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:

Bron, Jhustine A.

Saturday (4:30-8:30) / BSCpE 1-A

Submitted to

Engr. Maria Rizette Sayo

Professor

Date Performed:

18-01-2025

Date Submitted

18-01-2025

I. Objectives

Object-oriented programming (OOP) is a programming paradigm that is used by every developer and utilizes it at some point in their programming career. It is the most popular programming paradigm used for software development. It relies on the concept of classes and objects and is used to for the structure of software programs into simple pieces of blueprints called **classes**, which are used to create individual instances of objects. Object-oriented programming languages include JavaScript, C++, Python etc. OOP in Python involves creating classes as blueprints for objects. These objects contain data, and the methods needed to manipulate that data. (educative.io, 2024)

Object-Oriented Programming organizes software design around data or objects instead of functions or logic unlike other programming paradigms**. Procedural Programming** on the other hand is a language derived from structure programming and is based on procedures or the functions that specify the computational steps that need to be performed. **Functional Programing** uses JavaScript to have Language independence, and its roots come from mathematics. All programming paradigms aim for reusable and focus on step-by-step instructions to solve problems. They build upon other programming paradigms by grouping related functions and data into objects. (Atatus, 2022)

II. Methods

In Object-oriented programming specifically in Python, you define a **class** by using the *class* keyword followed by a name and a colon to declare which attributes each instance of the class should have. Classes allow us to create a user defined data structure, it acts as a blueprint for how to define something, in a simpler way, class is like a form or a questionnaire. (RealPython, 2024)

Ex. Classes are usually created like this;

```
class MyClass:
  x = 5
```

Objects in OOP (specifically in Python) are the instance of a Class. It is the collection of data and behaviors that are defined in a class. It contains real data and is like a lot of people, it has data on how to do things and has memory inside them so they can remember it. The connection between class and object is that Class is the blueprint for that object, in other words, object is the house that is built through the blueprint. (GeeksforGeeks, 2023)

Ex. The syntax of a Python Object typically looks like this;

```
obj = MyClass()
print(obj.x)
```

Field in Python represents the data associated with objects and plays a role in defining the structure and behavior of classes. It refers to a variable that is associated with an object which are the instances of the class or the class itself. They are used to store data or attributes related to the state of an object.

Methods in OOP are the representation of behaviors, it performs actions; methods also return information about an object or update an object's data. It identifies the behaviors and actions that an object created from the class. It is essential for encapsulating functionality and promoting code reusability.

Properties in Python, more specifically the Property function return the object of the property class, and it is used to create the property of a class. It is a built-in function that allows us to create a special type of attribute called property. It encapsulates access to an object attribute and adds some logic to the proofs such as computation.

Ex. Properties in Python are input like this.

```
class Circle:
    def __init__(self, radius):
     self._radius = radius

  @property
   def radius(self):
     return self._radius
```

III. Results

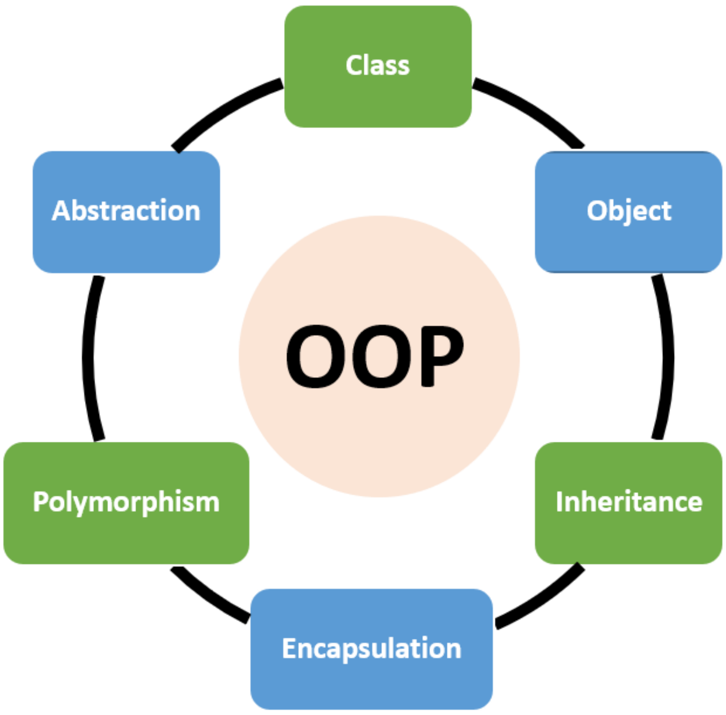

Figure 1. Pari, "Object oriented programming (OOP) in Python – Data stories," Nov. 20, 2020. https://nonlineardata.com/object-oriented-programming-oop-in-python

Throughout this extensive research about OOP (Object-Oriented Programming), numerous advantages were discovered. It was more reusable and a free area when it comes to sharing codes, it can use the class and subclasses multiple times without rewriting everything from scratch. It is easy to troubleshoot because of the separate various functionalities in the classes.

You can easily reuse functionalities of multiple classes without recreating the wheel and when you apply a modification in the class, the change is propagating in all of the dependent classes & objects. From Classes, Objects, to Methods, OOP specifically in Python is a useful programming tool.

For more information about the advantages and definitions of OOP in python, you can click the link below the Figure.

IV. Conclusion

Object-Oriented Programming (OOP) is a popular programming paradigm that can easily organize software designs around objects rather than logic or functions unlike but closely related to other programming paradigms such as functional and procedural programming. In python, a class is a structure that serves as a blueprint for creating objects. (educative.io, 2024; Atatus, 2022)

Objects are instances of classes that contain data and behaviors. Fields represent the variables associated with an object or class; they also store data that defines the object’s state. Methods are functions defined within a class that also represent the behaviors or actions of objects. Lastly, Properties in Python enable controlled access to an object’s attributes that incorporate computation when accessing data. (RealPython 2024; GeeksforGeeks, 2023)

There is a lot to say about OOP like advantages, their reusability and easy troubleshooting are one of the main advantages of it. You can reuse classes and subclasses without rewriting the code. OOP in Python is a powerful tool for creating organized, efficient, and reusable software solutions.

Reference

Websites:

Aarthi, “Programming paradigms compared: functional, procedural, and Object-Oriented,” *Atatus Blog - for DevOps Engineers, Web App Developers and Server Admins.*, Oct. 14, 2022. https://www.atatus.com/blog/programming-paradigms-compared-function-procedural-and-oop/

D. Amos, “Object-Oriented Programming (OOP) in Python,” Dec. 15, 2024. https://realpython.com/python3-object-oriented-programming/

D. E. C, “Fundamentals of object-oriented programming (OOP) - Dev Genius,” *Medium*, Dec. 13, 2022. [Online]. Available: https://blog.devgenius.io/fundamentals-of-object-oriented-programming-oop-3b68dcc2d179

E. Doherty, “What is object-oriented programming (OOP)? Explained in depth,” *Educative*. https://www.educative.io/blog/object-oriented-programming

GeeksforGeeks, “Python object,” *GeeksforGeeks*, Jun. 21, 2023. https://www.geeksforgeeks.org/python-object/

GeeksforGeeks, “Python property() function,” *GeeksforGeeks*, Jul. 03, 2024. https://www.geeksforgeeks.org/python-property-function/

Pari, “Object oriented programming (OOP) in Python – Data stories,” Nov. 20, 2020. https://nonlineardata.com/object-oriented-programming-oop-in-python

T. C. Monteiro, “Object-Oriented Programming in Python – explained in Plain English,” *freeCodeCamp.org*, Sep. 07, 2023. https://www.freecodecamp.org/news/object-oriented-programming-python/