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:
Ampong, J-kevin L.
SAT-4:30-8:30 / CpE-1A

Submitted to
Engr. Maria Rizette H. Sayo
Professor / Instructor

Date Performed:
18-01-2025

Date Submitted
DD-MM-YYYY

I. Objectives

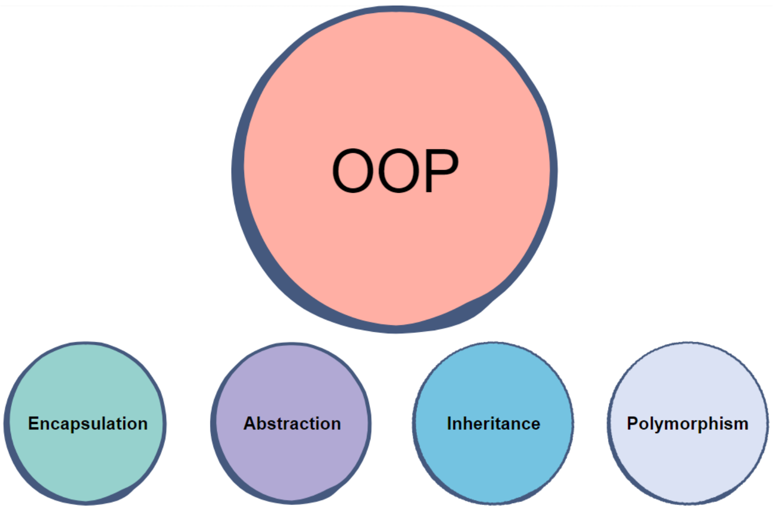
In this section, the goals in this laboratory are:

- 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

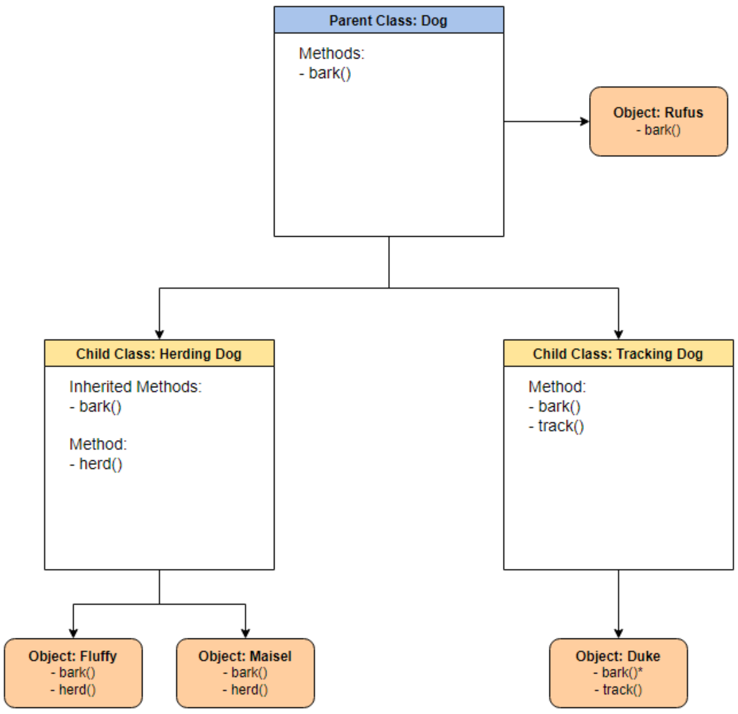
Object-oriented programming

Object-oriented programming (OOP) is one of core programming paradigms that are widely used in software development. It is a standard approach for coding and often taught as an essential skill throughout a programmer.



Classes

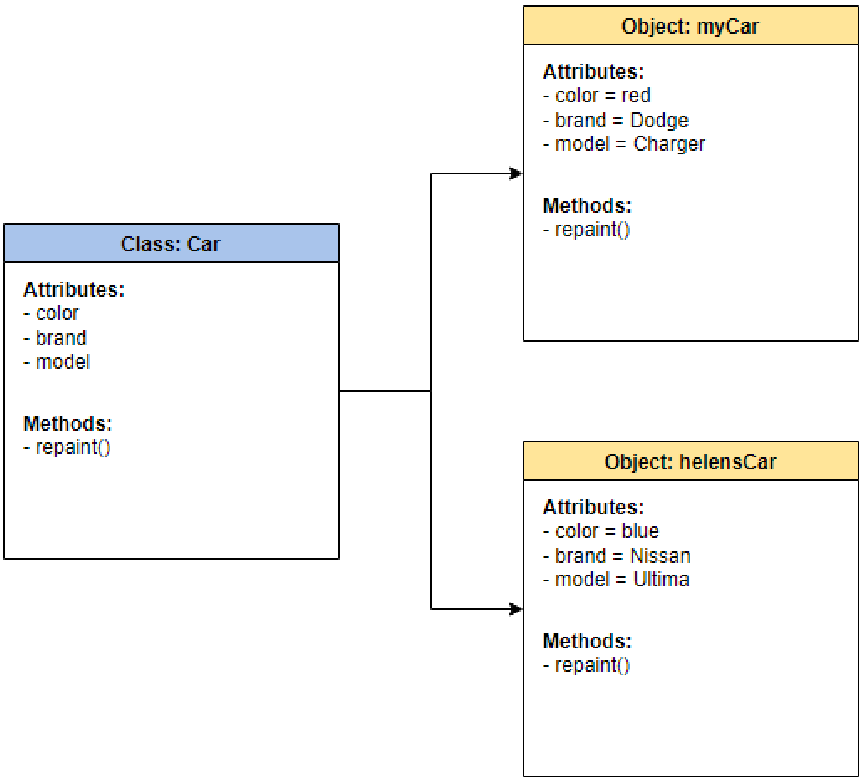
Classes can be thought of as user-defined templates or blueprints that define the structure and behavior of objects. They include methods (functions) and attributes (data) that allow it to be modified.



OOP for software engineering

Objects

Objects represent actual instances created based on the structure of a class. Each individual object contains its own set of specific data and operates according to the class's blueprint.

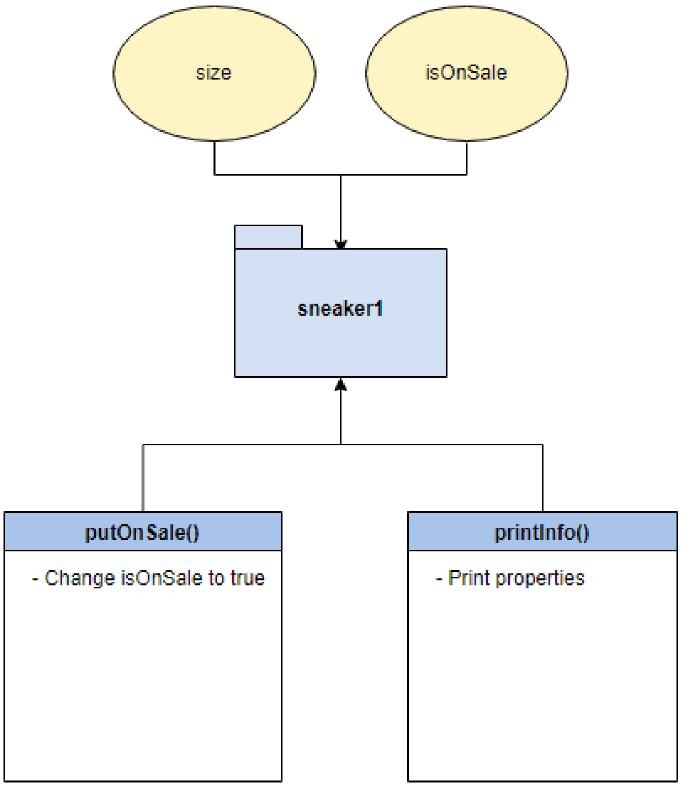


Fields

Fields refer to instance variables, which store data unique to each object. These are attributes that hold values specific to an individual instance of a class.

Methods

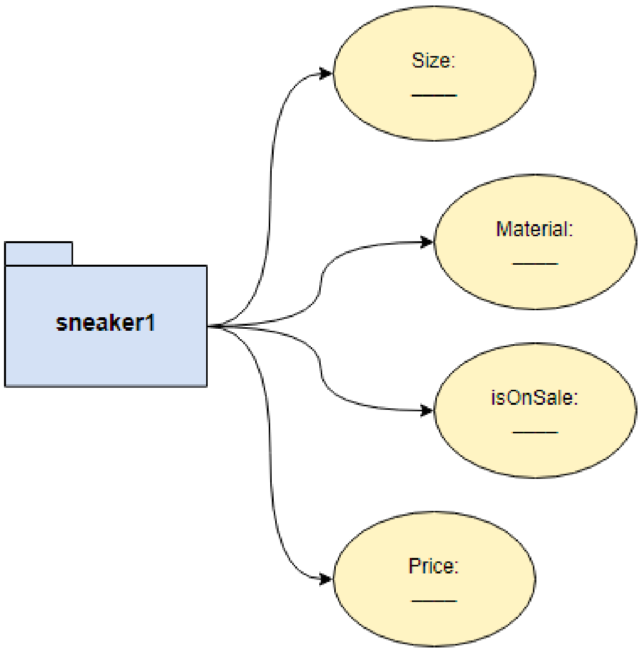
Methods determine an object's behavior. They are functions associated with an object that can modify its properties or perform specific actions of each individual object.



sneaker1 object that contains properties (top) and methods (bottom)

Properties

Properties describe the characteristics of an object. These are data elements unique to each object meaning each data is separated for one to another and can be accessed or modified by methods in the program, essentially acting as descriptors for the object.



Property fields within sneaker1 object

III. Results

Benefits of Object-Oriented Programming (OOP):

1. Easier Maintenance: OOP organizes code into modular parts, making it simpler to update or manage one section without affecting the rest.

2. Scalability: OOP helps software grow smoothly by adapting to new requirements.

3. Code Reusability: Through inheritance, existing code can be reused in other programs, saving time and resources.

4. Reliability: Features like encapsulation and abstraction protect code, reducing bugs and unintended interference.

Drawbacks of Object-Oriented Programming (OOP):

1. Steep Learning Curve: Beginners might find OOP concepts challenging at first.

2. Performance Issues: OOP can be slower than procedural programming due to added abstraction and bulkier objects.

3. Complex Design: Interconnected objects can make debugging and problem isolation harder.

4. Higher Memory Usage: Objects may consume more memory compared to simpler programming approaches.

IV. Conclusion

The lab exercise covered Object-Oriented Programming (OOP), one of the core concepts in contemporary software development, in detail. The fundamental elements of OOP, such as classes, objects, fields, methods, and properties, were thoroughly understood. Every concept was thoroughly examined, emphasizing its function in producing organized, effective, and reusable code.

The advantages of OOP were highlighted, demonstrating how its modularity makes maintenance easier by separating modifications to individual code segments. In addition to improving reliability, its modular design facilitates software scaling by adapting to changing needs. Additionally, the idea of code reusability through inheritance showed how developers might build on previous work to save time and money. Program robustness is further enhanced by features like abstraction and encapsulation, which lessen errors and stop accidental interference.

But OOP's drawbacks were also recognized. Although its design patterns and abstraction are beneficial, they can also present a challenging learning curve for novices, which could be frightening for individuals who are not familiar with programming. Additionally, when compared to more straightforward paradigms like procedural programming, OOP may result in slower performance and higher memory utilization. It was also mentioned that debugging and isolating problems in large systems may be difficult due to the intricacy of interrelated elements.

All things considered, this lab exercise provided a useful chance to examine and comprehend OOP's fundamentals and software engineering applications. By balancing the pros and cons, the exercise offered a fair assessment of how OOP can be used successfully in practical situations. The knowledge acquired helps students get ready for the practical application of OOP in future projects and highlights its importance as a potent technique for creating scalable, maintainable, and dependable software solutions.

Reference

Website

Doherty, E. (n.d.). What is object-oriented programming (OOP)? Explained in depth. Educative. Retrieved 18 January 2025, from https://www.educative.io/blog/object-oriented-programming

Thelin, R. (n.d.). How to use Object-Oriented programming in Python. Educative. Retrieved 18 January 2025, from https://www.educative.io/blog/how-to-use-oop-in-python

OOP Terminology: class, attribute, property, field, data member. (n.d.). Stack Overflow. Retrieved 18 January 2025, from https://stackoverflow.com/questions/16751269/oop-terminology-class-attribute-property-field-data-member

S. Chatterjee, “What is Object-Oriented Programming? Top 4 Benefits and Limitations,” Emeritus India, Retrieved 18 January 2025, from https://emeritus.org/in/learn/what-is-object-oriented-programming/