Data Structure and Algorithm

Laboratory Activity No. 1

Object-oriented Programming

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Enverzo, Kyle Andrey D. | Engr. Maria Rizette H. Sayo |

07-26-2025

# 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

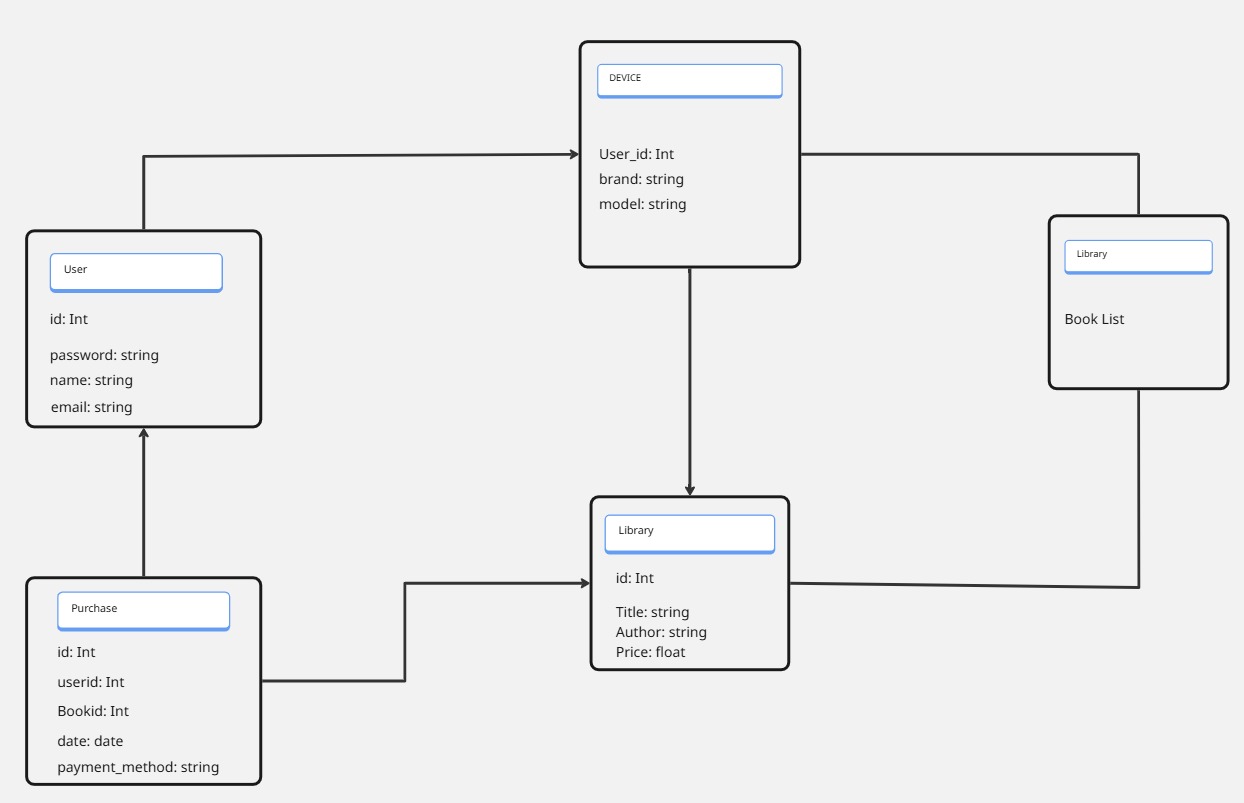
# Methods

* 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 illustrates the interactions between the different parts of the eBook reader system. The User class represents those who have the ability to purchase and access books. Each book's details are stored in the Book class, and transactions are recorded by the Purchase class, which links users and books. The Shop (or eBook platform) is the primary setting where users engage with the system, while the Library houses the books that customers can access. Because each class is made to concentrate on a certain role, the system is well-organized and simple to use. The connections between classes make it evident how information moves from book purchases to book storage and reading.

B. To represent a polygon's attributes, the Python Class was developed. It contains the properties for the name, areas, and number of sides of the polygon. The class is used to set and get these values, as seen in the images below.

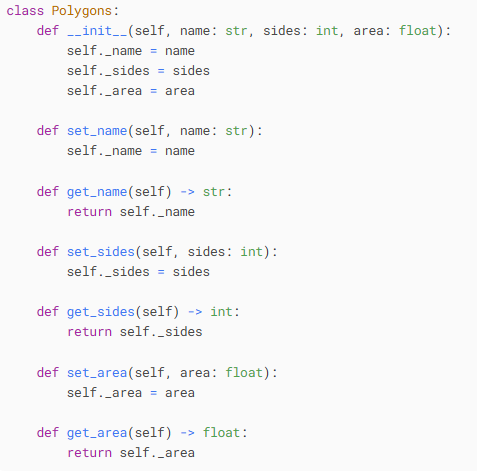


Figure 2 Polygon Class

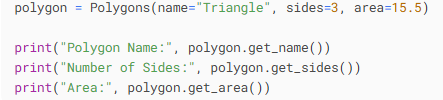


Figure 3 Example

The Polygons class is used to create the object polygon in Figure 3, which has the name "Triangle," three sides, and an area of 15.5. These values are saved in the object after being supplied to the constructor method \_\_in\_\_.

.



Figure 4 Result

The result in figure 4 showed that the object had been formed properly and that the values had been stored and retrieved appropriately. By permitting data to be retrieved via a method, it demonstrates that the class is operating as intended.

.

# Conclusion

Our goal in this lab exercise was to develop practical software. We learned how to use classes to arrange various program components and how they interact with one another through the design of the e-book reader. We learned how to build objects with particular properties and how to set and get their values by writing the Polygons class. This exercise made it clear to us how crucial sound design and coding techniques are when creating software.

.

**References**

[Untitled1.ipynb - Colab](https://colab.research.google.com/drive/1HPkCkqiKdm3Nw00-o_Xe7O-8cTyT-VLM#scrollTo=4jjWURhxkJP8)