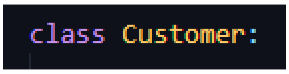
OOP

Student Number: C20402732

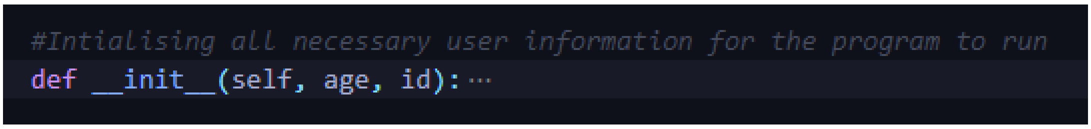
Student Name: Aleksey Makarevich

The Customer Class:



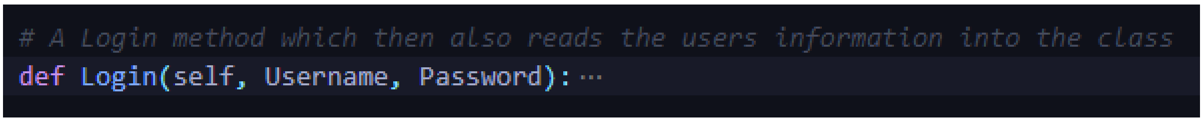
```
class Customer:
```

This contains the current users id and age.



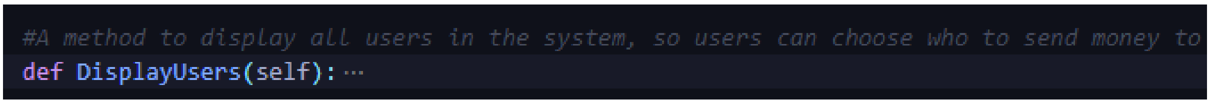
```
#Intialising all necessary user information for the program to run
def __init__(self, age, id): ···
```

This magic method initializes the variable I need.



```
# A Login method which then also reads the users information into the class
def Login(self, Username, Password): ···
```

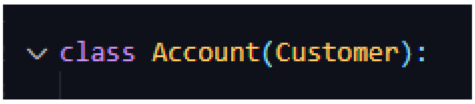
This Method takes in the users entered username and password, logging them in if the information is correct, and reads the users information into the class.



```
#A method to display all users in the system, so users can choose who to send money to
def DisplayUsers(self): ···
```

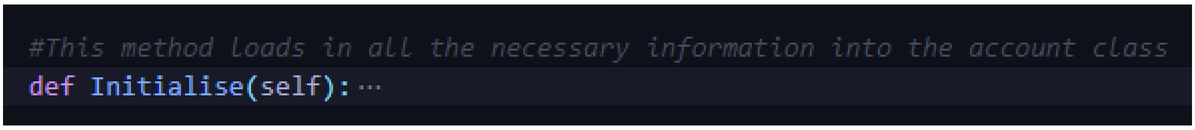
This Method reads from the users.txt file, displaying all user id’s with their name to allow the current user to know who they want to transfer money to.

The Account class:



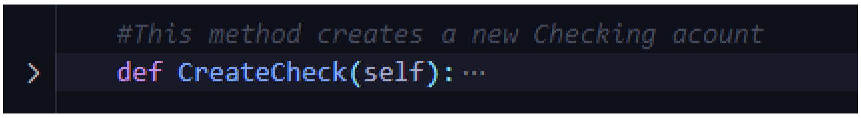
```
class Account(Customer):
```

This contains the account information on the current user.
It is also a subclass of the users class.



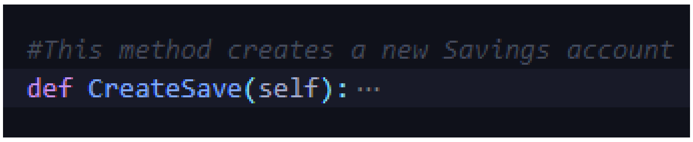
```
#This method loads in all the necessary information into the account class
def Initialise(self): ···
```

This magic method initializes the variable I need.



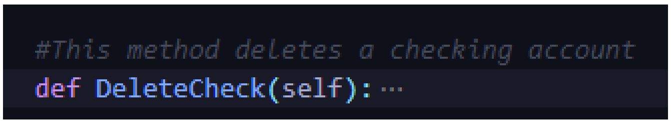
```python
    #This method creates a new Checking acount
    def CreateCheck(self): ···
```

This method creates a new checking account. It first checks if the user has an account first before creating one.



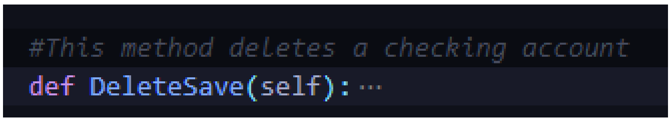
```python
#This method creates a new Savings account
def CreateSave(self): ···
```

This method creates a new savings account. It first checks if the user has an account first before creating one.



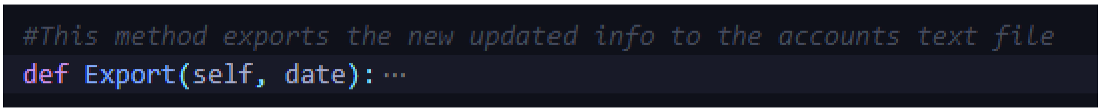
```python
#This method deletes a checking account
def DeleteCheck(self): ···
```

This method deletes the users checking account if they have one.



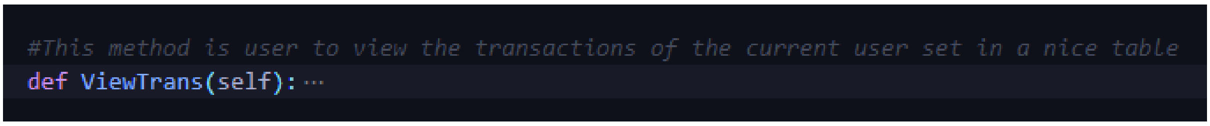
```python
#This method deletes a checking account
def DeleteSave(self): ···
```

This method deletes the users savings account if they have one.



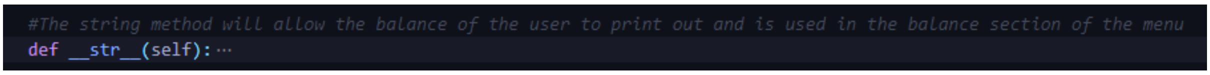
```python
#This method exports the new updated info to the accounts text file
def Export(self, date): ···
```

This method exports all the new updated information to the accounts.txt file



```python
#This method is user to view the transactions of the current user set in a nice table
def ViewTrans(self): ···
```

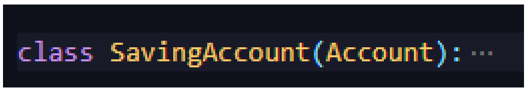
This method is all the previous transactions of the current user.



```python
#The string method will allow the balance of the user to print out and is used in the balance section of the menu
def __str__(self): ···
```

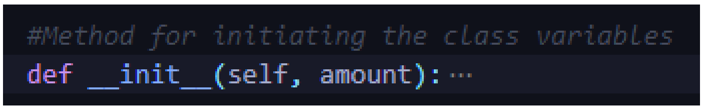
This is a magic method which prints the users balances.

The SavingAccount Class:



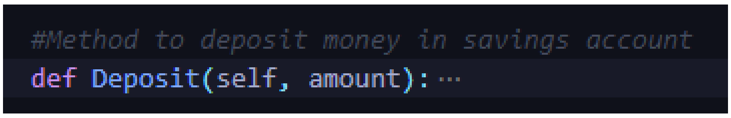
```python
class SavingAccount(Account): ···
```

This contains the account balance of the current user.
It is also a subclass of the accounts class.



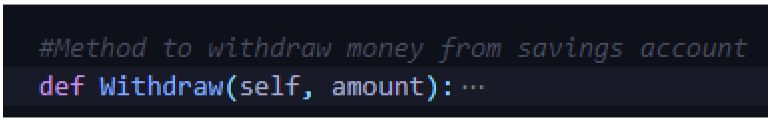
```python
#Method for initiating the class variables
def __init__(self, amount): ···
```

This magic method initializes the variable I need.



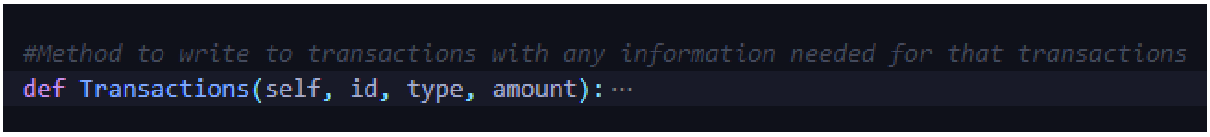
```
#Method to deposit money in savings account
def Deposit(self, amount):···
```

This method allows the user to deposit money to their savings account.



```
 #Method to withdraw money from savings account
 def Withdraw(self, amount):···
```

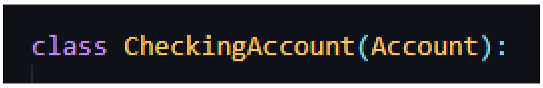
This method allows the user to withdraw from their savings account.



```
#Method to write to transactions with any information needed for that transactions
def Transactions(self, id, type, amount):···
```

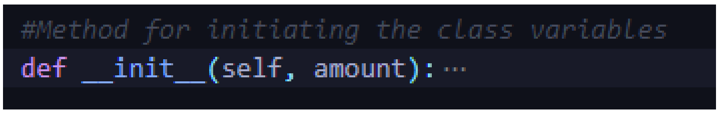
This method sends the information of a transaction and prints it to the accounttransactions.txt file.

The CheckingAccount Class:



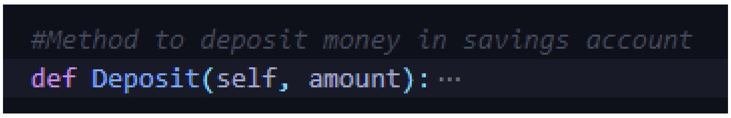
```
class CheckingAccount(Account):
```

This contains the account balance of the current user.
It is also a subclass of the accounts class.



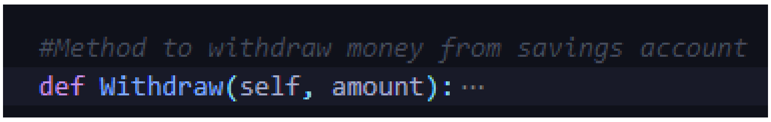
```
#Method for initiating the class variables
def __init__(self, amount):···
```

This magic method initializes the variable I need.



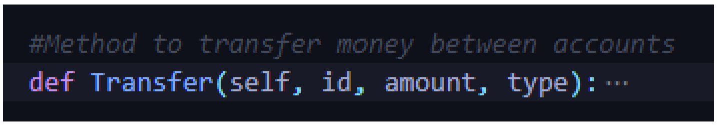
```
#Method to deposit money in savings account
def Deposit(self, amount):···
```

This method allows the user to deposit money to their savings account.



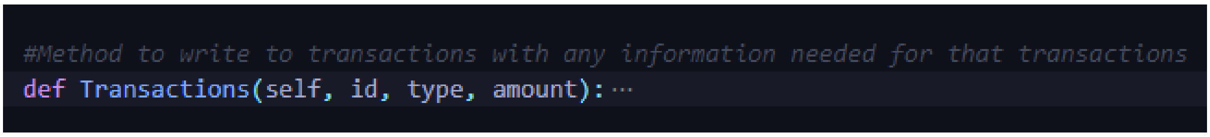
```
 #Method to withdraw money from savings account
 def Withdraw(self, amount):···
```

This method allows the user to withdraw from their savings account.



```
#Method to transfer money between accounts
def Transfer(self, id, amount, type):···
```

This method allows the user to transfer money from their checking account to other people's accounts.



```
#Method to write to transactions with any information needed for that transactions
def Transactions(self, id, type, amount):···
```

This method sends the information of a transaction and prints it to the accounttransactions.txt file.

User Manual:

My program consists of a few nested menu screens.

In order to choose an option you must enter your chosen option and press enter.

To get to the main menu of the program you must login by pressing 1 + Enter.

You are then prompted to enter your username and then password.
If these do not match an account, you are brought back to the welcome screen where you can then register if you do not have an account.

All menu's operate the same way, so by reading the menu that is printed, you may choose where you want to go.

Team Member contributions:

I am the only person on my team, I did it all. I contributed to my own project being the only person on my team.

Difficulties :

Starting the assignment.
This was by far the hardest part of doing this assignment.

After sitting down and finally getting my first bit of code done, the rest of it flowed like butter on hot toast.

The other challenging part was finding out what a class is, and how they work.
After doing some research from my fellow peers and the internet, I think I found an understanding and think its all complete.