



UNIVERSITY OF CALOOCAN CITY  
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Activity No. 1

---

# Object-oriented Programming

---

*Submitted by:*  
Delinia, Filjohn B.

*Instructor:*  
Engr. Maria Rizette H. Sayo

07, 26, 2025

## I. Objectives

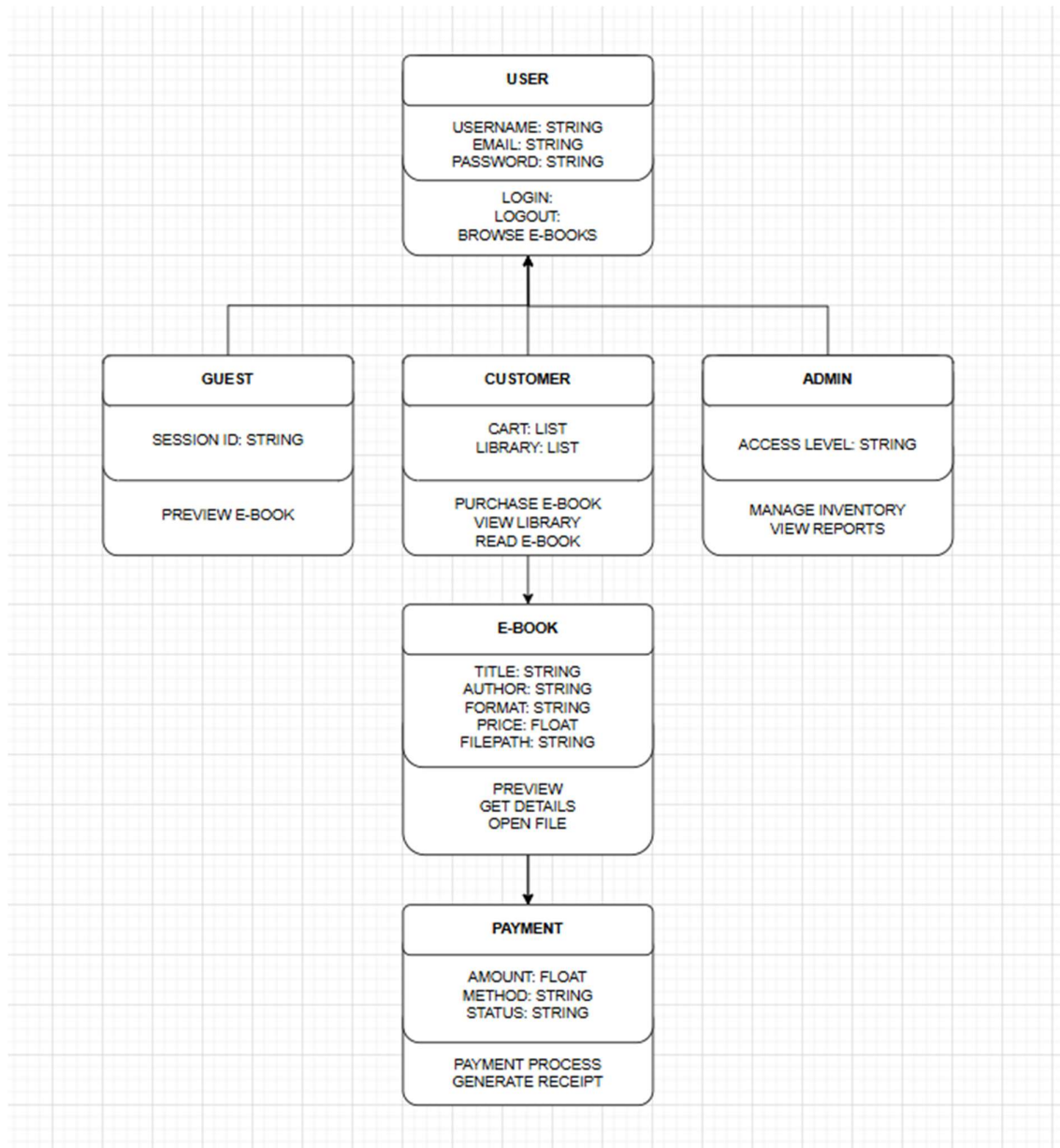
This laboratory activity aims to implement the principles and techniques in object-oriented programming specifically through:

- 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



This class diagram outlines an e-book management system with a base **User** class, inherited by **Guest**, **Customer**, and **Admin**—each with role-specific attributes and actions. Guests can only preview e-books, Customers can purchase and access their library, and Admins manage inventory and view reports. The **E-Book** class contains details like title and file path, while the **Payment** class handles transaction data and receipt generation. Together, these classes support user interaction, content management, and financial processing in a structured and scalable system.

```
class Polygon:
    def __init__(self, name, sides, area):
        if sides < 3:
            raise ValueError("A polygon must have at least 3 sides.")
        if area <= 0:
            raise ValueError("Area must be a positive value.")

        self.name = name
        self.sides = sides
        self.area = area

    def display(self):
        print("\nPolygon Details:")
        print("Name:", self.name)
        print("Number of Sides:", self.sides)
        print("Area:", self.area)

try:
    name = input("Enter the name of the polygon: ")
    sides = int(input("Enter the number of sides: "))
    area = float(input("Enter the area of the polygon: "))

    polygon = Polygon(name, sides, area)
    polygon.display()

except ValueError as e:
    print("Error:", e)
```

Enter the name of the polygon: square  
Enter the number of sides: 4  
Enter the area of the polygon: 10

Polygon Details:  
Name: square  
Number of Sides: 4  
Area: 10.0

[https://colab.research.google.com/drive/1WONDYd\\_32QudRwTHmrETKQ-rA2UVywNg#scrollTo=c8omUvoGwsYl&line=11&uniqifier=1](https://colab.research.google.com/drive/1WONDYd_32QudRwTHmrETKQ-rA2UVywNg#scrollTo=c8omUvoGwsYl&line=11&uniqifier=1)

This Python program defines a Polygon class with attributes for name, sides, and area. It checks that sides are at least 3 and area is positive, raising errors if not. The user inputs values, and if valid, the program displays the polygon's details otherwise, it shows an error message.

## IV. Conclusion

In conclusion, this lab exercise gave me invaluable practical experience using the concepts of object-oriented programming in real-world situations. I witnessed the successful application of abstraction, inheritance, and user interaction in a software system through creating class structures and input validation procedures. In addition to reinforcing theoretical ideas, this exercise highlighted how crucial logical design and implementation are to building useful and maintainable applications. All things considered, it was a significant step in improving my programming and system modeling skills.

## References

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