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:
Asugas, Kenneth R.
S 04:30PM-08:30PM / 1A

Submitted to
<Facilitator Name>
<Position>

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 - are essentially user-defined data types that serve as blueprints for creating objects. A class defines the structure of methods and attributes, allowing for the specification of how objects will behave and store data. From a class, individual objects are instantiated, each adhering to the class's structure, but capable of holding unique values for its attributes.
2. Objects - are instances of classes that represent real-world entities. They encapsulate both state, which is the data or attributes specific to each instance, and behavior, which is the functionality provided by methods defined within the class. Each object can operate independently and hold different values for its attributes, even though they share the same structure as other objects instantiated from the same class.
3. Fields (or attributes) - are variables declared within a class that store the data associated with objects of that class. These fields define the state of an object, and each instance of the class has its own copy of these variables, which can vary in value depending on the object.
4. Methods are the behaviors or actions associated with a class and its objects. They are functions defined within the class that can modify the object’s state, perform computations, or provide information about the object. Methods typically access and manipulate the object's attributes, and they may return values or trigger changes in the object's state.

5. Properties provide controlled access to an object's fields, often using getter and setter methods. They are designed to allow for encapsulation by controlling how fields are accessed or modified. With properties, developers can define specific rules or validation for getting or setting a value, ensuring that the internal state of an object remains consistent and secure.

III. Results

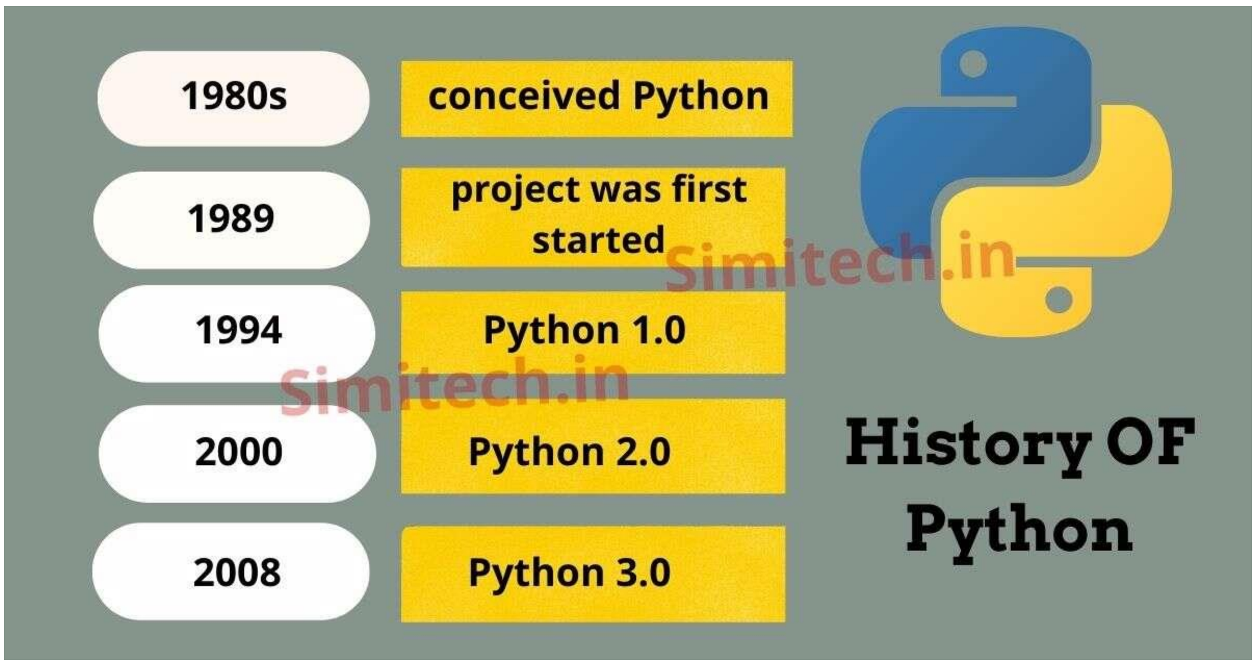

Figure 1. Python History and Version

Source: Simitech, "Python History," [Online]. Available: https://simitech.in/python-history/. [Accessed: Jan. 18, 2025].

This image illustrates a timeline highlighting key milestones in the history of Python, a popular programming language. Python, conceived by Guido van Rossum in the late 1980s, was designed to be a simple yet powerful programming language inspired by ABC. The project began in 1989, and the first official version, Python 1.0, was released in 1994, introducing core features like functions and modules. Python 2.0 followed in 2000, adding innovations like list comprehensions but faced legacy issues. In 2008, Python 3.0 was launched as a backward-incompatible overhaul to fix design flaws, marking a significant transition. Today, Python remains a leading language, widely used in web development, data science, and AI, known for its simplicity and versatility.

IV. Conclusion

The laboratory activity on Object-Oriented Programming provided a foundational understanding of key OOP concepts such as classes, objects, fields, methods, and properties. Through detailed discussions and definitions, the activity emphasized the significance of encapsulation, abstraction, and modularity in software design. Additionally, it highlighted the role of Python in demonstrating OOP principles and its evolution as a versatile and widely-used programming language. The integration of theoretical knowledge with practical examples facilitated a deeper appreciation of how OOP simplifies the development and maintenance of complex software systems.

Reference

Book
[1]

L. J. P. van der Meer, *Object-Oriented Programming with Java*, 3rd ed. New York, NY, USA: Pearson, 2021.

A. B. J. Smith, *Introduction to Object-Oriented Programming*, 2nd ed. Boston, MA, USA: McGraw-Hill Education, 2019.

G. P. O'Rourke, *Programming in Java: An Introduction to Object-Oriented Concepts*, 4th ed. San Francisco, CA, USA: Wiley, 2020.

Website
[2]

“Object-Oriented Programming Basics,” Programiz. [Online]. Available: **https://www.programiz.com/python-programming/object-oriented-programming**. [Accessed: Jan. 18, 2025].

“Classes and Objects in Python,” W3Schools. [Online]. Available: **https://www.w3schools.com/python/python_classes.asp**. [Accessed: Jan. 18, 2025].

“What is a Field in OOP?” Studytonight. [Online]. Available: **https://www.studytonight.com/post/field-in-object-oriented-programming**. [Accessed: Jan. 18, 2025].

“Methods in Object Oriented Programming,” GeeksforGeeks. [Online]. Available: **https://www.geeksforgeeks.org/methods-in-python/**. [Accessed: Jan. 18, 2025].

“Getters and Setters in Python,” Real Python. [Online]. Available: **https://realpython.com/python-getter-setter/**. [Accessed: Jan. 18, 2025].

“Object-Oriented Programming Concepts Explained Simply,” Educative. [Online]. Available: **https://www.educative.io/blog/object-oriented-programming**. [Accessed: Jan. 18, 2025].

Python Software Foundation, "Python Essays," [Online]. Available: https://www.python.org/doc/essays/. [Accessed: Jan. 18, 2025].

Python Software Foundation, "Python 3.0 Release," [Online]. Available: https://www.python.org/download/releases/3.0/. [Accessed: Jan. 18, 2025].