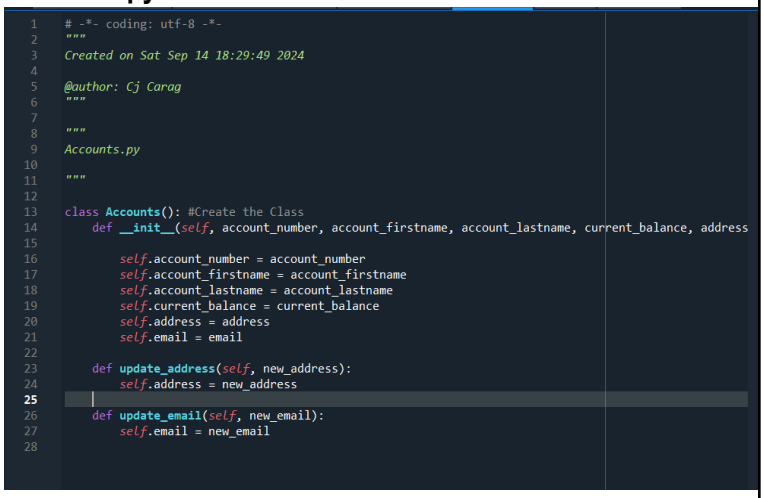
Laboratory Activity 1 - Class, Objects, Methods	
Carag, Carl Jervie B.	09/15/2024
Course/Section CPE 009B-CPE21S4	Engr. Maria Rizette Sayo

Code Body Part

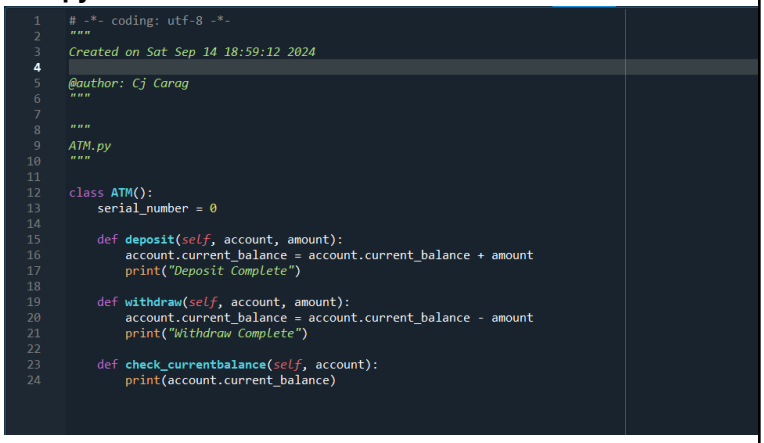
Accounts.py



```python
# -*- coding: utf-8 -*-
"""
Created on Sat Sep 14 18:29:49 2024

@author: Cj Carag
"""

"""
Accounts.py

"""

class Accounts(): #Create the Class
    def __init__(self, account_number, account_firstname, account_lastname, current_balance, address

        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

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

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

Observation: I have created a file separated name “Accounts.py” containing the class named “Accounts” and after that I define variables such as account number, account first name, account last name, current balance, email, address. Self was added before the names of the variables because it allows me to access their attributes.

ATM.py



```python
# -*- coding: utf-8 -*-
"""
Created on Sat Sep 14 18:59:12 2024

@author: Cj Carag
"""

"""
ATM.py
"""

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)
```

Observation: In this part I have created a file named “ATM.py’ containing the class named “ATM” I have defined variables such as deposit which then has the attributes namely amount, account and current balance it allows the transfer of cash value in the accounts of the clients.

Main.py

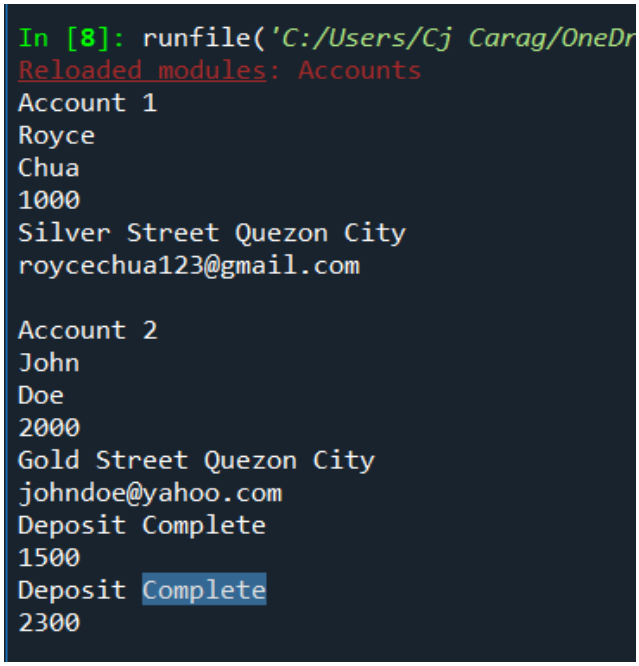
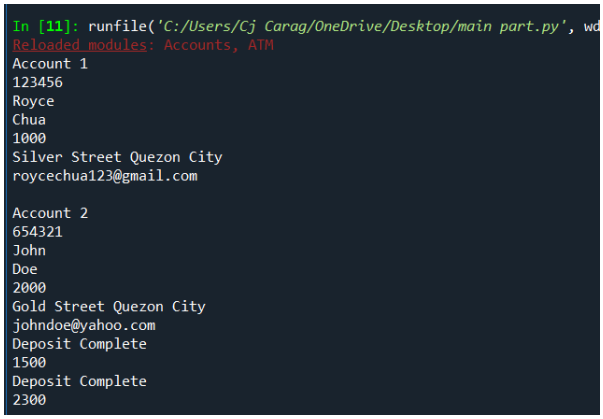
```
Created on Sat Sep 14 21:51:58 2024
Main.py
"""
import Accounts
import ATM
Account1 = Accounts.Accounts(account_number= 123456, account_firstname = "Royce", account_lastname = "Chua",
                             current_balance = 1000, address = "Silver Street Quezon City", email = "roycechua123@gmail.com")

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

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")

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

#Creating and Using an ATM Object
ATM1 = ATM.ATM()
ATM1.deposit(Account1, 500)
ATM1.check_currentbalance(Account1)

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

Observation: In this part I have created the last file which will serve as the body of the code, it is called the “Main.py”. I have created a class namely “Main”, here is where I could import the 2 files that I have created which are Accounts.py and ATM.py whose defined functions are being called in this file. The input being stored here uses the function of the two files that enables the placed input to be stored and processed, once it is done this file begins printing the finished results or outcome of the input.

Code Result Part.

Main.py (Basic)

```
In [7]: runfile('C:/Users/Cj Carag/OneDrive/Desktop/main part.py',
wdir='C:/Users/Cj Carag/OneDrive/Desktop')
Account 1
Royce
Chua
1000
Silver Street Quezon City
roycechua123@gmail.com

Account 2
John
Doe
2000
Gold Street Quezon City
johndoe@yahoo.com
```

Main.py(Modified with Creating and Using an ATM Object)	
Main.py (Modified part with the inclusion of variable "account_number)	

Supplementary Activity

Tasks

1. Modify the ATM.py program and add the constructor function.

```
"""

"""
ATM.py
"""
import Accounts


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)

Account1 = Accounts.Accounts(account_number= 123456, account_firstname = "Royce", account_lastname = "Chua",
                             current_balance = 1000, address = "Silver Street Quezon City", email = "roycechua123@gmail.com")

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

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")

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

Results

```
In [15]: runfile('C:/Users/Cj Carag/OneDrive/Desktop/ATM.py', wdir='C:/Users
Carag/OneDrive/Desktop')
Account 1
123456
Royce
Chua
1000
Silver Street Quezon City
roycechua123@gmail.com

Account 2
654321
John
Doe
2000
Gold Street Quezon City
johndoe@yahoo.com
```

2. Modify the main.py program and initialize the ATM machin with any integer serial number combination and display the serial number at the end of the program.

```
"""
import Accounts


class ATM():
    def Serial (self, account, serial_number):
        account.serial_number = serial_number
        print(account.serial_number)

    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)

Account1 = Accounts.Accounts(serial_number=887345, account_number= 123456, account_firstname = "Royce", account_lastname = "Chua",
                             current_balance = 1000, address = "Silver Street Quezon City", email = "roycechua123@gmail.com")

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

print()

Account2 = Accounts.Accounts(serial_number=997455, account_number= 654321,account_firstname = "John", account_lastname = "Doe",
                             current_balance = 2000, address = "Gold Street Quezon City", email = "johndoe@yahoo.com")

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

Result	

Questions.

1.What is a class in object oriented programming?

-Class bundles data and functionality together. The mechanism class of python adds a minimum new syntax and systematics. It is a mixture of class mechanisms from C++ and Modula-3. The class inheritance of python allows multiple base classes. A derived class can override any of its bases.

2.Why do you think classes are being implemented in certain programs while some are sequential(line-by-line)?

-The classes and its details can be concealed and main syntaxes and its functionality can be displayed which is easier to understand. Classes also combines attributes and methods in one phase which is much more convenient to compile.

3. How is it that there are variables of the same name such account_firstname and account_lastname that exist but have different values?

-They are being updated through the attributes such as defined functions allowing their datas to be accessed and therefore can be changed easily based on the method applied in its defined function.

4.Explain the constructor functions role in initializing the attributes of the class? When does the Constructor function execute or when is the constructor function called?

-The role of the constructor intializes an object instance in a class. Its main purpose was to create an object and assign value or data to the object's members. It is also needed especially when we are to assign new objects. A constructor is remotely or automatically called when an instance of the class is made.

5. Explain the benefits of using Constructors over initializing the variables one by one in the main program?

-The benefits of using constructor is that first in code organization all of the set data type such as numeric data, string data, and sequence data and other kinds of data types are stored together with their attributes and is initialized once called in the program.

Conclusion:

Doing this Activity makes me aware of the functions of classes, methods, and constructors. Classes helped me bundle data attributes and methods together using only one body of code. Inheritance became easy as the set attributes were directly allocated inside the class making it operate once called. Constructors also contributes to the ease of allocation of datas to execute operations in its variables. Having a constructor improves the organization of code but in much detailed way alike to classes. I have used methods wherein I renamed the file into their corresponding names and imported them into the main program which allowed me to use their functions inside to initialize the operations needed to carry out. The account names together with current balance, email, address and serial number where created separately as to maintain proper organization of code they were set under a specific class called "Accounts", the function of this code was to be the storage of the input details in which it will be placed together with the separated code which I classified as the program that enables the modification of the datas of the attributes, this code was named "ATM" which allowed me to add value to the current balance of Account 1 and Account 2 making their variables changed.

Assessment Rubric: