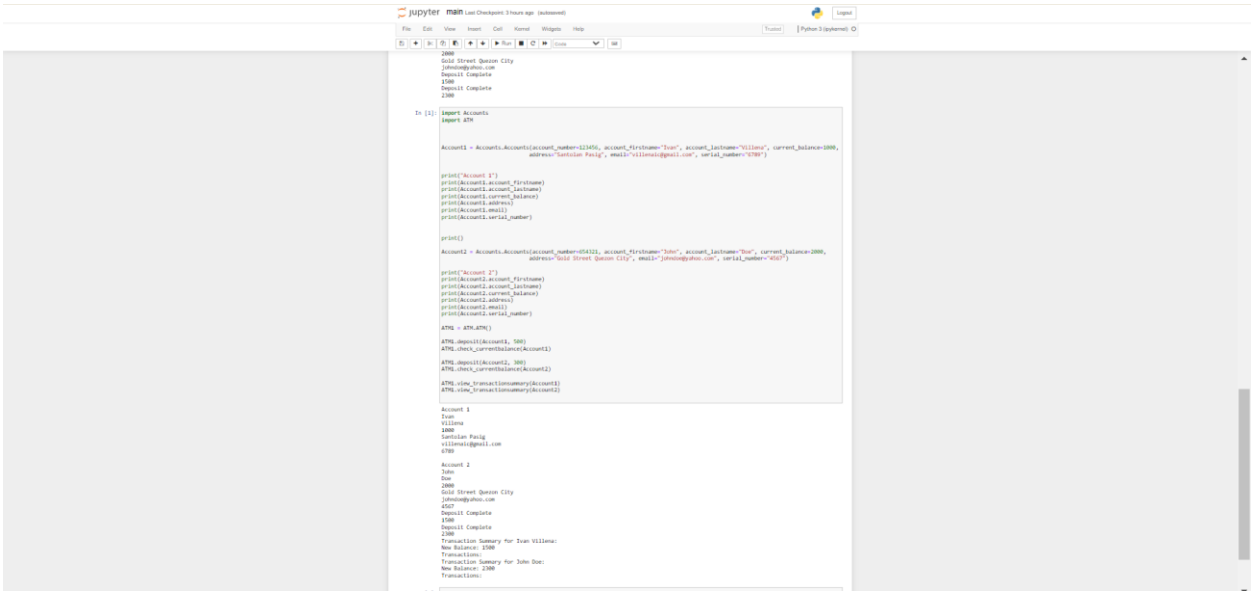
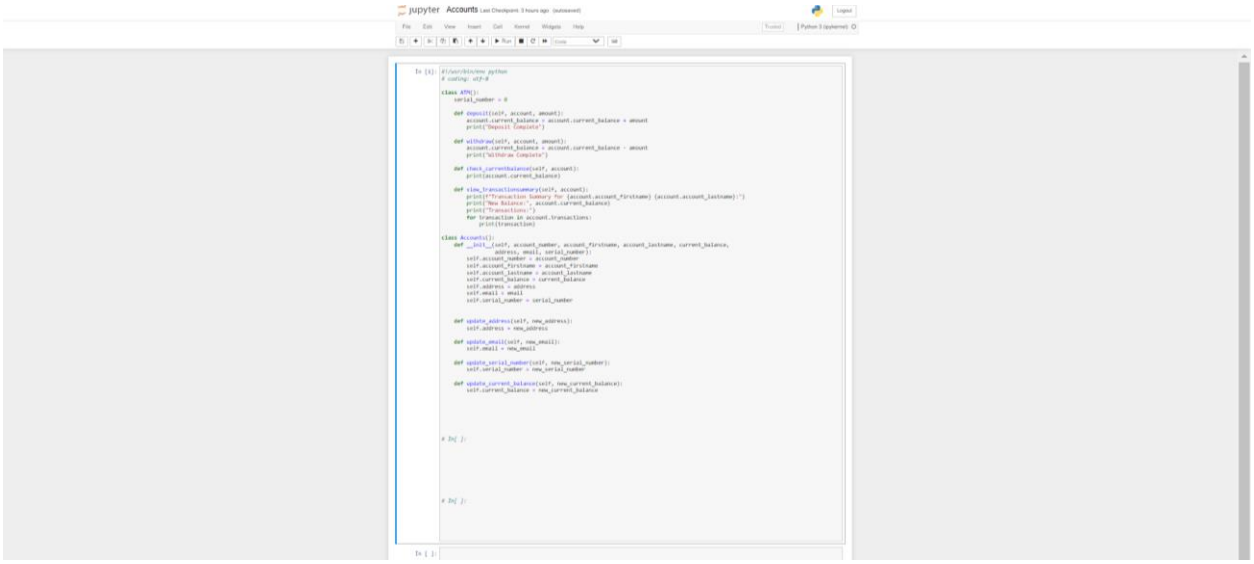


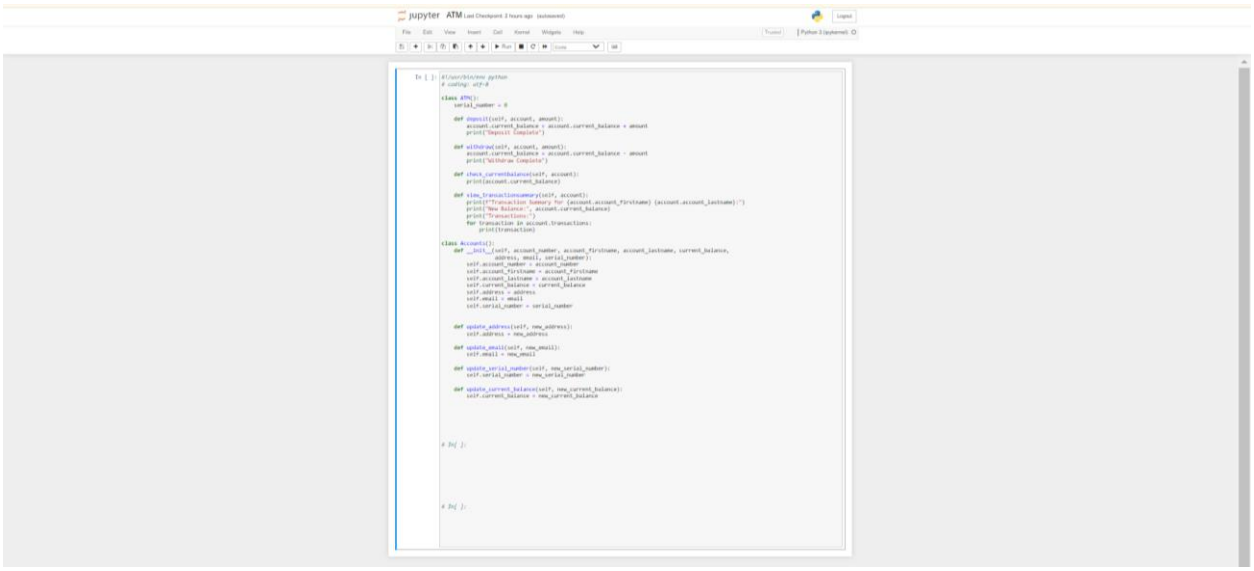
```
2000
Gold Street Quezon City
johndoe@yahoo.com
Deposit Complete
1500
Deposit Complete
2300
```

```
In [1]: import Accounts
        import ATM


        Account1 = Accounts.Accounts(account_number=123456, account_firstname="Ivan", account_lastname="Villena", current_balance=1000,
                                     address="Santolan Pasig", email="villenaic@gmail.com", serial_number="6789")


        print("Account 1")
        print(Account1.account_firstname)
        print(Account1.account_lastname)
        print(Account1.current_balance)
        print(Account1.address)
        print(Account1.email)
        print(Account1.serial_number)


        print()


        Account2 = Accounts.Accounts(account_number=654321, account_firstname="John", account_lastname="Doe", current_balance=2000,
                                     address="Gold Street Quezon City", email="johndoe@yahoo.com", serial_number="4567")

        print("Account 2")
        print(Account2.account_firstname)
        print(Account2.account_lastname)
        print(Account2.current_balance)
        print(Account2.address)
        print(Account2.email)
        print(Account2.serial_number)

        ATM1 = ATM.ATM()

        ATM1.deposit(Account1, 500)
        ATM1.check_currentbalance(Account1)

        ATM1.deposit(Account2, 300)
        ATM1.check_currentbalance(Account2)

        ATM1.view_transactionsummary(Account1)
        ATM1.view_transactionsummary(Account2)
```

```
Account 1
Ivan
Villena
1000
Santolan Pasig
villenaic@gmail.com
6789

Account 2
John
Doe
2000
Gold Street Quezon City
johndoe@yahoo.com
4567
Deposit Complete
1500
Deposit Complete
2300
Transaction Summary for Ivan Villena:
New Balance: 1500
Transactions:
Transaction Summary for John Doe:
New Balance: 2300
Transactions:
```



```
In [1]: #!/usr/bin/env python
        # coding: utf-8

        class ATM():
            serial_number = 0

            def deposit(self, account, amount):
                account.current_balance = account.current_balance + amount
                print('Deposit Complete')

            def withdraw(self, account, amount):
                account.current_balance = account.current_balance - amount
                print('Withdraw Complete')

            def check_currentbalance(self, account):
                print(account.current_balance)

            def view_transactionsummary(self, account):
                print(f"Transaction Summary for {account.account_firstname} {account.account_lastname}:")
                print("New Balance:", account.current_balance)
                print("Transactions:")
                for transaction in account.transactions:
                    print(transaction)

        class Accounts():
            def __init__(self, account_number, account_firstname, account_lastname, current_balance,
                         address, email, serial_number):
                self.account_number = account_number
                self.account_firstname = account_firstname
                self.account_lastname = account_lastname
                self.current_balance = current_balance
                self.address = address
                self.email = email
                self.serial_number = serial_number


            def update_address(self, new_address):
                self.address = new_address

            def update_email(self, new_email):
                self.email = new_email

            def update_serial_number(self, new_serial_number):
                self.serial_number = new_serial_number

            def update_current_balance(self, new_current_balance):
                self.current_balance = new_current_balance




        # In[ ]:




        # In[ ]:


In [ ]:
```



Observations:

- Inconsistencies in writing the parameters may lead to an error, hence, spelling and capitalization are very important in object-oriented programming
- Moreover, indentation is very important in creating classes, otherwise, would result in error.
- The main.py can be modified to suit the user needs as long as the modules are intact