UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 1

Object-oriented Programming

Submitted by:
Bron, Jhustine A.

Instructor:
Engr. Maria Rizette H. Sayo

July 26, 2025

I. Objectives

This laboratory activity aims to implement the principles and techniques in object-oriented programming specifically through:

- Identifying object-orientation design goals
- Identifying the relevance of design pattern to software development

II. Methods

- Software Development
  - The design steps in object-oriented programming
  - Coding style and implementation using Python
  - Testing and Debugging
  - Reinforcement of below exercises

A. Suppose you are on the design team for a new e-book reader. What are the primary classes and methods that the Python software for your reader will need? You should include an inheritance diagram for this code, but you do not need to write any actual code. Your software architecture should at least include ways for customers to buy new books, view their list of purchased books, and read their purchased books.

B. Write a Python class, Polygons that has three instance variables of type str, int, and float, that respectively represent the name of the polygon, its number of sides, and its area. Your class must include a constructor method that initializes each variable to an appropriate value, and your class should include methods for setting the value of each type and retrieving the value of each type.

III. Results

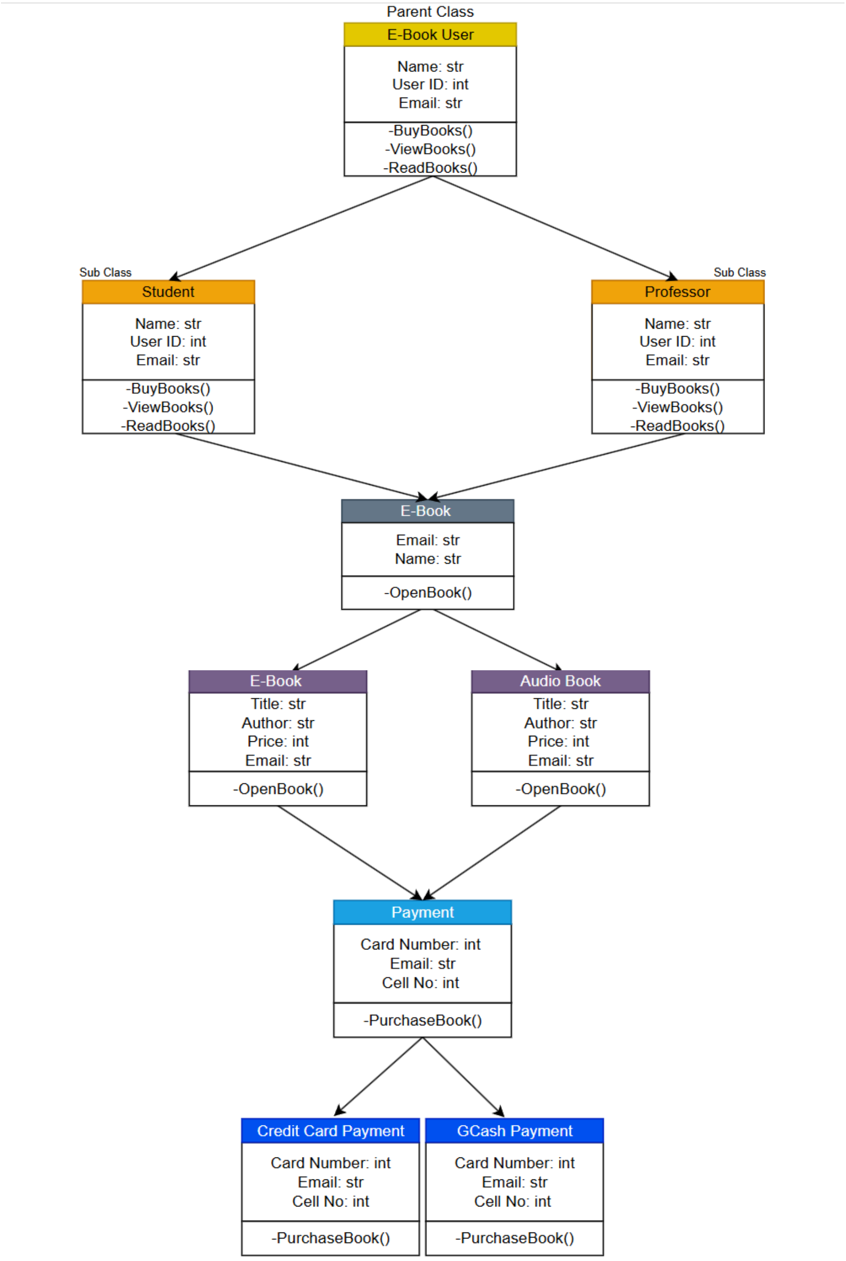

Figure 1. Flowchart of Task A

Insights:

This flowchart shows the inheritance diagram and its flow and how it inherits different attributes from the parent class up to the subclasses, it also shows the methods on each class.

Colab Link for Task B:
https://colab.research.google.com/drive/1O_iIffbtVE71vvjE4fVkit0O8vQl_QgR#scrollTo=8vrQNfIS-3HJ&line=7&uniqifier=1

Insights:

This program lets the user input the name, number of sides, and area of a polygon, then stores and displays that information. It uses a class to organize the data, checks if the inputs make sense (like at least 3 sides and a positive area), and makes the code easier to manage. It's a simple way to learn how object-oriented programming works in Python.

IV. Conclusion

In this activity, I designed a software architecture for an e-book reader using classes and inheritance, showing how components like E-Book User, Student, Professor, E-Book, Audio Book, and Payment work together with methods such as BuyBooks(), ViewBooks(), and ReadBooks() to allow users to purchase, view, and read books efficiently. The inheritance structure makes the design organized and easy to maintain. For Task B, I created a Polygons class in Python that includes a constructor, getter and setter methods, and input validation to ensure correct values for the polygon's name, number of sides, and area. These tasks helped me understand object-oriented programming concepts such as encapsulation, inheritance, and validation, which are essential for creating structured and reliable programs in real-world applications.