

Activity Name # 1 - Class, Objects, Methods

Bona, Andrei Nycole So

09/15/24

CPE009B - CPE21S4

Prof. Maria Rizette Sayo

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_transactionssummary() method. The method should display all the transaction made in the ATM object.

```
2 class ATM:
3     def __init__(self, serial_number):
4         self.serial_number = serial_number
5         self.transactions = []
6
7     def deposit(self, account, amount):
8         account.current_balance += amount
9         transaction = f"Deposit: ₱{amount} to Account {account.account_number}"
10        self.transactions.append(transaction)
11        print("Deposit Complete")
12
13    def withdraw(self, account, amount):
14        if amount > account.current_balance:
15            print("Insufficient funds")
16            transaction = f"Failed Withdrawal: ₱{amount} from Account {account.account_number} (Insufficie
17        else:
18            account.current_balance -= amount
19            transaction = f"Withdrawal: ₱{amount} from Account {account.account_number}"
20            self.transactions.append(transaction)
21            print("Withdraw Complete")
22
23    def check_current_balance(self, account):
24        print("Current Balance:", account.current_balance)
25
26    def view_transaction_summary(self):
27        print("Transaction Summary:")
28        if not self.transactions:
29            print("No transactions have been made.")
30        for transaction in self.transactions:
31            print(transaction)
32
```

```
40 ATM1 = ATM.ATM(serial_number=919191)
41 ATM1.deposit(Account1, 500)
42 print("Account 1 balance after deposit:", Account1.current_balance)
43 ATM1.check_current_balance(Account1)
44 ATM1.withdraw(Account1, 100)
45 print("Account 1 balance after withdrawal:", Account1.current_balance)
46 ATM1.check_current_balance(Account1)
47 print("ATM Serial Number:", ATM1.serial_number)
48
49 print()
50 ATM1.view_transaction_summary()
51 print()
52
```

```

PS D:\Software\codes\python> & D:/Software/Python/python.exe d:/Software/codes/python/main.py
Account 1
First Name: Royce
Last Name: Chua
Balance: 1000
Address: Silver Street Quezon City
Email: roycechua123@gmail.com

Deposit Complete
Account 1 balance after deposit: 1500
Current Balance: 1500
Withdraw Complete
Account 1 balance after withdrawal: 1400
Current Balance: 1400
ATM Serial Number: 919191

Transaction Summary:
Deposit: P500 to Account 123456
Withdrawal: P100 from Account 123456

PS D:\Software\codes\python>

```

Questions

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

In object-oriented programming (OOP) a class serves as a blueprint or template that defines the structure of data and functions to operate on that data within an object context.

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

When a class is created, we can use it to create multiple objects. We can reduce the duplication of codes, it will inherit properties and behaviors from existing ones.

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 can have the name but different values when they are linked to objects or instances of a class, in programming languages like object oriented programming (OOP). Each object, in a class holds its set of attributes as defined by the class structure.

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 you create a class object, in Python using the `__init__` function (also called the constructor) it automatically triggers a process known as initialization. The primary task of this procedure is to set up the variables or attributes of the object at the time of its creation. In essence, the constructor ensures that the values of the object attributes are properly set when a new instance of the class is made.

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

Constructors simplify the process of defining object properties while providing control over the initialization process during object creation. Creating object properties is made easier and more

controlled by using constructors. They ensure that all properties are properly set during object creation. This approach enhances the structure of the code. Promotes reusability while reducing errors. Constructors offer an alternative to manually setting variables one by one as they encapsulate the initialization logic resulting in clearer and more concise code.

7. Conclusion:

In conclusion, classes and constructors are essential to object-oriented programming because they help automate object initialization, organize code, and encapsulate data. Constructor utilization improves reusability, minimizes errors, and makes the code clearer and more effective.

8. Assessment Rubric: