UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 1

Object-oriented Programming

Submitted by:
Mamanao, Kurt Marwin, C.

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

A. In this section, the diagram below shows the structures of the design for the e-book reader.

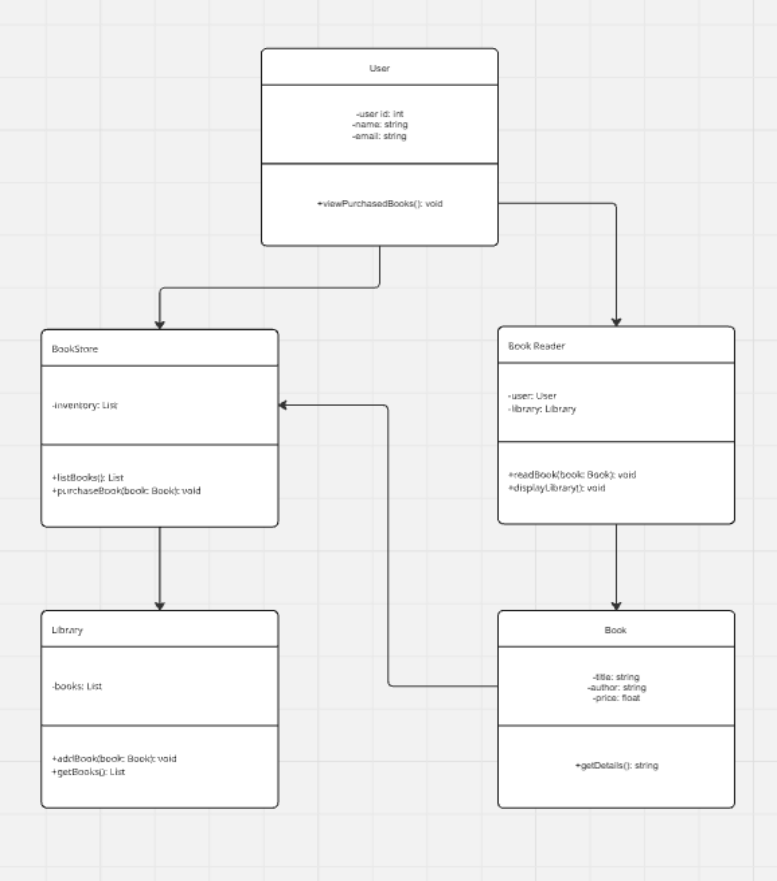

Figure 1 Diagram for e-book reader

The class diagram shows how the eBook reader system is organized. The **User** buys books from the **Book Store**, which manages available titles. Purchased books go into the **Library**, and the **Book Reader** lets users read them. Each class has a clear role, keeping the system organized and easy to manage.

B.

The Python Class was created to represent the properties of a polygon. It includes the attributes for the polygon's name, number of sides, and areas. The figures below show how the class is used to set and retrieve these values.

```python
class Shape:
    def __init__(self, shape_name: str, num_sides: int, shape_area: float):
        self._shape_name = shape_name
        self._num_sides = num_sides
        self._shape_area = shape_area

    def rename(self, new_name: str):
        self._shape_name = new_name

    def change_sides(self, sides: int):
        self._num_sides = sides

    def update_area(self, area: float):
        self._shape_area = area

    def name(self) -> str:
        return self._shape_name

    def sides(self) -> int:
        return self._num_sides

    def area(self) -> float:
        return self._shape_area
```

Figure 2 Polygon Class

```python
print("Polygon Name:", octagon.name())
print("Number of Sides:", octagon.sides())
print("Area:", float(octagon.area()))
```

Figure 3 Example

In the given code, the object octagon is created from the Polygon class using the values "octagon" for the name, 8 for the number of sides, and 60.0 for the area. These values are passed as arguments to the constructor method __init__() and are then stored as attributes of the octagon object. This allows the object to represent a specific polygon and enables access to its details using the class's getter methods.

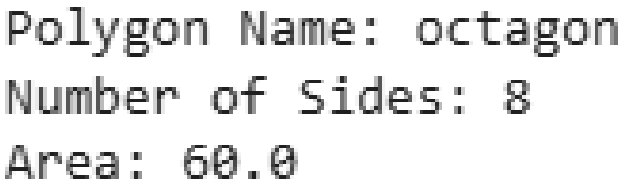

Figure 4 Result

In Figure 4, the output confirms that the object was successfully created and that the values were correctly stored and retrieved. This demonstrates that the class functions as intended, allowing the stored data to be accessed through its defined method.

IV. Conclusion

V.

In this lab activity, I learned about making inheritance diagram even though I don't know if it's right and then the eBook reader system is designed in a clear and organized way. Each part has its own job, making it easy to use and manage. Also, the Polygon example shows how Python classes can store information and let us get that information easily, which means the class works the way it should.

References

[1]Real Python, "Python classes: The power of object-oriented programming," *Real Python*. https://realpython.com/python-classes/ (accessed Jul. 26, 2025).

[2]Readest, "Readest: An open-source ebook reader project," *GitHub*. https://github.com/readest/readest (accessed Jul. 26, 2025)