Data Structure and Algorithm

Laboratory Activity No. 1

Object-oriented Programming

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Luminario, Venice Lou Gabrielle M. | Engr. Maria Rizette H. Sayo |

July 26, 2025

# 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

# Methods

* Software Development
  + The design steps in object-oriented programming
  + Coding style and implementation using Python
  + Testing and Debugging
  + Reinforcement of below exercises
  1. 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.
  2. 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.

# Results

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

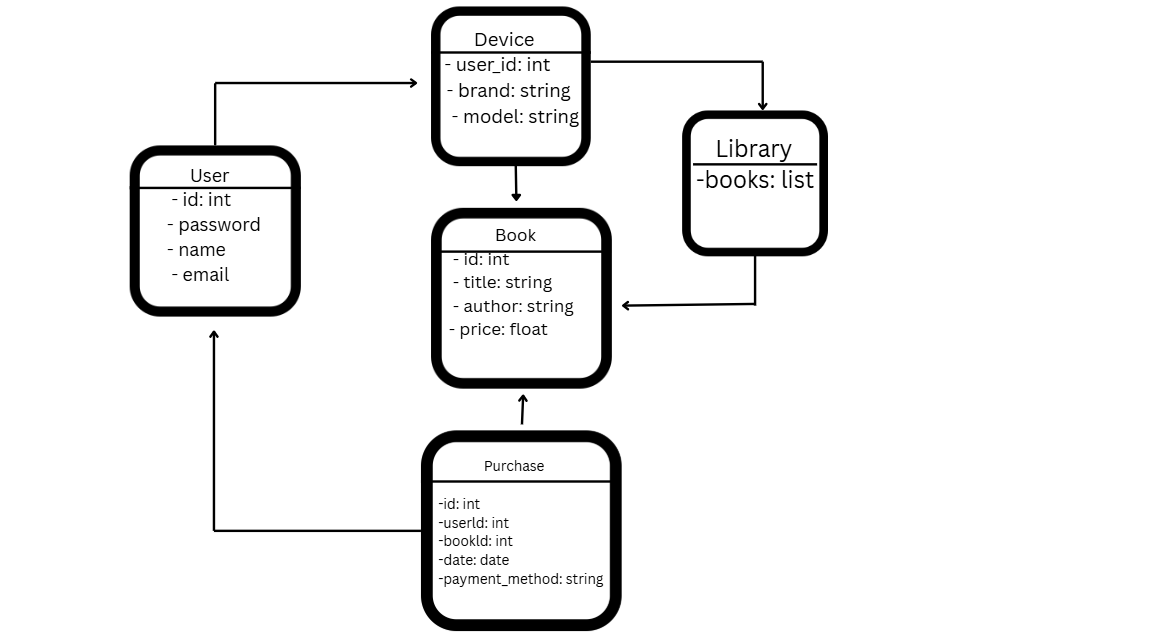


Figure 1 Diagram for E-Book Reader

The class diagram outlines how different parts of the eBook reader system work together. The User class represents individuals who can purchase and access books. The Book class holds information about each book, while the Purchase class connects users and books, recording transactions. The library stores the books that users have access to, and the Shop (or eBook platform) serves as the main environment where users interact with the system. Each class is designed to focus on a specific role, making the system organized and easy to manage. The relationships between classes clearly show how data flows from purchasing to storing and reading books.

* 1. The Python Class was created to represent the properties of a quadrilateral. 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.

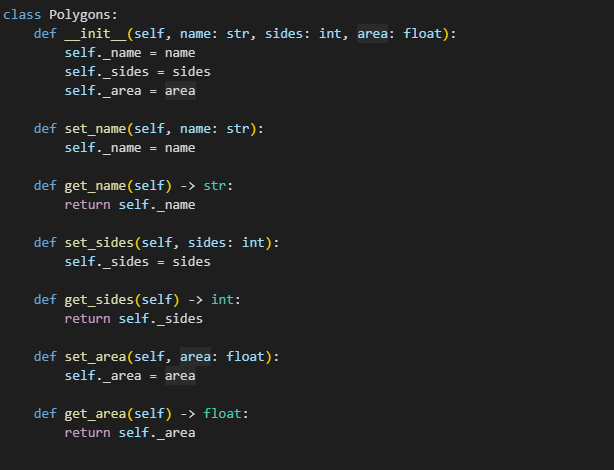
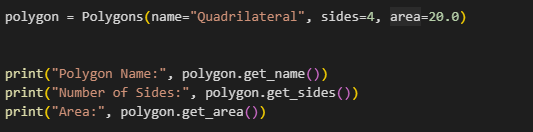


Figure 2 Polygon Class

Figure 3 Example

In figure 3, the object quadrilateral is created from the Polygons class with the values “Quadrilateral” as the name, 4 as the number of sides, and 20.0 as the area. These values are passed to the constructor method in and stored in the object.

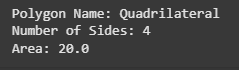


Figure 4 Result

In figure 4, the output indicated that the object was successfully created and the values were correctly stored and retrieved. It shows that the class is working as expected by allowing data to be accessed using a method.

# Conclusion

In this laboratory activity, we performed to create useful software. Designing the e-book reader helped us understand how to organize different parts of a program using classes and how they work together. Writing the Quadrilateral class showed us how to create objects with specific properties and how to set and get their values. This activity helped us see how important good design and coding practices are when building programs.

**References**

**- Ossher, H., & Tarr, P. (2001). Using multidimensional separation of concerns to (re)shape evolving software. *Communications of the ACM, 44*(10), 43–50.** [**https://doi.org/10.1145/383845.383856**](https://doi.org/10.1145/383845.383856)[**arXiv+2ResearchGate+2Stack Overflow+2**](https://www.researchgate.net/publication/221610595_Practical_Use_of_Encapsulation_in_Object-Oriented_Programming?utm_source=chatgpt.com)[**Software Engineering Stack Exchange+13ACM+13ACM Digital Library+13**](https://cacmb4.acm.org/magazines/2001/10/7252-using-multidimensional-separation-of-concerns-to-reshape-evolving-software/fulltext?utm_source=chatgpt.com)

**- *Encapsulation (computer programming)*. (2025, June). *Wikipedia*. Retrieved from** [**https://en.wikipedia.org/wiki/Encapsulation\_(computer\_programming)**](https://en.wikipedia.org/wiki/Encapsulation_(computer_programming))[**Medium+2Wikipedia+2Stack Overflow+2**](https://en.wikipedia.org/wiki/Encapsulation_%28computer_programming%29?utm_source=chatgpt.com)

**- *Information hiding*. (2024, April). *Wikipedia*. Retrieved from** [**https://en.wikipedia.org/wiki/Information\_hiding**](https://en.wikipedia.org/wiki/Information_hiding)[**ResearchGate+3Wikipedia+3Wikipedia+3**](https://en.wikipedia.org/wiki/Information_hiding?utm_source=chatgpt.com)

**- *Modular programming*. (2025, July). *Wikipedia*. Retrieved from** [**https://en.wikipedia.org/wiki/Modular\_programming**](https://en.wikipedia.org/wiki/Modular_programming)[**arXiv+2Wikipedia+2Reddit+2**](https://en.wikipedia.org/wiki/Modular_programming?utm_source=chatgpt.com)

**- Shahzad, U. (2013, August 18). Why encapsulation is an important feature of OOP languages? *Stack Overflow*. Retrieved from** [**https://stackoverflow.com/questions/18300953/why-encapsulation-is-an-important-feature-of-oop-languages**](https://stackoverflow.com/questions/18300953/why-encapsulation-is-an-important-feature-of-oop-languages)[**Beginners Coding 101+2Stack Overflow+2ResearchGate+2**](https://stackoverflow.com/questions/18300953/why-encapsulation-is-an-important-feature-of-oop-languages?utm_source=chatgpt.com)