UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 1

Object-oriented Programming

Submitted by:
Regondola, Jezreel P.

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
  - o The design steps in object-oriented programming
  - o Coding style and implementation using Python
  - o Testing and Debugging
  - o 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

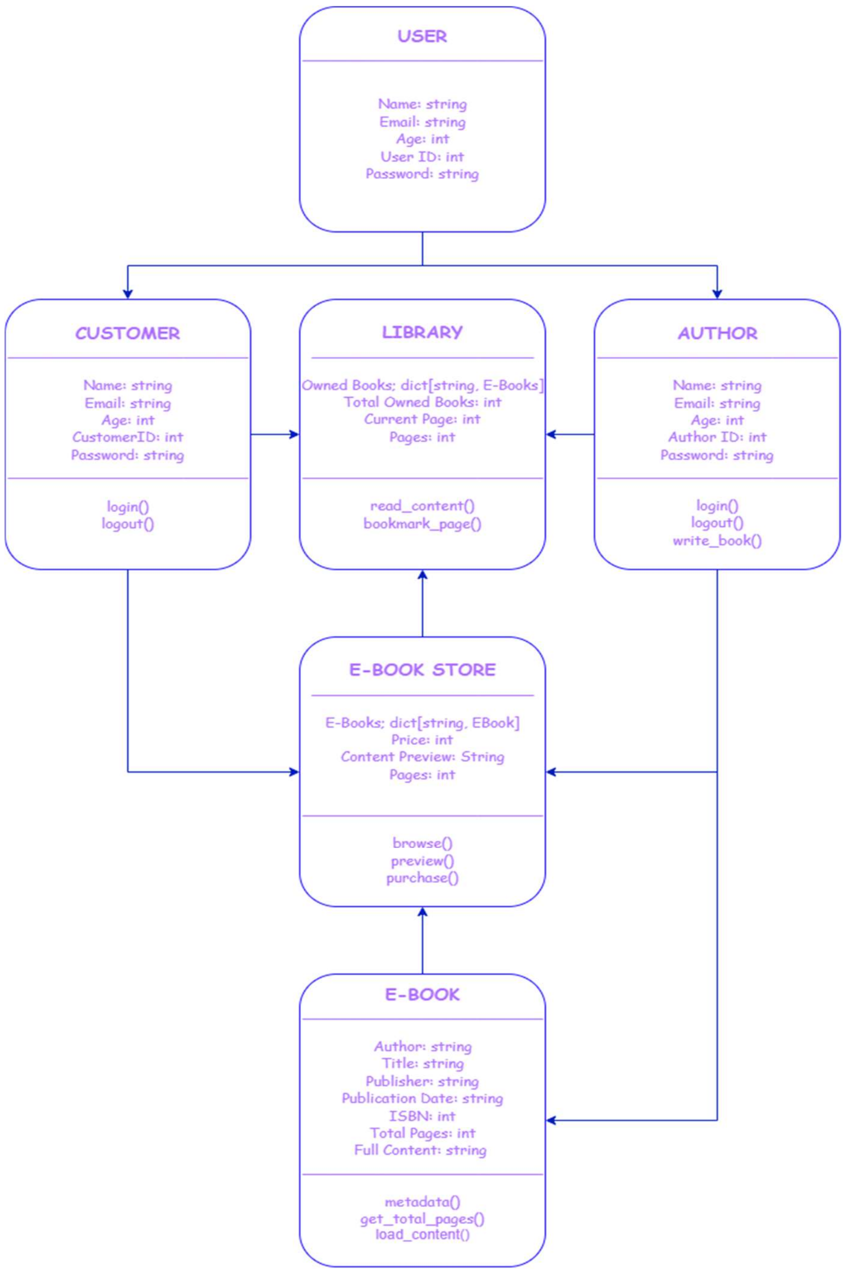
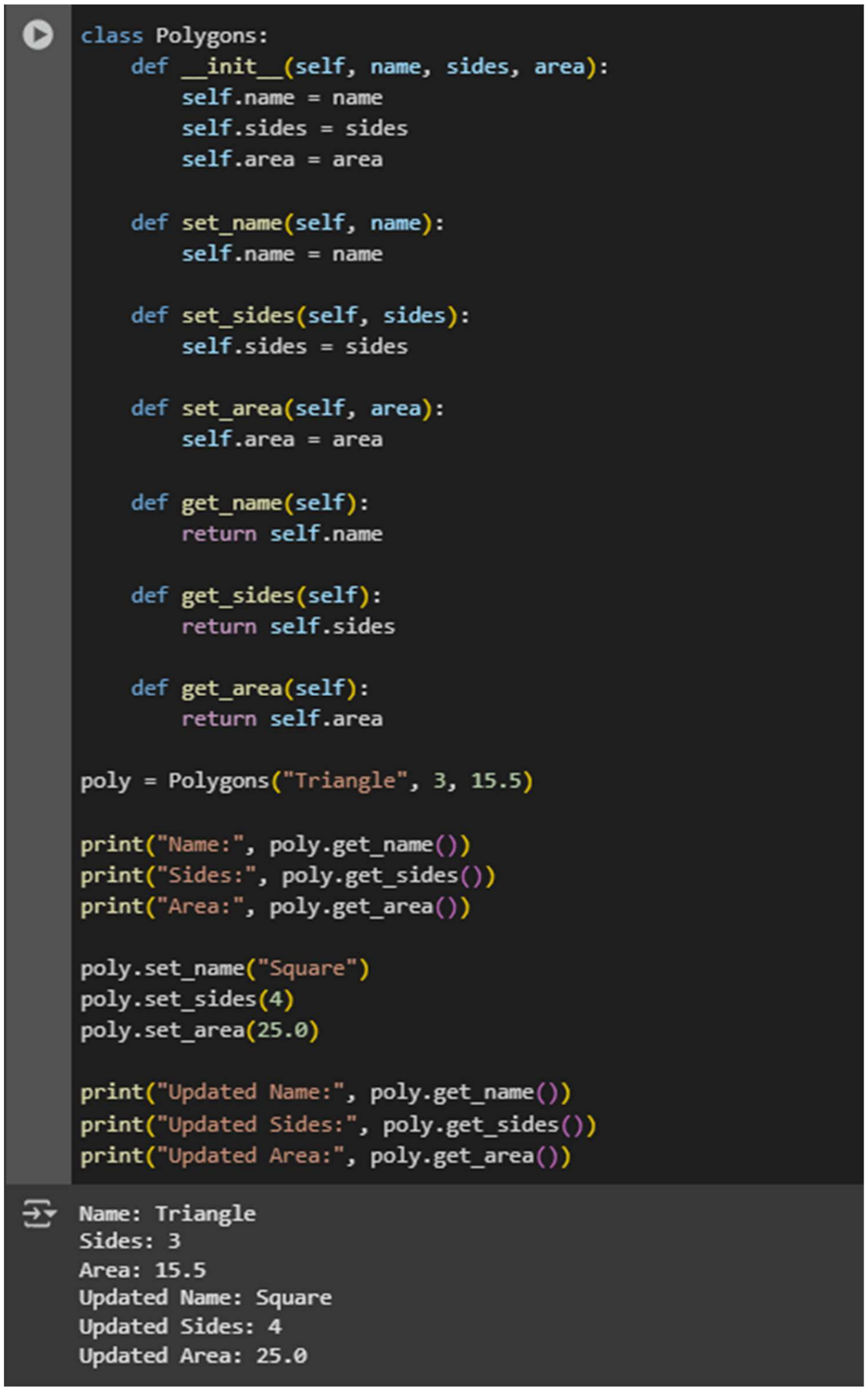


Figure 1 Image of diagram

This diagram shows an e-book system where authors write books and add them to the store. Customers buy books from the store and keep them in their library, where they can read and bookmark them. Each e-book has details like title, author, and content.



```python
class Polygons:
    def __init__(self, name, sides, area):
        self.name = name
        self.sides = sides
        self.area = area

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

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

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

    def get_name(self):
        return self.name

    def get_sides(self):
        return self.sides

    def get_area(self):
        return self.area

poly = Polygons("Triangle", 3, 15.5)

print("Name:", poly.get_name())
print("Sides:", poly.get_sides())
print("Area:", poly.get_area())

poly.set_name("Square")
poly.set_sides(4)
poly.set_area(25.0)

print("Updated Name:", poly.get_name())
print("Updated Sides:", poly.get_sides())
print("Updated Area:", poly.get_area())
```

```
Name: Triangle
Sides: 3
Area: 15.5
Updated Name: Square
Updated Sides: 4
Updated Area: 25.0
```

Figure 2 Screenshot of program

The program creates a class Polygons with three attributes: name (string), sides (integer), and area (float). The constructor sets these values when an object is created. The class also has setter methods to update each attribute and getter methods to access them.

IV. Conclusion

This activity applied object-oriented programming by making a class with attributes and methods. It showed how OOP helps organize code and how design patterns make programs easier to understand and maintain.

References

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