tells them they can’t open this account and returns to the main function, else my code generates a unique ID for them and writes down all information into the accounts text file.

***Menu function***The menu function is quite simple, using a while loop it asks the user to choose an option and then calls the chosen option, there’s a try and except within to handle any unexpected input and asks the user again to select a valid option,

***accountOperation function***This function like the menu function, asks the user to select whether they would like to withdraw, deposit or return, error checking is implemented too. If they choose to withdraw, they are asked to input an amount and using the instance, the withdraw method is called. If the withdraw was made successfully, the account must be updated so a new data for the account is created using information from the current instance and replaces the old line in the file. If they choose to deposit, everything happens just like withdraw except this time funds are added to the account.

***transferFunds function***This function allows the user to transfer some funds to another account within the account text file. The user may return if they choose to not transfer when asked to. If they did want to transfer funds, they must first enter the recipients IBAN and the amount they wish to transfer, if an account is found, the amount is deducted from users account using the transfer method and the account is updated, meanwhile an instance for the recipients account is created and the amount is deposited, and it’s also updated.

***addTransaction function***This function is called after a withdraw, deposit or a transfer. The transactions text file is opened, and a transaction is recorded with details such as which account made the transaction and what kind of transaction was it and the date and time it was made. Accounts that received funds via transfer is also recorded.

***viewTransactions function***This function is quite simple, it first opens the transactions text file, loops through it, finding any transactions that has the currently opened bank account’s ID and displays the results.

***deleteAccount function***This function is to permanently delete an account from the customer and account text files, the user is asked twice if they wish to do so, and deletion will occur if they press YES twice however it will not delete if the user still has any remaining funds left in their account. If all steps are performed successfully, the customer and account texts file are overwritten with the same data except for the deleted account.

**USER MANUAL**

When you first load up the program you are presented with 2 choices, login or create and account, please enter in a number corresponding to the option and hit enter.

If you login, you are presented with another menu with 6 different choices, like before please enter a number and hit enter. Then depending on your choice, just follow what the screen tells you such as entering an amount to deposit.

If you wish to create and account, you must then choose what type of account you wish to open and then you must provide your first name, surname, age and you must create a 4-digit pin number (advise you to remember this).  
Once everything is done you are given your accounts name, unique ID (advise to also remember this as its important to login with) and your IBAN.

**CONTRIBUTIONS**

I am working alone so everything is done by me, Aman Song

**DIFFUCULTIES AND CHALLENEGES**

The most difficult part of the project was definitely at the beginning, mainly about the files. It was a massive challenge to figure out and implement a way to have persistent data as required, I had to figure out how to write data to the files and overcome the problem of corrupting data already in the files.  
Reading data from files was a big challenge as well, but once I learned how to write to files, access files, manipulate it and update, everything was sort of smooth sailing from there on  
It was a rocky start but it went well at the end (hopefully)