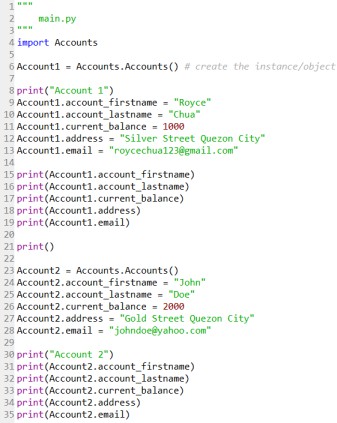
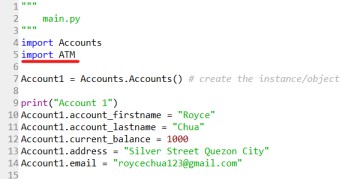
|  |  |
| --- | --- |
| **Laboratory Activity No. 3.1** | |
| **Introduction to Object-Oriented Programming** | |
| **Course Code:** CPE103 | **Program:** BSCPE |
| **Course Title:** Object-Oriented Programming | **Date Performed:** January 25,2025 |
| **Section:** 1-A | **Date Submitted:** January 26,2025 |
| **Name:** Polestico,Paul Justine D. | **Instructor:** Engr. Maria Rizette Sayo |
| **1. Objective(s):** | |
| This activity aims to familiarize students with the concepts of Object-Oriented Programming | |
| **2. Intended Learning Outcomes (ILOs):** | |
| The students should be able to:   * 1. Identify the possible attributes and methods of a given object   2. Create a class using the Python language   3. Create and modify the instances and the attributes in the instance. | |
| **3. Discussion:** | |
| Object-Oriented Programming (OOP) is an approach to programming that views the world and systems as consisting of objects that relate and interact with each other. This involves identifying the characteristics that describe the object which are known as the Attributes of the object. Furthermore, it also deals with identifying the possible capabilities or actions that an object is able to do which are called Methods.  An object is simply composed of Attributes and Methods wherein Attributes are variables that hold the information describing the object and Methods are functions which allow the object to perform its defined capabilities/actions. A UML Class Diagram is used to formally represent the collection of Attributes and Methods.  An example is given below considering a simple banking system.  Accounts ATM  + account\_number: int + serial\_number: int  + account\_firstname: string  + account\_lastname: string  + current\_balance: float  + address: string + deposit(account: Accounts, amount: int) + email: string + widthdraw(account: Accounts, amount: int) + update\_address(new\_address: string) + check\_currentbalance(account: Accounts) + update\_email(new\_email: string) + view\_transactionsummary() | |
| **4. Materials and Equipment:** | |

|  |
| --- |
| Desktop Computer with Anaconda Python/Python Colab Windows Operating System |
| **5. Procedure:** |
| **Creating Classes**   1. Create a folder named **OOPIntro\_LastName** 2. Create a Python file inside the **OOPIntro\_LastName** folder named **Accounts.py** and copy the code shown below: |

|  |
| --- |
| 1. Modify the Accounts.py and add ***self,*** before the new\_address and new\_email. 2. Create a new file named ATM.py and copy the code shown below:     **Creating Instances of Classes**   1. Create a new file named main.py and copy the code shown below: |

6.

Run the main.py program and observe the output. Observe the variables names account\_firstname, account\_lastname as well as other variables being used in the Account1 and Account2. 7. Modify the main.py program and add the code underlined in red.

8. Modify the main.py program and add the code below line 38.

|  |
| --- |
| 9. Run the main.py program.  **Create the Constructor in each Class**   1. Modify the Accounts.py with the following code:   Reminder: def init (): is also known as the constructor class   1. Modify the main.py and change the following codes with the red line. Do not remove the other codes in the program. |

|  |
| --- |
| 3. Run the main.py program again and run the output. |
| **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.   **Questions**   1. What is a class in Object-Oriented Programming?   A class,in my understanding,represents the attributes and methods that the user defined in their program.  It is like how a student represents their class,or called the representative,their skills and abilities.   1. Why do you think classes are being implemented in certain programs while some are sequential(line-by-line)?   Because classes are the representation of objects with similar characteristics and methods,while  Sequential,from the word itself,is the execution of code from one step to the next to maintain order.   1. How is it that there are variables of the same name such account\_firstname and account\_lastname that exist but have different values?   The firstname variable is declared outside the function,while lastname is declared inside |

|  |
| --- |
| The function.  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?  It stores all the attributes of objects with similar characteristics into one code,and is  Executed automatically to initialize those attributes when the code is run. |

|  |
| --- |
| 5. Explain the benefits of using Constructors over initializing the variables one by one in the main program?  Using constructors in initializing variables can help in organizing the code by putting all of  their characteristics and methods into one single line of code, making it cleaner and shorter. |
| **7. Conclusion:** |
| The class and constructor function are essential concepts in Object Oriented Programming,with each  serving their purposes. The Class function is what represents objects with similarities to each other.  an example would be motorcycles. The Constructor function arranges all these characteristics and  Methods all into one single package,just like how you put clothes into a drawer. These helped me know  how to arrange code into more comprehensible and neat order.  **Please refer to this link:**  [**https://github.com/PaulJustinePolestico/CPE-103-OOP-1-A/blob/main/Laboratory\_Activity\_3.ipynb**](https://github.com/PaulJustinePolestico/CPE-103-OOP-1-A/blob/main/Laboratory_Activity_3.ipynb) |
| **8. Assessment Rubric:** |