UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 1

Object-oriented Programming

0

Submitted by:
Barbas, Steven Jade P.

Instructor:
Engr. Maria Rizette H. Sayo

July, 30, 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: Inheritance Diagram

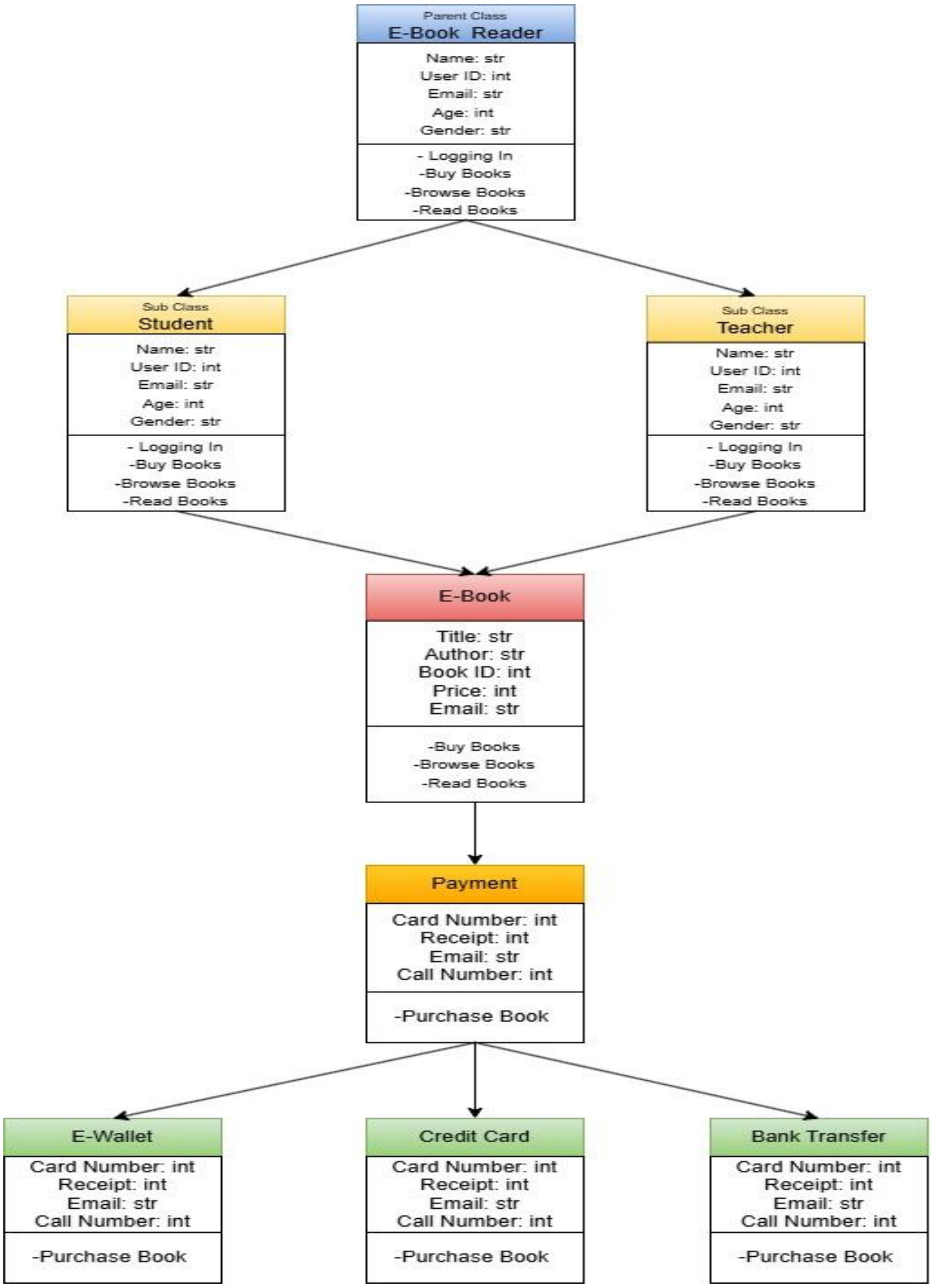
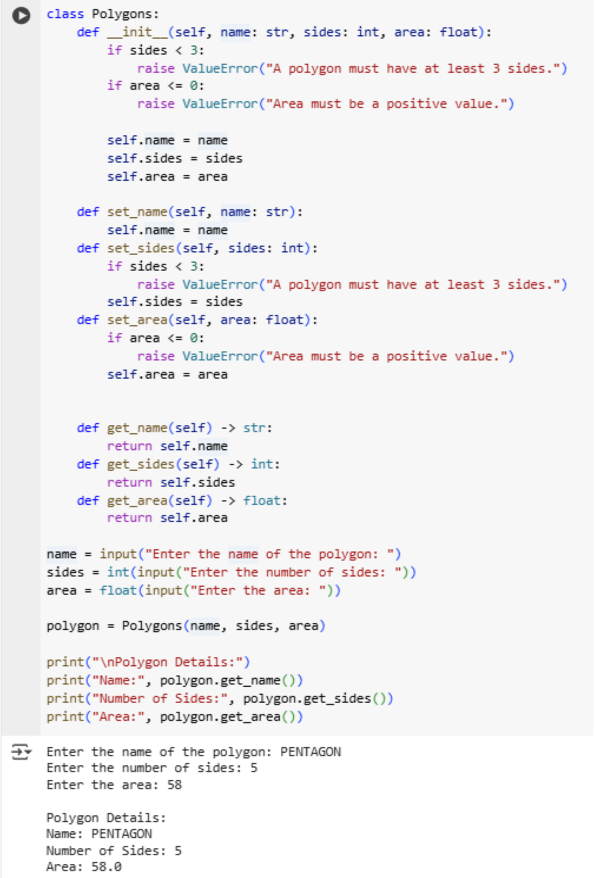


Figure 1 Inheritance Diagram of Task A

The main class is E-Book Reader, which has basic user details like name, ID, email, age, and gender. It also includes functions like logging in, buying books, browsing books, and reading books. Two types of users Student and Teacher inherit these same features. The E-Book class stores book details such as title, author, ID, and price. Users can buy, browse, or read these books. To make a purchase, users can pay through different methods: E-

Wallet, Credit Card, or Bank Transfer. All payment options require details like card number, receipt, email, and a call number. Once a book is bought, the user can read it.

B: Polygon Class



```python
class Polygons:
    def __init__(self, name: str, sides: int, area: float):
        if sides < 3:
            raise ValueError("A polygon must have at least 3 sides.")
        if area <= 0:
            raise ValueError("Area must be a positive value.")

        self.name = name
        self.sides = sides
        self.area = area

    def set_name(self, name: str):
        self.name = name
    def set_sides(self, sides: int):
        if sides < 3:
            raise ValueError("A polygon must have at least 3 sides.")
        self.sides = sides
    def set_area(self, area: float):
        if area <= 0:
            raise ValueError("Area must be a positive value.")
        self.area = area


    def get_name(self) -> str:
        return self.name
    def get_sides(self) -> int:
        return self.sides
    def get_area(self) -> float:
        return self.area

name = input("Enter the name of the polygon: ")
sides = int(input("Enter the number of sides: "))
area = float(input("Enter the area: "))

polygon = Polygons(name, sides, area)

print("\nPolygon Details:")
print("Name:", polygon.get_name())
print("Number of Sides:", polygon.get_sides())
print("Area:", polygon.get_area())
```

```
Enter the name of the polygon: PENTAGON
Enter the number of sides: 5
Enter the area: 58

Polygon Details:
Name: PENTAGON
Number of Sides: 5
Area: 58.0
```

Link for B: Polygon Class: Untitled0.ipynb - Colab

This Python program defines a Polygons class that represents a polygon with attributes: name, number of sides, and area. It includes input validation to ensure the polygon has at least 3 sides and a positive area. The program allows users to input a polygon’s details, creates a Polygons object, and then displays its properties using getter methods.

IV. Conclusion

In this laboratory activity I worked on two task, the first task involved creating an inheritance diagram for an E-Book Reader system, which illustrated how different user types (Student and Teacher) can inherit common features from a main class. It also showed how supporting classes like E-Book and various payment methods interact within the system. The second task was a Python program that defined a Polygons class with input validation, encapsulation, and getter/setter methods. Through these activities, I learned how to design structured class relationships and implement functional programs with proper data handling and object-oriented principles.