

Laboratory Activity No. 1 - Introduction to Object-Oriented Programming

Rio, Aries, C.

09/14/24

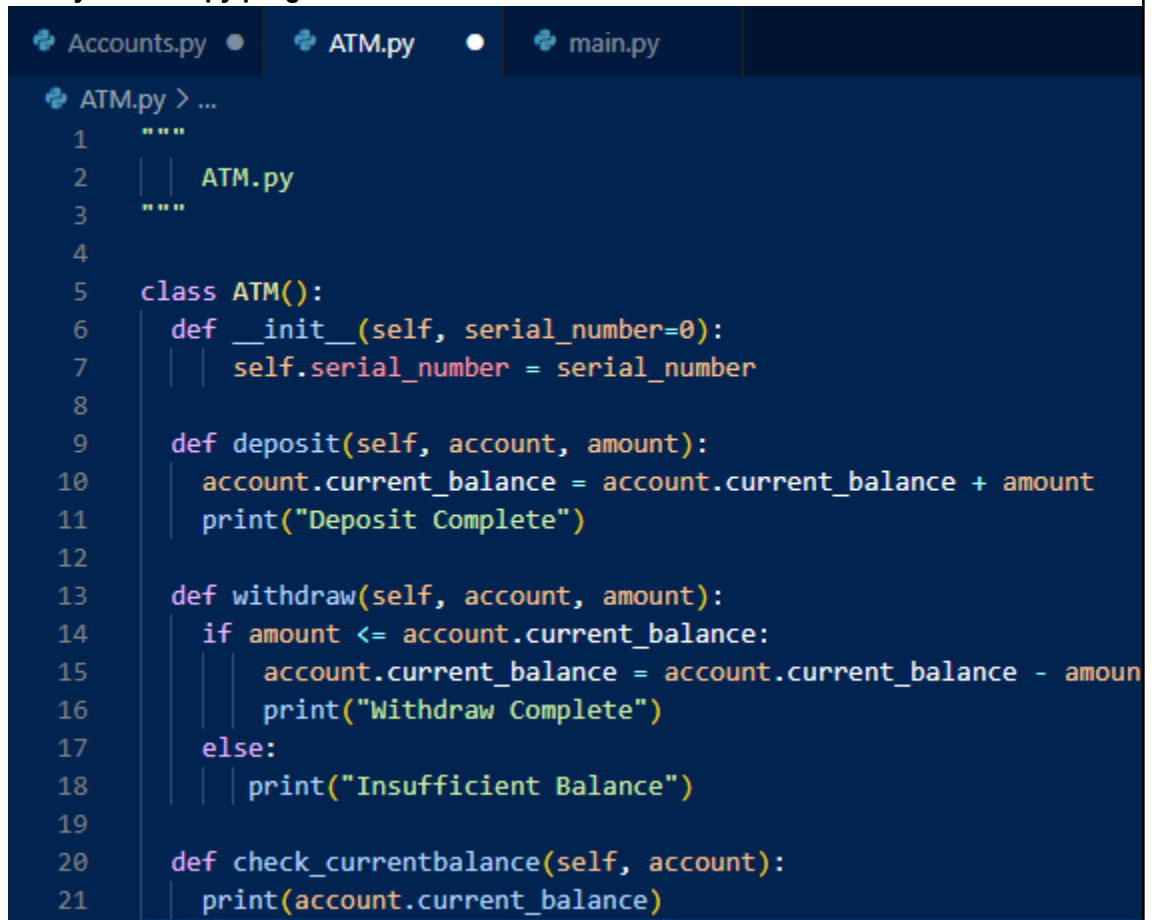
CPE009B / CPE21S4

Ma'am Sayo

6. Supplementary Activity

Tasks

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



```
1 '''
2 | | ATM.py
3 '''
4
5 class ATM():
6 | def __init__(self, serial_number=0):
7 | | self.serial_number = serial_number
8
9 | def deposit(self, account, amount):
10 | | account.current_balance = account.current_balance + amount
11 | | print("Deposit Complete")
12
13 | def withdraw(self, account, amount):
14 | | if amount <= account.current_balance:
15 | | | account.current_balance = account.current_balance - amount
16 | | | print("Withdraw Complete")
17 | | else:
18 | | | print("Insufficient Balance")
19
20 | def check_currentbalance(self, account):
21 | | print(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.

[illegible]

```
Account 1
Royce
Chua
1000
Silver Street Queon City
roycechua123@gmail.com
Deposit Complete
Current Balance: 1500

Serial Number: 987654321

Account 2
John
Doe
2000
Gold Street Quezon City
johndoe@yahoo.com
Deposit Complete
Current Balance: 2300

Serial Number: 12345678
```

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

Accounts.py ATM.py main.py

ATM.py > ...

```
1  """
2  |  ATM.py
3  """
4
5  class ATM():
6      def __init__(self, serial_number=0):
7          self.serial_number = serial_number
8          self.transactions = []
9
10     def deposit(self, account, amount):
11         account.current_balance = account.current_balance + amount
12         self.transactions.append(f"Deposit of {amount} to account {account.account_number}")
13         print("Deposit Complete")
14
15     def withdraw(self, account, amount):
16         if amount <= account.current_balance:
17             account.current_balance = account.current_balance - amount
18             self.transactions.append(f"Withdrawal of {amount} from account {account.account_number}")
19             print("Withdraw Complete")
20         else:
21             print("Insufficient Balance")
22
23     def check_currentbalance(self, account):
24         print(f"Current Balance: {account.current_balance}")
25
26     def view_transactionssummary(self):
27         print(f"Transaction Summary:{self.transactions}")
```

```
Account 1
Royce
Chua
1000
Silver Street Quezon City
roycechua123@gmail.com
Deposit Complete
Current Balance: 1500
Transaction Summary:['Deposit of 500 to account 123456']
```

```
Serial Number: 987654321
```

```
Account 2
John
Doe
2000
Gold Street Quezon City
johndoe@yahoo.com
Deposit Complete
Current Balance: 2300
Transaction Summary:['Deposit of 300 to account 654321']
```

```
Serial Number: 12345678
```

Questions

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

-It defines what properties (variables) and actions (methods) the objects will have. Think of it as a template for creating similar items, like a recipe for baking cookies where each cookie is an object.

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

-Classes are used in programs to keep things organized when dealing with complex data and actions. They help to group similar things together. On the other hand, sequential programs, which run line-by-line, are simpler and work well when the task is straightforward and doesn't need a lot of organization.

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?

-Even though the variables have the same name, each object created from the class gets its own version of these variables. This allows different objects (like two different accounts) to store different values for the same variable name.

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 constructor function's job is to set up (or initialize) the values for an object when it is first created. It runs automatically as soon as a new object is made from the class, so you don't have to manually set each value later.

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

-Using a constructor saves time and makes the code easier to manage. Instead of setting each variable one by one, the constructor sets all of them at once when you create the object. This makes the program cleaner and reduces the chance of mistakes.

7. Conclusion:

In conclusion, classes in object-oriented programming are a powerful tool for organizing code and managing complex programs. They allow you to group related data and behaviors together, making the code easier to understand and reuse. The constructor function simplifies the process of creating objects by automatically initializing their properties when the object is created, saving time and reducing errors. By using classes and constructors, programmers can write cleaner, more efficient code, which is especially useful for larger and more complex projects. For simpler tasks, line-by-line code may work, but classes become essential as the program grows in complexity.