Data Structure and Algorithm

Laboratory Activity No. 1

Object-oriented Programming

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Polestico,Paul Justine D. | 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

Present the visualized procedures done. Also present the results with corresponding data visualizations such as graphs, charts, tables, or image . Please provide insights, commentaries, or explanations regarding the data. If an explanation requires the support of literature such as academic journals, books, magazines, reports, or web articles please cite and reference them using the IEEE format.

Please take note of the styles on the style ribbon as these would serve as the style format of this laboratory report. The body style is Times New Roman size 12, line spacing: 1.5. Body text should be in Justified alignment, while captions should be center-aligned. Images should be readable and include captions. Please refer to the sample below:

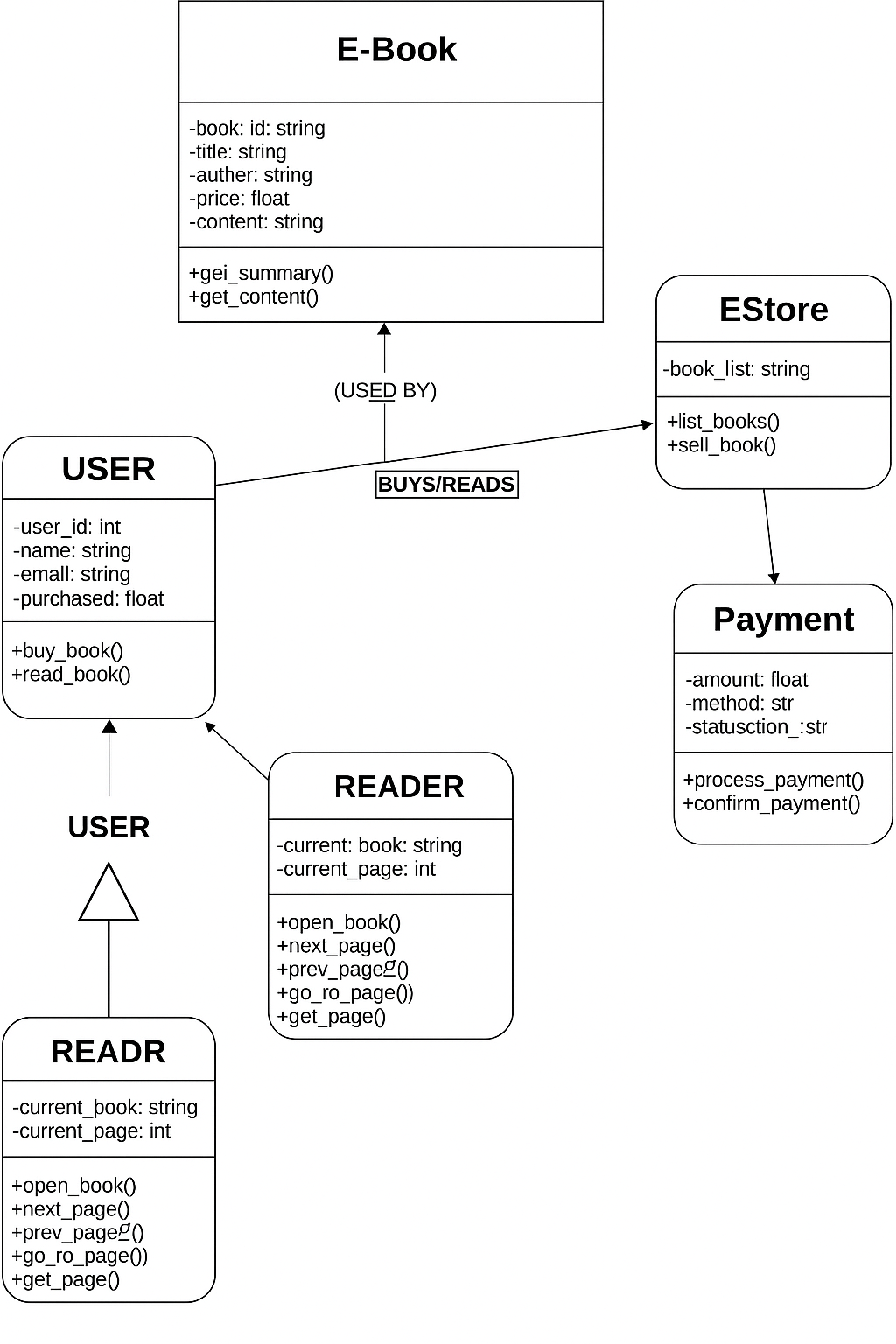
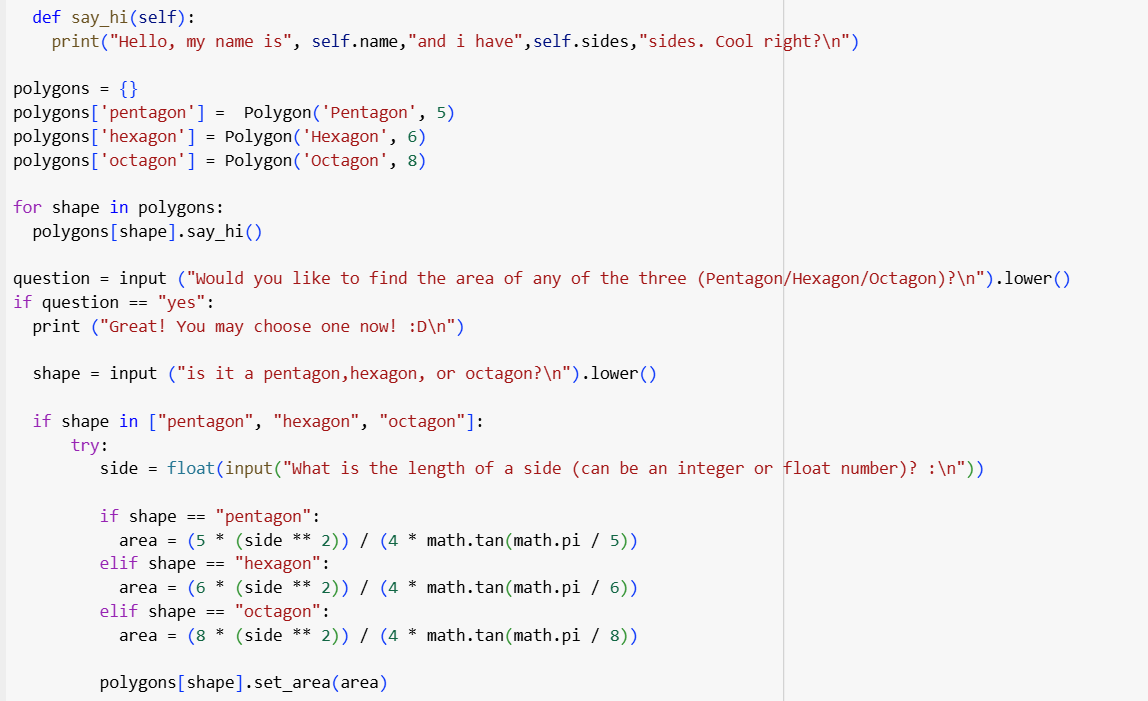


Figure 1 Diagram of the E-Book Reader

This diagram demonstrates the structure of the E-Book Reader system. The E-Book class allows users to obtain the summary and content of the e-book they selected. The User class provides core functionalities such as buying. selling, and reading e-books that the user owns. The reader class inherits attributes from the user class and focuses specifically on the reading process, including user identification for access control. Lastly, The EStore class enables the user to search for e-books, sell, and buy them through the help of the Payment class, which is responsible for handling the payment process.





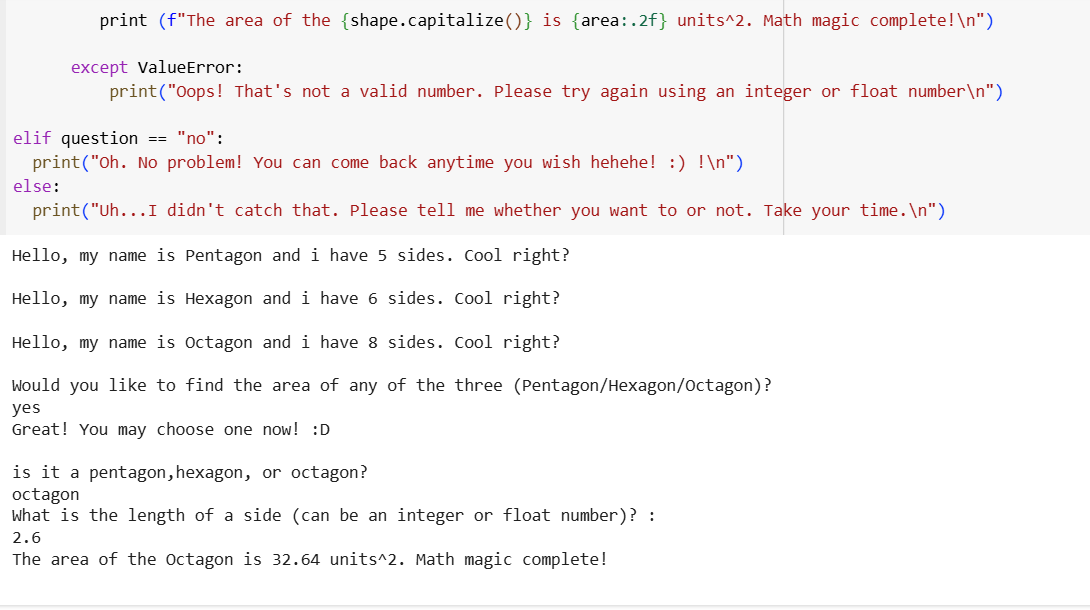


Figure 2 Program with application of Get and Set Functions

This program demonstrates encapsulation through the use of get and set functions. The get methods retrieve the values of the instance variables, while the set methods assign or update those values. These values are then used in output strings to display relevant information to the user.

All polygon objects are stored in a dictionary for better organization and more efficient access. The program then prompts the user to determine whether they want to calculate the area of one of the three polygons (Pentagon, Hexagon, Octagon), using if-else statements for flow control.

If the user proceeds, the program calculates the area based on the formula for regular polygons. The try-except block handles invalid inputs to ensure the program doesn't crash due to incorrect data types.

If no errors are encountered, the program proceeds to perform the calculation and display the result. This showcases the integration of all the components — encapsulation, data organization, user interaction, and error handling — working together to complete the task smoothly.

# Conclusion

In the midst of making the E-Book Reader diagram,i have realized that design pattern in relation to software development is essential for developing programs that are concise and intuitive to the user. With the help of diagrams,developers would have a comprehensive blueprint that can be followed and modified according to specific needs. Together with the fundamentals of Object-Oriented programming , having a diagram significantly helps in executing plans essential for helping people,whether in everyday life or in natural and manmade disasters.

**References**

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

[2] “Create UML class diagrams” draw.io

<https://www.drawio.com/blog/uml-class-diagrams> (accessed July 26, 2025).

[3]Programiz, “Python if, if...else, if...elif...else and Nested if Statement,” *Programiz.com*, 2019. <https://www.programiz.com/python-programming/if-elif-else> (accessed July 26, 2025)

[4]W3schools, “Python Dictionaries,” *W3schools.com*, 2018. <https://www.w3schools.com/python/python_dictionaries.asp> (accessed July 27, 2025)

‌

[5] “Python \_\_init\_\_() Function,” *www.w3schools.com*. <https://www.w3schools.com/python/gloss_python_class_init.asp> (accessed July 26, 2025)

‌

[6]W3 Schools, “Python Try Except,” *www.w3schools.com*, 2024. <https://www.w3schools.com/python/python_try_except.asp> (accessed July 27, 2025)

‌

‌