UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
2nd Semester, School Year 2024-2025

Object-Oriented Programming

Laboratory Activity No. 1

Review of Technologies

Submitted by:
Amutan, Rogenvan B.
Sat. 12:00PM to 8:30PM / BSCPE 1A

Submitted to
Engr. Maria Rizette H. Sayo
Instructor

Date Performed:
18-01-2025

Date Submitted
18-01-2025

I. Objectives

In this section, the goals in this laboratory are:

- To define the key terms in Object-oriented programming
- To be able to know the construction of OO concepts in relation to other types of programming such as procedural or functional programming

II. Methods

General Instruction:

A. Define and discuss the following Object-oriented programming concepts:

1. Classes

   In object-oriented programming, a class is a blueprint for creating objects (a particular data structure), providing initial values for state (member variables or attributes), and implementations of behavior (member functions or methods).

2. Objects

   In object-oriented programming (OOP), objects are the basic entities that exists in the memory. Each object is based on a blueprint of attributes and behaviors (variables and functions) defined as Class. A class is a blueprint that defines common attributes and behaviors, allowing for the creation of reusable objects. Objects are instances of a class, which can access the class's data members and functions. Memory is not allocated when a class is defined, but it is allocated when an object is created from the class.

3. Fields

   In object-oriented programming (OOP), a field—also known as an attribute or property—is a variable declared within a class that defines the characteristics or state of objects created from that class. Fields represent the data that an object holds. For example, in a Car class, fields might include color, make, and model, each storing specific information about a car object.

4. Methods

   Methods represent behaviors. Methods perform actions; methods might return information about an object or update an object’s data. The method’s code is defined in the class definition.

5. Properties

   Properties in OOP are also able to be looked at as functions, a property houses a function that can have its procedures or variables altered without directly going to the code and editing it. A property can be changed or updated based on user input which allows for a lot of user-interactive programs and applications.

III. Results

In object-oriented programming (OOP), a class is a blueprint that defines the structure and behavior of objects, like a template for creating cars. An object is a specific instance of a class, such as a red Tesla. Fields are variables within a class that store data about the object, like its color or speed. Methods are actions or behaviors that the object can perform, such as driving or braking. Properties are special methods used to control access to fields, ensuring the data is handled properly, like checking a car's speed before updating it.

These concepts—class, object, fields, methods, and properties—are essential to structuring code in a way that models real-world scenarios, supporting modularity, reusability, and easier maintenance in software design. These principles are further explored in works such as Design Patterns: Elements of Reusable Object-Oriented Software by Gamma, Helm, Johnson, and Vlissides (1994)

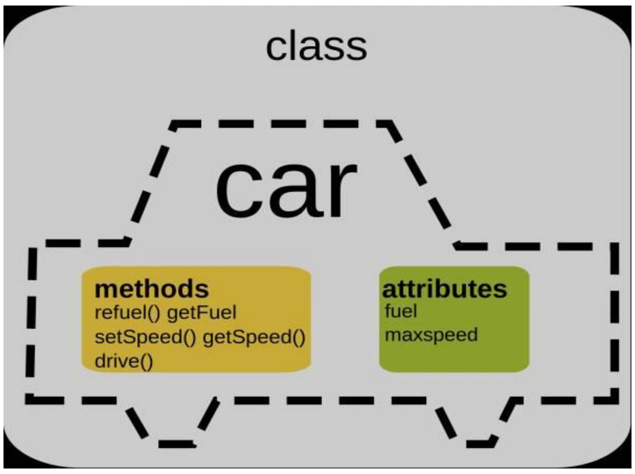

Figure 1. Understanding Object-Oriented Programming Through a Car Example

https://www.unionsquaredesign.com

Here's an image that illustrates the core concepts of object-oriented programming (OOP) using a car as an example. One commonly cited explanation of object-oriented programming (OOP) concepts comes from the book "Object-Oriented Analysis and Design with Applications" by Grady Booch. Booch explains, "The essence of OOP lies in the identification and organization of software elements as objects, each encapsulating data (fields) and behavior (methods), and interacting through well-defined interfaces."

IV. Conclusion

In conclusion, object-oriented programming (OOP) offers a strong framework for organizing and structuring software by emphasizing the creation of classes and objects that represent real-world entities. By defining fields, methods, and properties within classes, developers can build modular, reusable, and maintainable systems. These concepts are fundamental for managing complexity in software design, allowing for better interaction between components and simplifying code management. Studies and authoritative texts have consistently underscored the importance of OOP in creating efficient and scalable applications, making it a cornerstone of modern software development.

Reference

Book

[1] B. Booch, Object-Oriented Analysis and Design with Applications, 2nd ed. Reading, MA: Addison-Wesley, 1994. [Online]. Available: https://archive.org/details/objectorientedan00booc/page/n3/mode/2up. [Accessed: Jan. 18, 2025].

Website

[2] GeeksforGeeks, "What are objects in programming?" [Online]. Available: https://www.geeksforgeeks.org/what-are-objects-in-programming/. [Accessed: Jan. 18, 2025].

[3] Brilliant.org, "Classes in OOP," [Online]. Available: https://brilliant.org/wiki/classes-oop/. [Accessed: Jan. 18, 2025].

[4] GeeksforGeeks, "Object-Oriented Programming (OOP) Concepts for Designing Systems," [Online]. Available: https://www.geeksforgeeks.org/object-oriented-programingoop-concepts-for-designing-sytems/?utm_source=chatgpt.com&fbclid=IwY2xjawH4X_BleHRuA2FlbQIxMAABHfyeS_9-yLHlMrDgmRhifd8mADSdM1TMiwbkFySD3BecxEsuDe8nf7CHlg_aem_gkUcGRQW63GjJn6se-L72Q. [Accessed: Jan. 18, 2025].

[5] Educative.io, "Object-Oriented Programming," [Online]. Available: https://www.educative.io/blog/object-oriented-programming. [Accessed: Jan. 18, 2025].

[6] Wikibooks, "Object Oriented Programming/Properties," [Online]. Available: https://en.wikibooks.org/wiki/Object_Oriented_Programming/Properties. [Accessed: Jan. 18, 2025].