Laboratory Activity 1 - Class, Objects, Methods - Pornobe	
Pornobe, Reuel Christian, M.	09/15/24
CPE21S4	Prof. Maria Rizette Sayo

Procedures:

1.

temp.py | Accounts.py | ATM.py | main.py

```python
import Accounts

Account1 = Accounts.Accounts()

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()

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

print(Account2.account_firstname)
print(Account2.account_lastname)
print(Account2.current_balance)
print(Account2.address)
print(Account2.email)
```

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

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

2. Modified

C:\Users\Reuel\Desktop\OOPIntro_Pornobe\main.py

temp.py × | Accounts.py × | ATM.py × | main.py ×

```python
import Accounts
import ATM

Account1 = Accounts.Accounts()

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()

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

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)
```

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

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

Supplementary Activity

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

```python
import random


class ATM():

    def __init__(self, amount):
        self.amount = amount

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

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

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

C:\Users\Reuel\Desktop\OOPIntro_Pornobe\main.py

temp.py | Accounts.py | ATM.py | main.py

```python
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@

print("Account 1")

print(Account1.account_firstname)
print(Account1.account_lastname)
print(Account1.current_balance)
print(Account1.address)
print(Account1.email)
print(Account1.account_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.co

print("Account 2")

print(Account2.account_firstname)
print(Account2.account_lastname)
print(Account2.current_balance)
print(Account2.address)
print(Account2.email)
print(Account2.account_number)

#Creating and Using an ATM object
ATM1 = ATM.ATM(amount = 500)

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

ATM2 = ATM.ATM(amount = 300)
ATM2.deposit(Account2, ATM2.amount)
ATM2.check_currentbalance(Account2)
```

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

Account 2
John
Doe
2000
Gold Street Quezon City
johndoe@yahoo.com
654321
Deposit Complete
1500
Deposit Complete
2300
```

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.

C:\Users\Reuel\Desktop\OOPIntro_Pornobe\ATM.py

temp.py × Accounts.py × ATM.py × main.py ×

```
1   import random
2
3
4   class ATM():
5
6       def __init__(self, amount, serialNumber):
7           self.amount = amount
8           self.serialNumber = serialNumber
9
10      def deposit(self, account, amount):
11          account.current_balance = account.current_balance + self.amount
12          print("Deposit Complete")
13
14      def withdraw(self, account, amount):
15          account.current_balance = account.current_balance - self.amount
16          print("Withdraw Complete")
17
18      def check_currentbalance(self, account):
19          return account.current_balance
20
21      def printSN(self, serialNumber):
22
23          for i in range(1, 10):
24              self.serialNumber.append(random.randint(0,9))
25
26          for i in self.serialNumber:
27              print(i, end="")
28
```

C:\Users\Reuel\Desktop\OOPIntro_Pornobe\main.py

temp.py | Accounts.py | ATM.py | main.py

```python
import Accounts
import ATM

#Initialization
Account1 = Accounts.Accounts(account_number = 123456, account_firstname = "Royce", account_lastname = "Chua", current_balance = 1000, address = "Silver Street Quezon City", email = "roycechua123@
Account2 = Accounts.Accounts(account_number = 654321, account_firstname = "John", account_lastname = "Doe", current_balance = 2000, address = "Gold Street Quezon City", email = "johndoe@yahoo.co
ATM1 = ATM.ATM(amount = 500, serialNumber = [])
ATM2 = ATM.ATM(amount = 300, serialNumber = [])


#Account1
print("Account 1")

print("First Name: ", Account1.account_firstname)
print("Last Name: ",Account1.account_lastname)
print("Current Balance: ",Account1.current_balance)
print("Address: ",Account1.address)
print("Email: ",Account1.email)
print("Account Number: ", Account1.account_number)
print("Current Balance: ", ATM1.check_currentbalance(Account1))
print("Deposited 500")
ATM1.deposit(Account1, ATM1.amount)
print("New Balance: ", ATM1.check_currentbalance(Account1))
#Outputing a Serial Number
ATM1.printSN(ATM1.serialNumber)

print("\n")


#Account 2
print("Account 2")

print("First Name: ",Account2.account_firstname)
print("Last Name: ",Account2.account_lastname)
print("Current Balance: ",Account2.current_balance)
print("Address: ",Account2.address)
print("Email: ", Account2.email)
print("Account Number: ", Account2.account_number)
print("Current Balance: ", ATM2.check_currentbalance(Account2))
print("Deposited 300")
ATM2.deposit(Account2, ATM2.amount)
print("New Balance: ", ATM2.check_currentbalance(Account2))

#Outputing a Serial Number
ATM2.printSN(ATM2.serialNumber)

```

```
In [1]: runfile('C:/Users/Reuel/Desktop/OOPIntro_Pornobe/
main.py', wdir='C:/Users/Reuel/Desktop/OOPIntro_Pornobe')
Account 1
First Name:  Royce
Last Name:  Chua
Current Balance:  1000
Address:  Silver Street Quezon City
Email:  roycechua123@gmail.com
Account Number:  123456
Current Balance:  1000
Deposited 500
Deposit Complete
New Balance:  1500
057362274

Account 2
First Name:  John
Last Name:  Doe
Current Balance:  2000
Address:  Gold Street Quezon City
Email:  johndoe@yahoo.com
Account Number:  654321
Current Balance:  2000
Deposited 300
Deposit Complete
New Balance:  2300
755797756
```

3. Modify the ATM.py program and add the view_transactionsummary() method. The method should display all the transaction made in the ATM object.

C:\Users\Reuel\Desktop\OOPIntro_Pornobe\ATM.py

temp.py | Accounts.py | ATM.py | main.py

```python
import random


class ATM():

    def __init__(self, amount, serialNumber):
        self.amount = amount
        self.serialNumber = serialNumber
        self.reportW = 0
        self.reportD = 0
        self.balance0 = 0
        self.balance1 = 0

    def deposit(self, account, amount):
        self.balance0 = account.current_balance
        account.current_balance = account.current_balance + self.amount
        self.reportD = self.amount
        self.balance1 = account.current_balance
        print("Deposit Complete")

    def withdraw(self, account, amount):
        self.balance0 = account.current_balance
        account.current_balance = account.current_balance - self.amount
        self.reportW = self.amount
        self.balance1 = account.current_balance
        print("Withdraw Complete")

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

    def printSN(self, serialNumber):

        for i in range(1, 10):
            self.serialNumber.append(random.randint(0,9))

        for i in self.serialNumber:
            print(i, end="")

    def view_transactionsummary(self,account):

        if self.reportD >> 0:
            print("Old Balance: ", self.balance0)
            print("Money Deposited in Account Number ", account.account_number, " is ", self.reportD)
            print("New Balance: ", self.balance1)
        elif self.reportW >> 0:
            print("Old Balance: ", self.balance0)
            print("Money Withdrawn in Account Number ", account.account_number, " is ", self.reportW)
            print("New Balance: ", self.balance1)
```

C:\Users\Reuel\Desktop\OOPIntro_Pornobe\main.py

temp.py | Accounts.py | ATM.py | main.py

```python
import Accounts
import ATM

#Initialization
Account1 = Accounts.Accounts(account_number = 123456, account_firstname = "Royce", account_lastname = "Chua", current_balance = 1000, address = "Silver Street Quezon City", email = "roycechua1
Account2 = Accounts.Accounts(account_number = 654321, account_firstname = "John", account_lastname = "Doe", current_balance = 2000, address = "Gold Street Quezon City", email = "johndoe@yahoo.
ATM1 = ATM.ATM(amount = 500, serialNumber = [])
ATM2 = ATM.ATM(amount = 300, serialNumber = [])


#Account1
print("Account 1")

print("First Name: ", Account1.account_firstname)
print("Last Name: ",Account1.account_lastname)
print("Current Balance: ",Account1.current_balance)
print("Address: ",Account1.address)
print("Email: ",Account1.email)
print("Account Number: ", Account1.account_number)
print("Current Balance: ", ATM1.check_currentbalance(Account1))
print("Deposited 500")
ATM1.deposit(Account1, ATM1.amount)
print("New Balance: ", ATM1.check_currentbalance(Account1))

#Outputing a Serial Number
print("SERIAL NUMBER: ", )
ATM1.printSN(ATM1.serialNumber)
print('\n')
#Transaction Summary
print("=========TRANSACTION SUMMARY=========")
ATM1.view_transactionsummary(Account1)

print("\n")


#Account 2
print("Account 2")

print("First Name: ",Account2.account_firstname)
print("Last Name: ",Account2.account_lastname)
print("Current Balance: ",Account2.current_balance)
print("Address: ",Account2.address)
print("Email: ", Account2.email)
print("Account Number: ", Account2.account_number)
print("Current Balance: ", ATM2.check_currentbalance(Account2))
print("Deposited 300")
ATM2.deposit(Account2, ATM2.amount)
print("New Balance: ", ATM2.check_currentbalance(Account2))


#Outputing a Serial Number
print("SERIAL NUMBER: ", )
ATM2.printSN(ATM2.serialNumber)
print('\n')
#Transaction Summary
print("=========TRANSACTION SUMMARY=========")
ATM1.view_transactionsummary(Account2)
```

```
In [1]: runfile('C:/Users/Reuel/Desktop/OOPIntro_Pornobe/
main.py', wdir='C:/Users/Reuel/Desktop/OOPIntro_Pornobe')
Account 1
First Name:  Royce
Last Name:  Chua
Current Balance:  1000
Address:  Silver Street Quezon City
Email:  roycechua123@gmail.com
Account Number:  123456
Current Balance:  1000
Deposited 500
Deposit Complete
New Balance:  1500
SERIAL NUMBER:
742479208

=========TRANSACTION SUMMARY=========
Old Balance:  1000
Money Deposited in Account Number  123456  is  500
New Balance:  1500


Account 2
First Name:  John
Last Name:  Doe
Current Balance:  2000
Address:  Gold Street Quezon City
Email:  johndoe@yahoo.com
Account Number:  654321
Current Balance:  2000
Deposited 300
Deposit Complete
New Balance:  2300
SERIAL NUMBER:
746044758

=========TRANSACTION SUMMARY=========
Old Balance:  1000
Money Deposited in Account Number  654321  is  500
New Balance:  1500
```

Questions:

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

   Class is where we create objects in Object-Oriented Programming. We write the variables and functions related to the object needed in a Class. A class can be a Super Class. A Subclass can inherit the variables and functions of the SuperClass.

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

   Classes are used to create organized variables and functions. It is necessary to be used in a program that requires a group of variables that is under one category.

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

   It is because the variable name is only inherited. Their constructors were different, that is why their values are different.

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?**

   Constructors are used to initialize the attributes of an object. It is automatically called when a new object is created.

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

   Variables are initialized quicker when using Constructors. You would only need to input the values in one syntax form. Further, it organizes the code beautifully.

Conclusion:

In this lab activity, I learned to use classes and constructors in python. The lab activity taught me how classes can make the code organized and clean. Furthermore, Constructors made sense to me during the coding. It is easier and faster to initialize variables or attributes using constructors.

Supplementary activity cemented my foundation about classes and constructors. I have learned the syntax and uses of the classes in the Python Program. The activity was not that hard but it was a bit tedious to create solutions to the problems or prompts asked by the supplementary activity.

The areas I can improve on is how I write the syntax of the code. I am still making minor mistakes when it comes to coding. However, I am improving on understanding the underlying concepts of Class, Objects, and Methods of Python.

FULL CODE:
Accounts.py

```python
class Accounts():


    def __init__(self, account_number, account_firstname, account_lastname,
current_balance, address, email):
        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
```

ATM.py

```python
import random


class ATM():

    def __init__(self, amount, serialNumber):
        self.amount = amount
        self.serialNumber = serialNumber
        self.reportW = 0
        self.reportD = 0
        self.balance0 = 0
        self.balance1 = 0

    def deposit(self, account, amount):
        self.balance0 = account.current_balance
        account.current_balance = account.current_balance + self.amount
        self.reportD = self.amount
        self.balance1 = account.current_balance
        print("Deposit Complete")

    def withdraw(self, account, amount):
        self.balance0 = account.current_balance
        account.current_balance = account.current_balance - self.amount
        self.reportW = self.amount
        self.balance1 = account.current_balance
        print("Withdraw Complete")

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

    def printSN(self, serialNumber):

        for i in range(1, 10):
            self.serialNumber.append(random.randint(0,9))

        for i in self.serialNumber:
            print(i, end="")

    def view_transactionsummary(self,account):

        if self.reportD >> 0:
            print("Old Balance: ", self.balance0)
            print("Money Deposited in Account Number ", account.account_number,
" is ", self.reportD)
            print("New Balance: ", self.balance1)
        elif self.reportW >> 0:
```

```
            print("Old Balance: ", self.balance0)
            print("Money Withdrawn in Account Number ", account.account_number,
" is ", self.reportW)
            print("New Balance: ", self.balance1)
```

main.py

```python
import Accounts
import ATM

#Initialization
Account1 = Accounts.Accounts(account_number = 123456, account_firstname =
"Royce", account_lastname = "Chua", current_balance = 1000, address = "Silver
Street Quezon City", email = "roycechua123@gmail.com")
Account2 = Accounts.Accounts(account_number = 654321, account_firstname =
"John", account_lastname = "Doe", current_balance = 2000, address = "Gold
Street Quezon City", email = "johndoe@yahoo.com")
ATM1 = ATM.ATM(amount = 500, serialNumber = [])
ATM2 = ATM.ATM(amount = 300, serialNumber = [])


#Account1
print("Account 1")

print("First Name: ", Account1.account_firstname)
print("Last Name: ",Account1.account_lastname)
print("Current Balance: ",Account1.current_balance)
print("Address: ",Account1.address)
print("Email: ",Account1.email)
print("Account Number: ", Account1.account_number)
print("Current Balance: ", ATM1.check_currentbalance(Account1))
print("Deposited 500")
ATM1.deposit(Account1, ATM1.amount)
print("New Balance: ", ATM1.check_currentbalance(Account1))

#Outputing a Serial Number
print("SERIAL NUMBER: ", )
ATM1.printSN(ATM1.serialNumber)
```

```
print('\n')
#Transaction Summary
print("=========TRANSACTION SUMMARY=========")
ATM1.view_transactionsummary(Account1)

print("\n")


#Account 2
print("Account 2")

print("First Name: ",Account2.account_firstname)
print("Last Name: ",Account2.account_lastname)
print("Current Balance: ",Account2.current_balance)
print("Address: ",Account2.address)
print("Email: ", Account2.email)
print("Account Number: ", Account2.account_number)
print("Current Balance: ", ATM2.check_currentbalance(Account2))
print("Deposited 300")
ATM2.deposit(Account2, ATM2.amount)
print("New Balance: ", ATM2.check_currentbalance(Account2))


#Outputing a Serial Number
print("SERIAL NUMBER: ", )
ATM2.printSN(ATM2.serialNumber)
print('\n')
#Transaction Summary
print("=========TRANSACTION SUMMARY=========")
ATM1.view_transactionsummary(Account2)
```