this is group D's python code

customer class:

it has the \_\_init\_\_ method which is a constructor and initialises the attributes of the class, in this case, NRIC number, Name of customer and the accounts of the customers.

\_\_str\_\_ method is used to convert the attributes to string and display it.

addaccount method has 1 parameter which is 'acc'. then it checks with the existing accounts a customer has, if the added account is not there in the list of existing accounts' then it is appended (added) to the list. however, if the added account is already here in the list, then a message, "account already exists" is displayed.

removeaccount method also has only one parameter which again, is acc. the added account is compared with the existing list. it is deleted if the added account is there in the list of accounts else a message, "account already got removed" is displayed.

transaction\_report method has 2 parameters, start\_date and end\_date. for each account, it prints a transaction report for each account. for the date, the user needs to enter starting date using datetime.datetime(). this means starting date can be any day before and including today.for the ending date, datetime.datetime.now is used.

Account class:

\_\_init\_\_ method initialises the attributes, account\_number, balance, customer and branch.

\_\_str\_\_ method converts the attributes into string and displays them.

transaction report method has 2 parameters, start\_date and end\_date. then, the list is filtered and this will find all the transactions that have taken place between start\_time and end\_time for a particular account. then, every transaction in the list of transactions of the account is returned in string form.

bank class:

\_\_init\_\_ method initialises the attributes, name and branches.

\_\_str\_\_ method will display the branches in string form

transaction\_report has 2 parameters, start\_date and end\_date. a transaction report is made which will display the transaction report for all accounts under all branches.

branch class:

\_\_init\_\_ method initialises the attributes, branch\_name, parent\_bank and accounts

\_\_str\_\_ method will display all the accounts of customers for a particular branch

transaction class:

\_\_init\_\_ method initialises the attributes, amount, transaction\_type, account and remarks

\_\_str\_\_ method will display the attributes in string form

withdraw class:

this class is inherited from transaction class.

\_\_init\_\_ method is used to initialise attributes, amount, account and remarks. then the super class's constructor is inherited. the amount is deducted from balance

\_\_str\_\_ methods prints the attributes (string form)

deposit class:

this class is inherited from transaction class.

\_\_init\_\_ method is used to initialise attributes, amount, account and remarks. then the super class's constructor is inherited. the amount is added to balance

\_\_str\_\_ methods prints the attributes (string form)

Transfer function:

method is used to initialise attributes, acc1, acc2, amount and remarks

withdraw and deposit methods are used to transfer amount from 1 account to another, in this case, acc1 will transfer money to acc2.