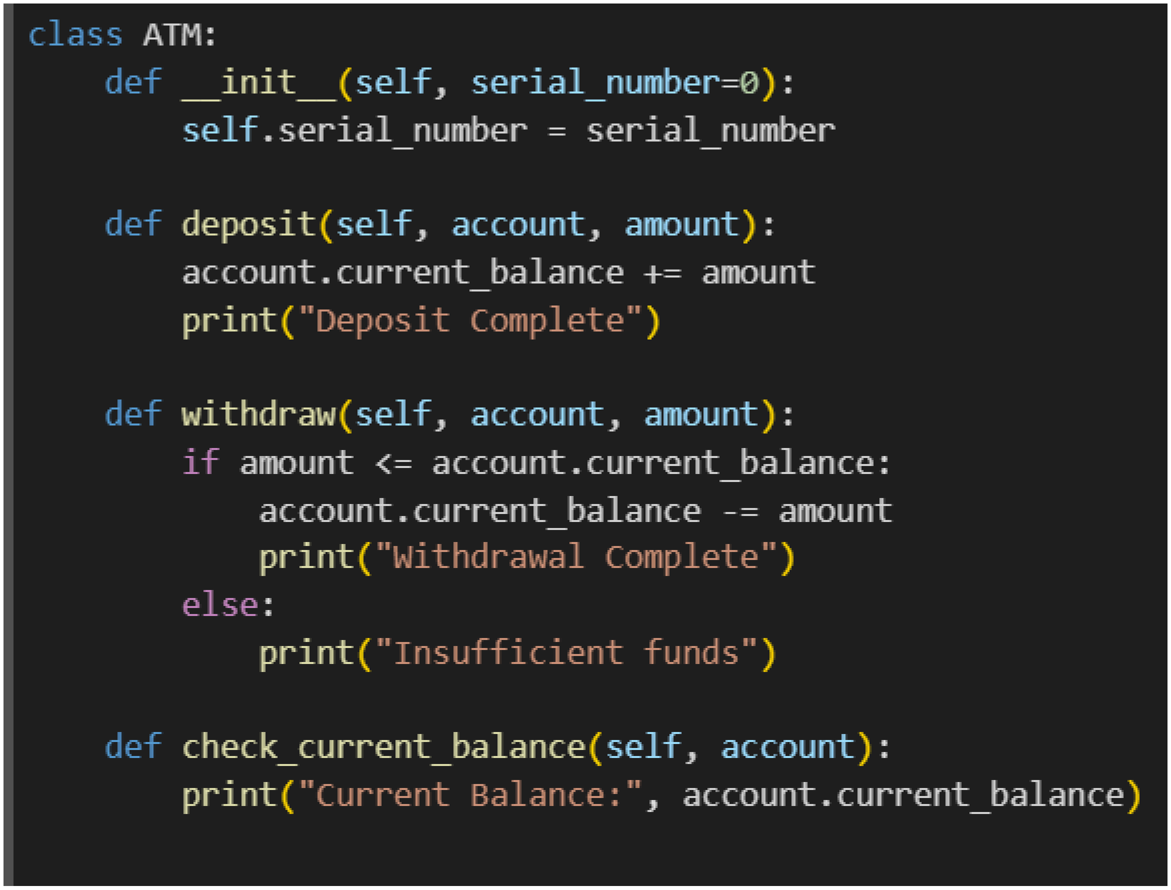
Activity Name 1 - Class, Objects, Methods

Virtucio, Dominic Joseph P.	09/14/24
CPE 009B - CPE 21S4	Ma'am Maria Rizette Sayo

6. Supplementary Activity

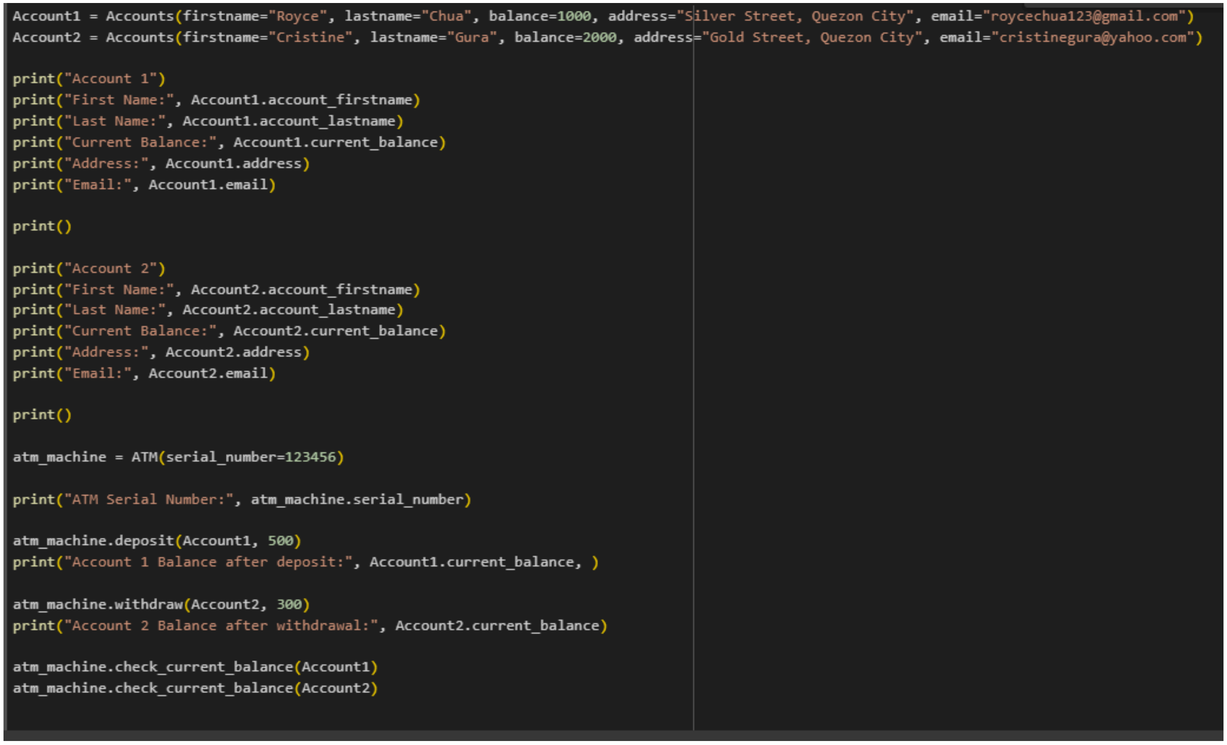
Tasks

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

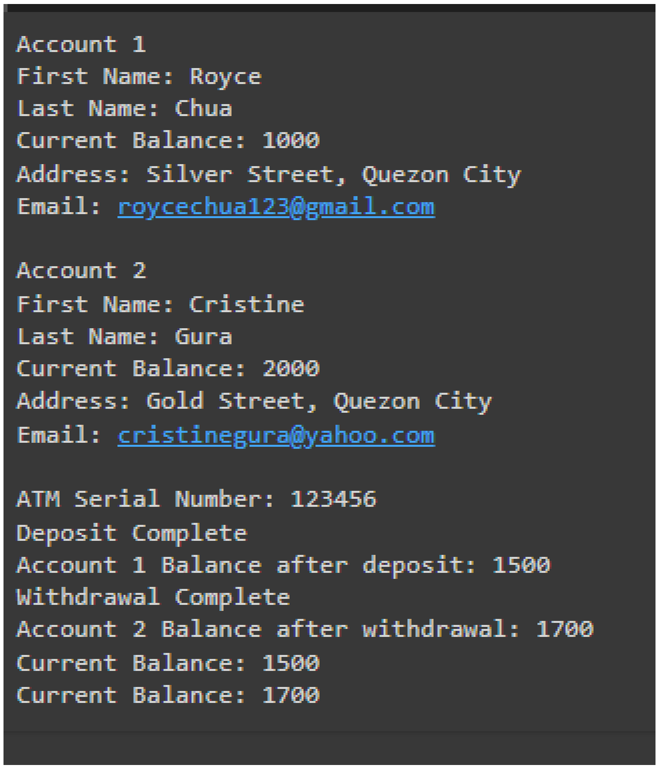


```
class ATM:
    def __init__(self, serial_number=0):
        self.serial_number = serial_number

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

    def withdraw(self, account, amount):
        if amount <= account.current_balance:
            account.current_balance -= amount
            print("Withdrawal Complete")
        else:
            print("Insufficient funds")

    def check_current_balance(self, account):
        print("Current Balance:", account.current_balance)
```

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.**

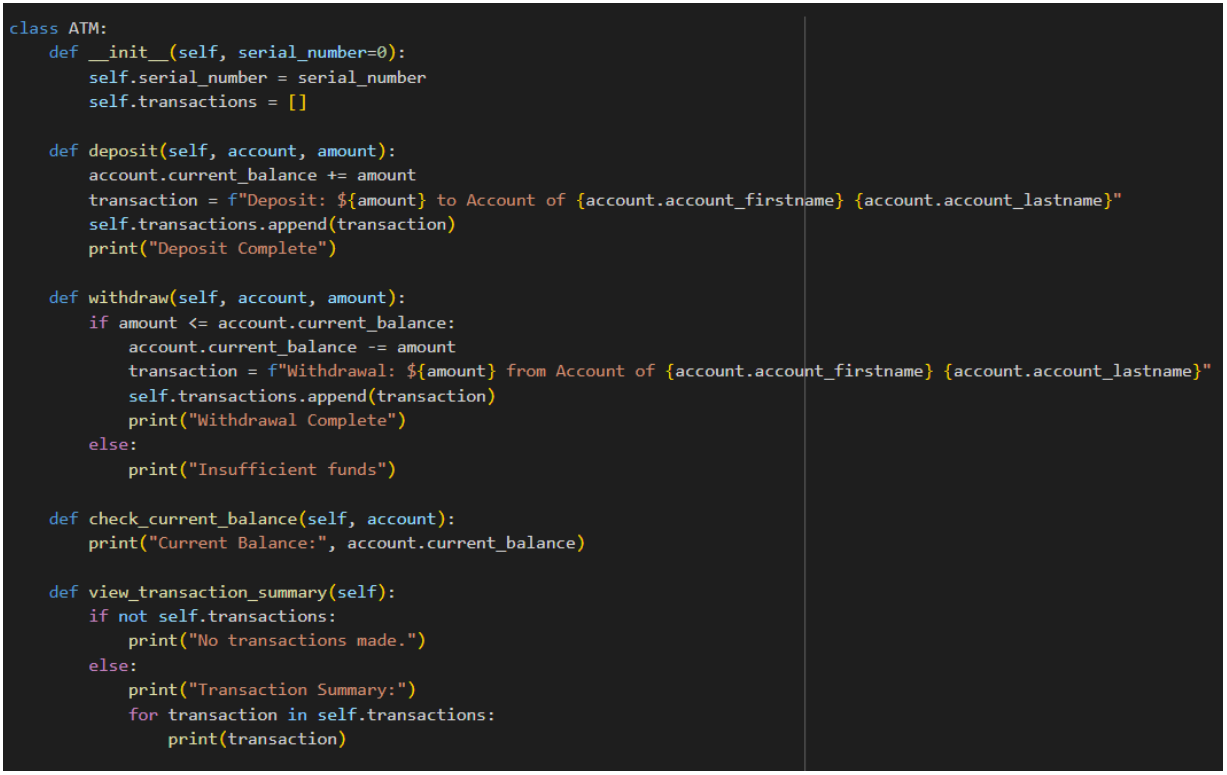


```python
Account1 = Accounts(firstname="Royce", lastname="Chua", balance=1000, address="Silver Street, Quezon City", email="roycechua123@gmail.com")
Account2 = Accounts(firstname="Cristine", lastname="Gura", balance=2000, address="Gold Street, Quezon City", email="cristinegura@yahoo.com")

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()

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()

atm_machine = ATM(serial_number=123456)

print("ATM Serial Number:", atm_machine.serial_number)

atm_machine.deposit(Account1, 500)
print("Account 1 Balance after deposit:", Account1.current_balance, )

atm_machine.withdraw(Account2, 300)
print("Account 2 Balance after withdrawal:", Account2.current_balance)

atm_machine.check_current_balance(Account1)
atm_machine.check_current_balance(Account2)
```



```
Account 1
First Name: Royce
Last Name: Chua
Current Balance: 1000
Address: Silver Street, Quezon City
Email: roycechua123@gmail.com

Account 2
First Name: Cristine
Last Name: Gura
Current Balance: 2000
Address: Gold Street, Quezon City
Email: cristinegura@yahoo.com

ATM Serial Number: 123456
Deposit Complete
Account 1 Balance after deposit: 1500
Withdrawal Complete
Account 2 Balance after withdrawal: 1700
Current Balance: 1500
Current Balance: 1700
```

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



```
class ATM:
    def __init__(self, serial_number=0):
        self.serial_number = serial_number
        self.transactions = []

    def deposit(self, account, amount):
        account.current_balance += amount
        transaction = f"Deposit: ${amount} to Account of {account.account_firstname} {account.account_lastname}"
        self.transactions.append(transaction)
        print("Deposit Complete")

    def withdraw(self, account, amount):
        if amount <= account.current_balance:
            account.current_balance -= amount
            transaction = f"Withdrawal: ${amount} from Account of {account.account_firstname} {account.account_lastname}"
            self.transactions.append(transaction)
            print("Withdrawal Complete")
        else:
            print("Insufficient funds")

    def check_current_balance(self, account):
        print("Current Balance:", account.current_balance)

    def view_transaction_summary(self):
        if not self.transactions:
            print("No transactions made.")
        else:
            print("Transaction Summary:")
            for transaction in self.transactions:
                print(transaction)
```

Questions

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

- A class is just a blueprint for constructing objects. It specifies a collection of properties and methods that the resulting objects will have.

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

- Classes are used in programming to organize and reuse code by encapsulating data and behavior into objects, hence increasing modularity and scalability. Sequential programs, on the other hand, use a clear, line-by-line execution style that is simpler but less adaptable to complicated tasks.

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

- Variables with the same name may exist in many contexts, such as within separate objects or classes. For example, account_firstname in one instance of a class may have a different value than in another instance. This is possible since each variable is contained within its own object or scope.

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?**

- When an object is formed, the constructor function initializes its class attributes. It is called automatically when a class instance is instantiated, and it sets the object's initial state with predetermined values or default settings.

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

- Constructors make object construction easier by automating attribute initialization, which eliminates the need for repetitious code. They ensure that all required properties are properly configured at the moment of object creation. This approach increases code readability and maintenance when compared to initializing variables individually in the main program.

7. Conclusion

- **Classes in Object-Oriented Programming serve as blueprints for constructing objects with specific features and methods, which improves modularity and code reuse. Furthermore, constructors make object creation easier by automatically initializing characteristics, which increases code readability and reduces repetitive work compared to manually establishing variables.**