



UNIVERSITY OF CALOOCAN CITY  
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Activity No. 1

---

# Object-oriented Programming

---

*Submitted by:*  
Palmes, Lewis Clark L.

*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 patterns to software development

## II. Methods

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

Present the visualized procedures done. Also present the results with corresponding data visualizations such as graphs, charts, tables, or image. Please provide insights, commentaries, or explanations regarding the data. If an explanation requires the support of literature such as academic journals, books, magazines, reports, or web articles please cite and reference them using the IEEE format.

## For activity A: Inheritance Diagram

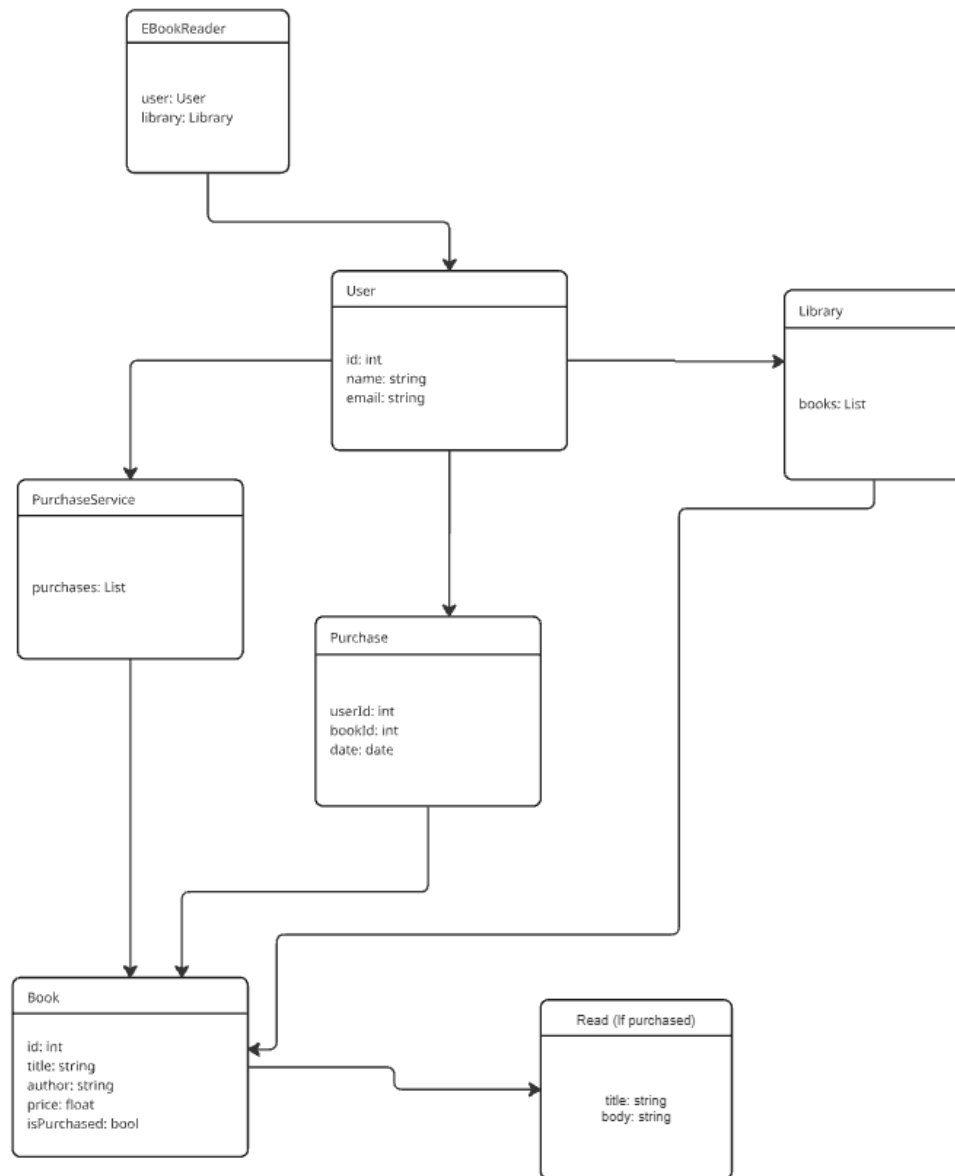


Figure 1 Screenshot of Inheritance Diagram

This diagram represents an eBook system where users can browse a library, purchase books, and read them. Users interact with an eBookReader that connects to a library of books. Purchases are handled through the PurchaseService, which records each transaction. Once a book is purchased, the user can read its content. The system is structured around entities like User, Book, Purchase, and Library, with clear relationships linking purchases to books and users, ensuring that reading access is granted only after a purchase.

## For activity B: Polygon Class



```
class Polygons:
    def __init__(self, name: str, sides: int, area: float):
        self._name = name
        self._sides = sides
        self._area = area

    def set_name(self, name: str):
        self._name = name

    def set_sides(self, sides: int):
        self._sides = sides

    def set_area(self, area: float):
        self._area = area

    def get_name(self) -> str:
        return self._name

    def get_sides(self) -> int:
        return self._sides

    def get_area(self) -> float:
        return self._area

hexagon = Polygons("Hexagon", 6, 93.5)

print(hexagon.get_name())
print(hexagon.get_sides())
print(hexagon.get_area())

hexagon.set_area(100.0)
print(hexagon.get_area())
```

Hexagon  
6  
93.5  
100.0

Figure 2 Screenshot of Polygon Class

[LAB 1.ipynb - Colab](#)

The code defines a Polygons class with instance variables for the polygon's name, number of sides, and area. It includes setter and getter methods for each variable, allowing you to set and retrieve values. The example creates a Polygons object representing a hexagon with 6 sides and an area of 93.5. The getter methods are used to print its attributes, and the setter method `set_area()` is used to update the area to 100.0. The output confirms the correct retrieval and modification of the polygon's attributes.

## IV. Conclusion

This laboratory activity helped us better understand and apply key object-oriented programming (OOP) concepts. By designing an eBook reader Inheritance Diagram, we explored how classes and relationships work together to create a functional system for users to purchase, view, and read books. The Polygons class exercise also showed us how to structure data and use methods to manage and modify it effectively. Overall, the activity emphasized the importance of good design and organization in building software, making it easier to maintain and expand in the future.

## **References**

- [1] Co Arthur O.. “University of Caloocan City Computer Engineering Department Honor Code,” UCC-CpE Departmental Policies, 2020.