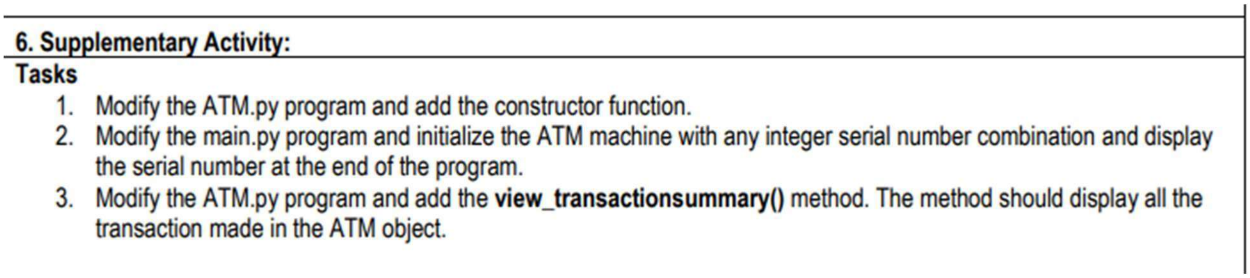
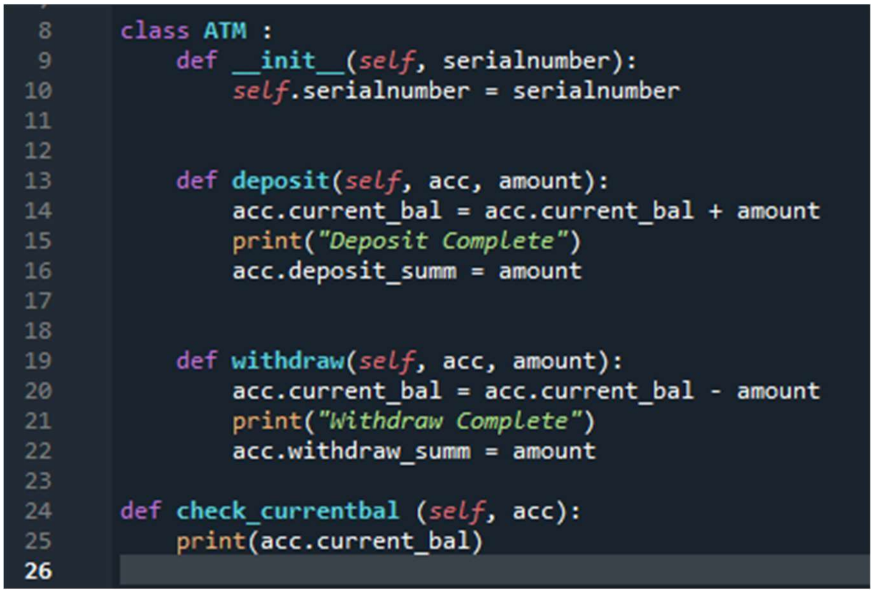
Hands-on Activity 4.1 Class, Objects, Methods	
Buenafe, Dhafny Efa, Christian Francisco, Lauper Xavier V.	4/8/2022
Course/Section – BSCPE12S1	Engr. Roman M. Richard



6. Supplementary Activity:

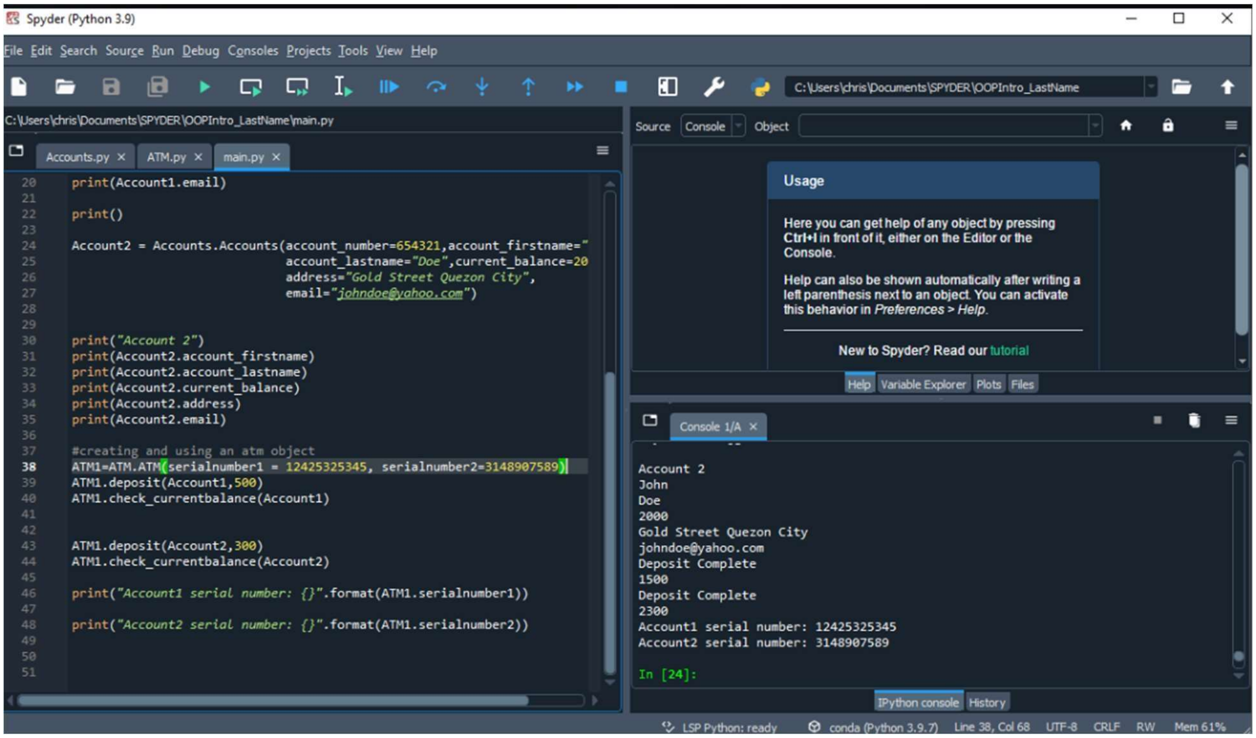
Tasks

1. Modify the ATM.py program and add the constructor function.
2. Modify the main.py program and initialize the ATM machine with any integer serial number combination and display the serial number at the end of the program.
3. Modify the ATM.py program and add the **view_transactionsummary()** method. The method should display all the transaction made in the ATM object.

1. 

```python
class ATM :
    def __init__(self, serialnumber):
        self.serialnumber = serialnumber


    def deposit(self, acc, amount):
        acc.current_bal = acc.current_bal + amount
        print("Deposit Complete")
        acc.deposit_summ = amount


    def withdraw(self, acc, amount):
        acc.current_bal = acc.current_bal - amount
        print("Withdraw Complete")
        acc.withdraw_summ = amount

def check_currentbal (self, acc):
    print(acc.current_bal)

```

2.



```python
print(Account1.email)

print()

Account2 = Accounts.Accounts(account_number=654321,account_firstname="
                             account_lastname="Doe",current_balance=20
                             address="Gold Street Quezon City",
                             email="johndoe@yahoo.com")


print("Account 2")
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(serialnumber1 = 12425325345, serialnumber2=3148907589)
ATM1.deposit(Account1,500)
ATM1.check_currentbalance(Account1)


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

print("Account1 serial number: {}".format(ATM1.serialnumber1))

print("Account2 serial number: {}".format(ATM1.serialnumber2))
```

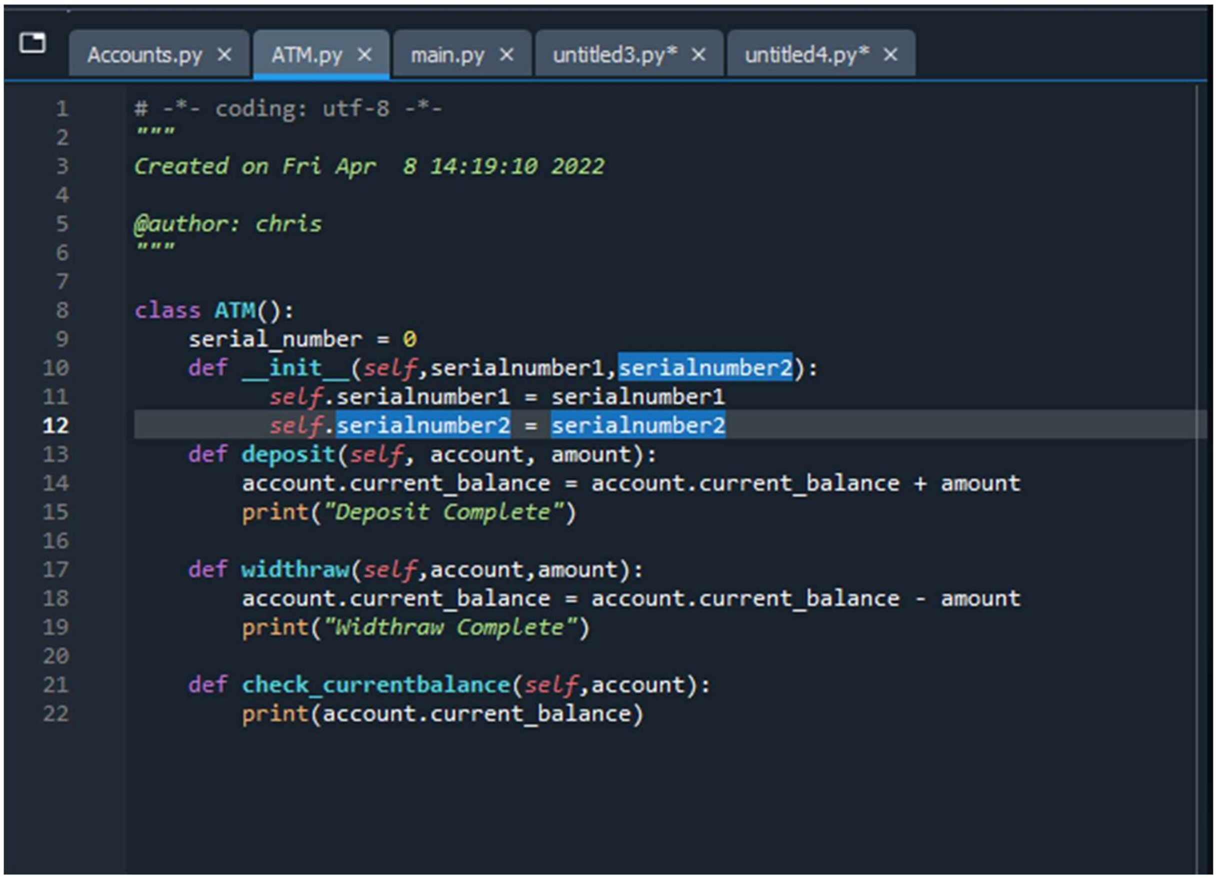
Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

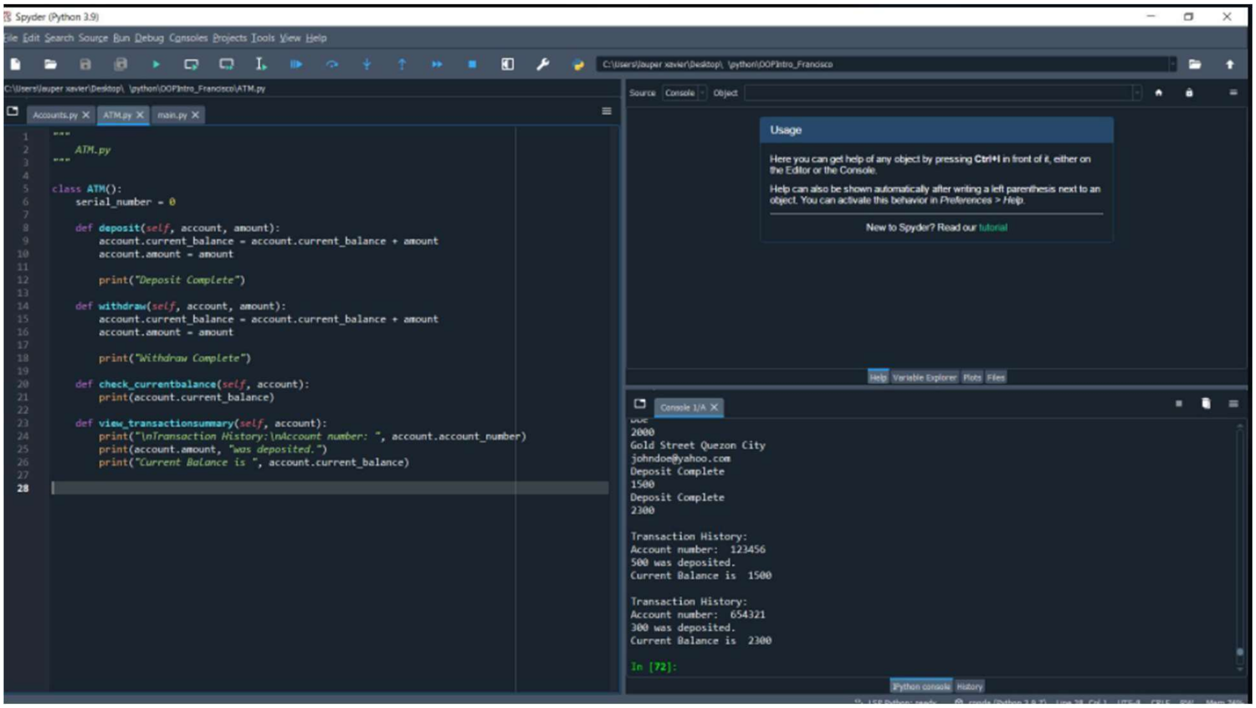
New to Spyder? Read our tutorial

```
Account 2
John
Doe
2000
Gold Street Quezon City
johndoe@yahoo.com
Deposit Complete
1500
Deposit Complete
2300
Account1 serial number: 12425325345
Account2 serial number: 3148907589

In [24]:
```



```python
# -*- coding: utf-8 -*-
"""
Created on Fri Apr  8 14:19:10 2022

@author: chris
"""

class ATM():
    serial_number = 0
    def __init__(self,serialnumber1,serialnumber2):
          self.serialnumber1 = serialnumber1
          self.serialnumber2 = serialnumber2
    def deposit(self, account, amount):
        account.current_balance = account.current_balance + amount
        print("Deposit Complete")

    def widthraw(self,account,amount):
        account.current_balance = account.current_balance - amount
        print("Widthraw Complete")

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

3.


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\auper xavier\Desktop\ \python\OOPIntro_Francisco
C:\Users\auper xavier\Desktop\ \python\OOPIntro_Francisco\ATM.py
Accounts.py
ATM.py
main.py
Source
Console
Object
Help
Variable Explorer
Plots
Files
Console 1/A
IPython console
History

```
"""
    ATM.py
"""

class ATM():
    serial_number = 0

    def deposit(self, account, amount):
        account.current_balance = account.current_balance + amount
        account.amount = amount

        print("Deposit Complete")

    def withdraw(self, account, amount):
        account.current_balance = account.current_balance + amount
        account.amount = amount

        print("Withdraw Complete")

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

    def view_transactionsummary(self, account):
        print("\nTransaction History:\nAccount number: ", account.account_number)
        print(account.amount, "was deposited.")
        print("Current Balance is ", account.current_balance)

```

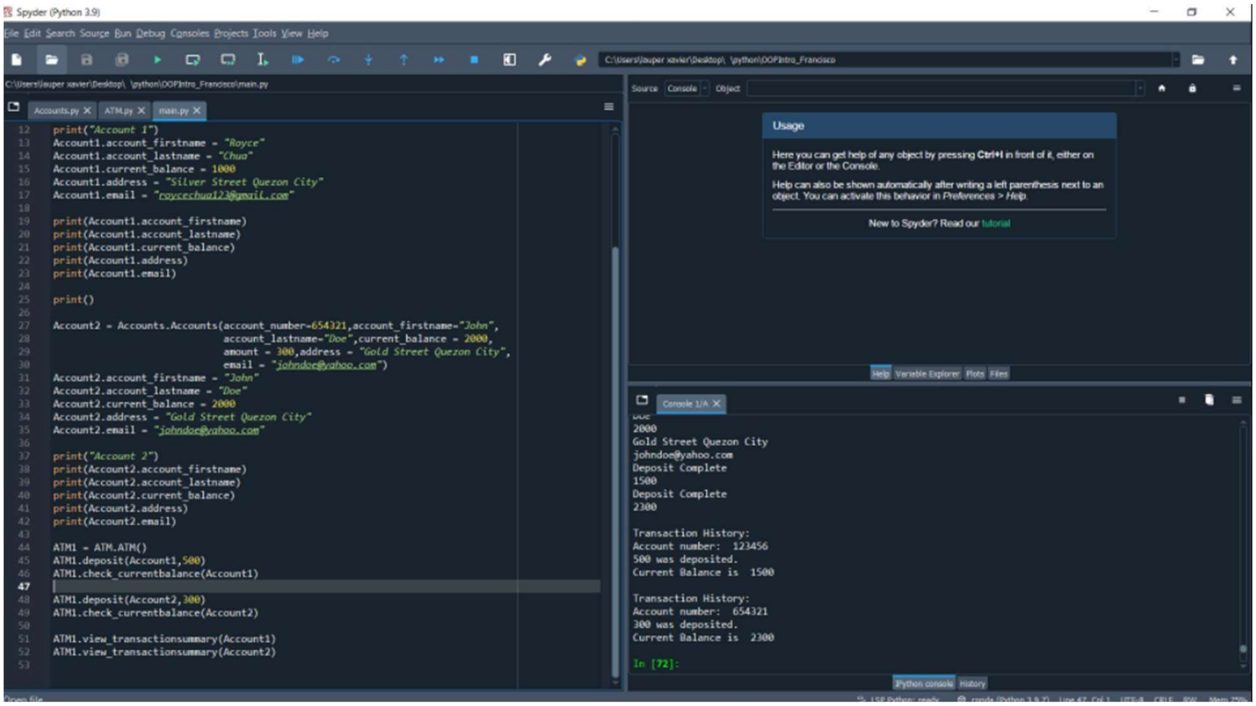
Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

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

Transaction History:
Account number:  123456
500 was deposited.
Current Balance is  1500

Transaction History:
Account number:  654321
300 was deposited.
Current Balance is  2300

In [72]:
```


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\auper xavier\Desktop\ \python\OOPIntro_Francisco
C:\Users\auper xavier\Desktop\ \python\OOPIntro_Francisco\main.py
Accounts.py
ATM.py
main.py
Source
Console
Object
Help
Variable Explorer
Plots
Files
Console 1/A
IPython console
History
Open file

```
print("Account 1")
Account1.account_firstname = "Royce"
Account1.account_lastname = "Chua"
Account1.current_balance = 1000
Account1.address = "Silver Street Quezon City"
Account1.email = "roycechua123@gmail.com"

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,
                             amount = 300,address = "Gold Street Quezon City",
                             email = "johndoe@yahoo.com")
Account2.account_firstname = "John"
Account2.account_lastname = "Doe"
Account2.current_balance = 2000
Account2.address = "Gold Street Quezon City"
Account2.email = "johndoe@yahoo.com"

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

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)

```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

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

Transaction History:
Account number:  123456
500 was deposited.
Current Balance is  1500

Transaction History:
Account number:  654321
300 was deposited.
Current Balance is  2300

In [72]:
```

Questions

1. What is a class in Object-Oriented Programming?

In OOP, a class is a labeled sketch of an object. It contains the object's name, color, and more.

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

- Organize the software such that the coder may reuse components.

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

- In fact, it is an attribute of the object (account), with the signifying _firstname.

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?

-A constructor is used in classes to initialize data members of class in order to avoid errors/segmentation faults.

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

-The benefits of constructors in initializing for me is that your assigned values is all in one place which is the initialization place where in we use _init_ and it is always called when an object is created.

in this place we will be declaring mostly all the information or variables that we will be using on our methods. Without using constructors we will be having type error because our class attribute will be having no arguments.

Conclusion:

During this activity it defines object oriented programming, attributes, methods and constructors more clear because of its given codes that needs to be done before proceeding to the supplementary activity. In doing this activity we understood the benefits of object oriented programming and the use of constructors. Doing this activity is kind of confusing having an error of small mistakes such as wrong spelling and indention. By having to work as a group by checking the mistakes or errors we solve the problem and the task given.

Members	**Participation**
Buenafe, Dhafny	**Task no. 1 Questions 1,2,3**
Efa, Christian	**Task no. 2 Question 4 Conclusion**
Francisco, Lauper Xavier V.	**Task no. 3 Question 5 Conclusion**

Honor Pledge for Grouped Projects " We accept responsibility for our role in ensuring the integrity of the work submitted by the group in which we participated"