UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 1

Object-oriented Programming

Submitted by:
Villacin, Justine R.

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 patterns 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 the exercises below

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

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:

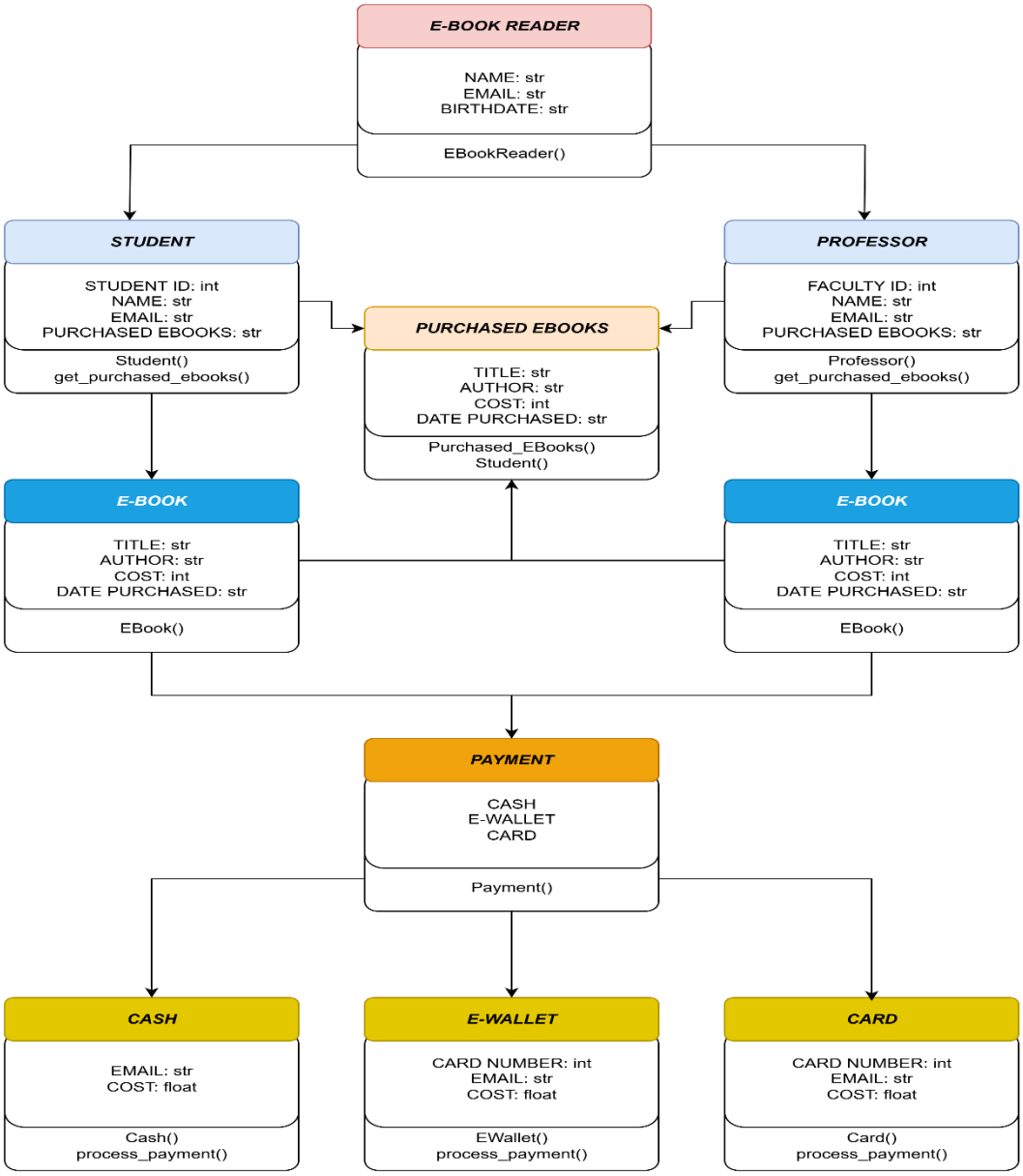

Figure 1 Screenshot of program

If an image is taken from another literature or intellectual property, please cite them accordingly in the caption. Always keep in mind the Honor Code [1] of our course to prevent failure due to academic dishonesty.

```python
class Polygons:
    def __init__(self, name = "", sides = 0, area = 0.0):
      self.name = name
      self.sides = sides
      self.area = area

    def get_name(self):
      return self.name

    def get_sides(self):
      return self.sides

    def get_area(self):
      return self.area

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

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

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

if __name__ == "__main__":

   shape = Polygons()
   print("Default Polygon: ")
   print(f"Name: {shape.get_name()}, Sides: {shape.get_sides()}, Area: {shape.get_area()}")

   shape.set_name("Triangle")
   shape.set_sides(3)
   shape.set_area(15.5)

   print("\nUpdated Polygon: ")
   print(f"Name: {shape.get_name()}, Sides: {shape.get_sides()}, Area: {shape.get_area()}")

   shape.set_name("Square")
   shape.set_sides(4)
   shape.set_area(16.0)

   print("\nNew Polygon:")
   print(f"Name: {shape.get_name()}, Sides: {shape.get_sides()}, Area: {shape.get_area()}")
```

```
Default Polygon:
Name: , Sides: 0, Area: 0.0

Updated Polygon:
Name: Triangle, Sides: 3, Area: 15.5

New Polygon:
Name: Square, Sides: 4, Area: 16.0
```

IV. Conclusion

This laboratory activity helped me understand the key concepts of object-oriented programming (OOP) in Python. I learned how to design classes, use inheritance, and apply getter and setter methods to manage data. The first exercise made me think about real-world software design, like an e-book reader, where we planned classes for books, users, and purchases. The second exercise reinforced basic OOP principles by creating a Polygons class with methods to store and modify shape details. These activities showed how OOP makes code organized, reusable, and easy to expand.

The results proved that good design and clear methods are essential in software development. By visualizing class structures and testing code, I saw how OOP helps solve problems in a structured way. Whether building an e-book system or working with geometric shapes, OOP principles like encapsulation and inheritance make programs more efficient. This lab was a great way to practice turning real-world objects into Python code, preparing us for more complex programming tasks in the future.

References

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

[2] GeeksforGeeks, “DSA Tutorial Learn Data Structures and Algorithms,” *GeeksforGeeks*, Jul. 25, 2025. https://www.geeksforgeeks.org/dsa/dsa-tutorial-learn-data-structures-and-algorithms/

[3] “W3Schools.com.” https://www.w3schools.com/dsa/dsa_intro.php