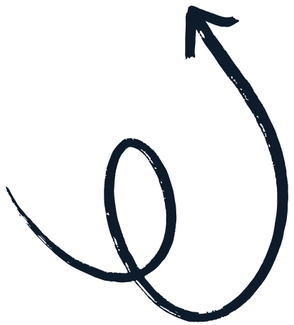
Object-Oriented Programming in Python

Presented by Mansi Bhagwat

What is OOP?

Understanding Object-Oriented Programming Concepts

Object-Oriented Programming (OOP) is a programming paradigm centered on objects, which fosters **modularity**, **reusability**, and **organization** in coding. Python inherently supports OOP, enhancing productivity and maintainability.



Classes and Objects

Building Blocks of OOP

Class Definition

A **class** serves as a blueprint for creating objects. It encapsulates data and behavior, allowing for a structured approach to building software applications.

Object Instance

An **object** is an instance of a class, representing a specific entity with distinct attributes and behaviors, similar to how a house is built from a blueprint.

Analogy Explained

Think of a class as a **blueprint**, while an object is the **house** constructed using that blueprint. This analogy highlights the relationship between classes and objects in OOP.

Attributes and Methods

Understanding Object Properties and Behaviors

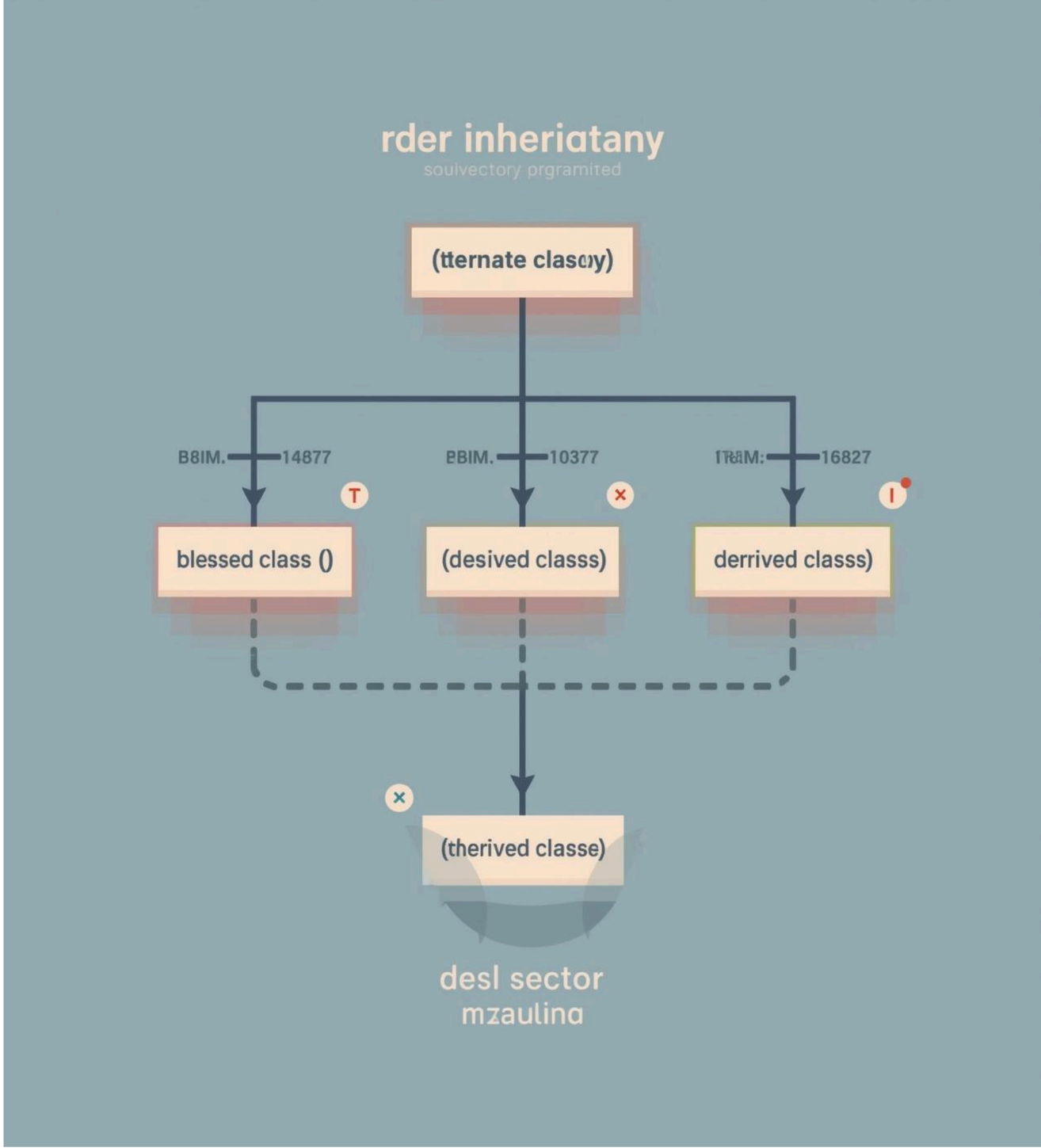
In Python, **attributes** are variables that store data for an object, while **methods** are functions defined within a class that outline its behaviors. For example, a 'Car' class may have attributes like color and methods such as drive().

Encapsulation in Object-Oriented Programming

Encapsulation is a fundamental concept in OOP that restricts access to certain components, using private variables to maintain data integrity and control. This ensures robust and secure programming practices.

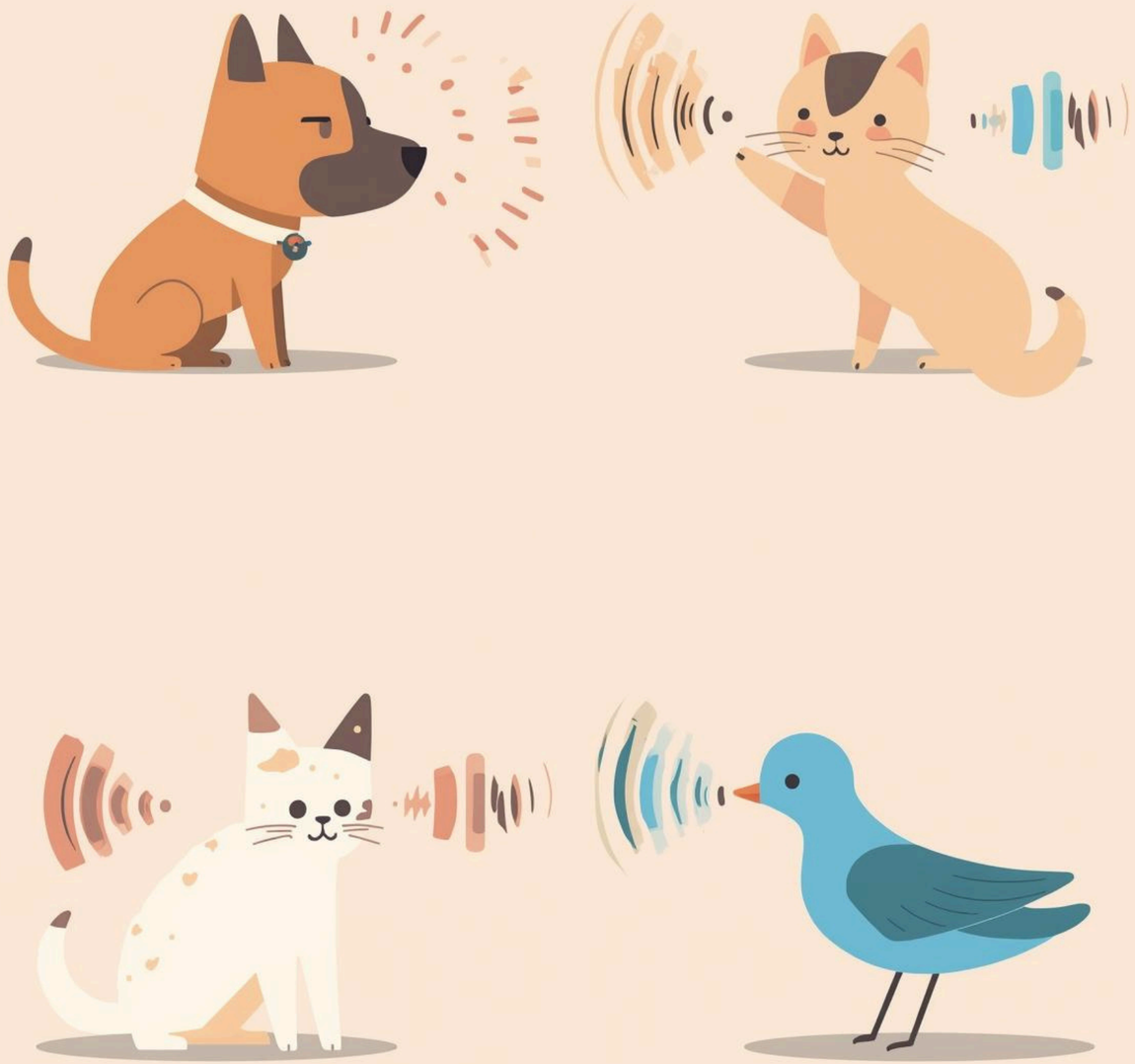
Inheritance in Object-Oriented Programming

Inheritance allows the creation of a new class based on an existing one, promoting code reuse and hierarchical relationships. Base classes serve as foundations, while derived classes extend functionality.



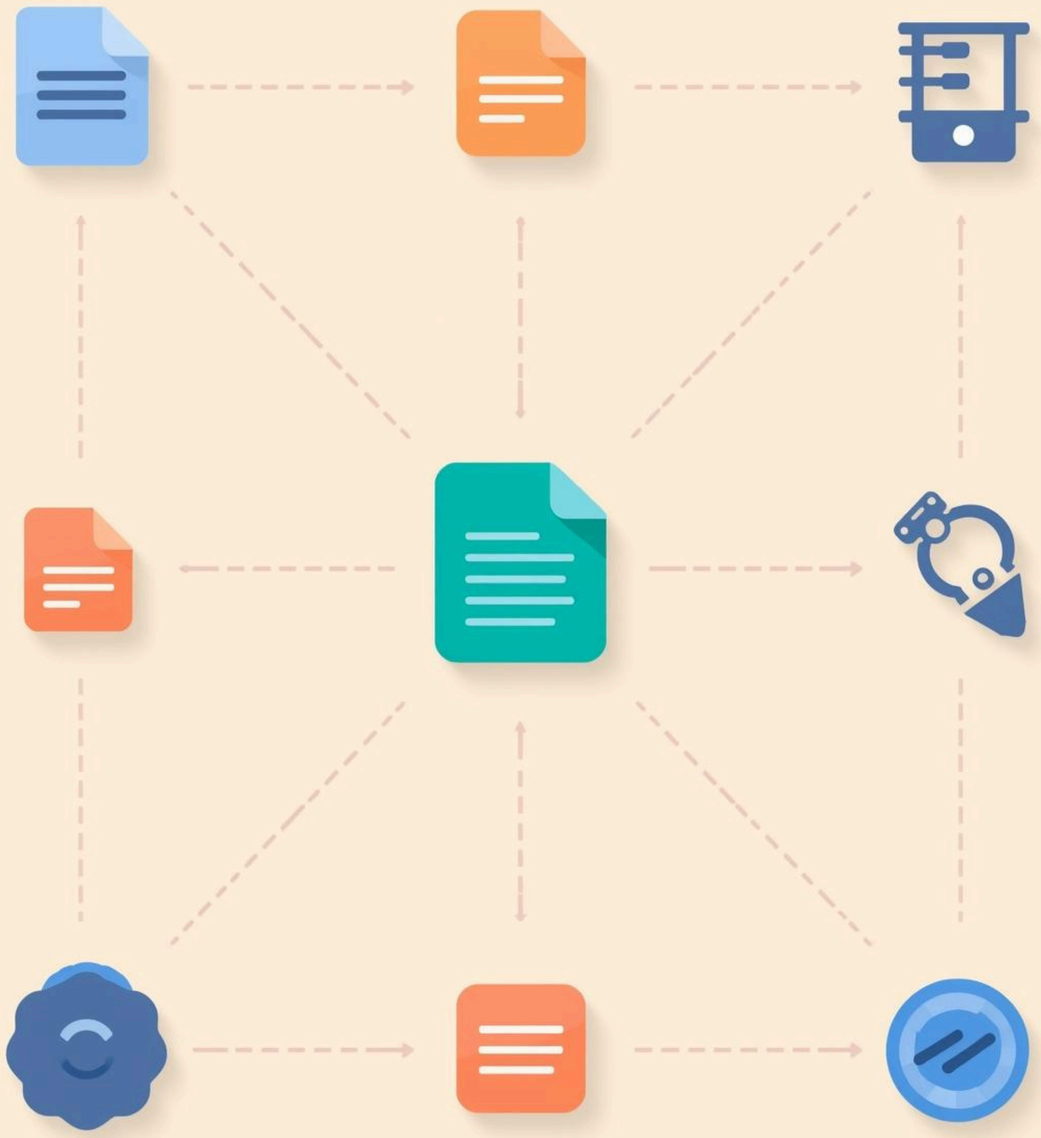
Understanding Polymorphism in OOP

Polymorphism allows methods with the same name to operate differently based on the object invoking them. This flexibility simplifies code management and enhances program adaptability, fostering cleaner and more maintainable code.



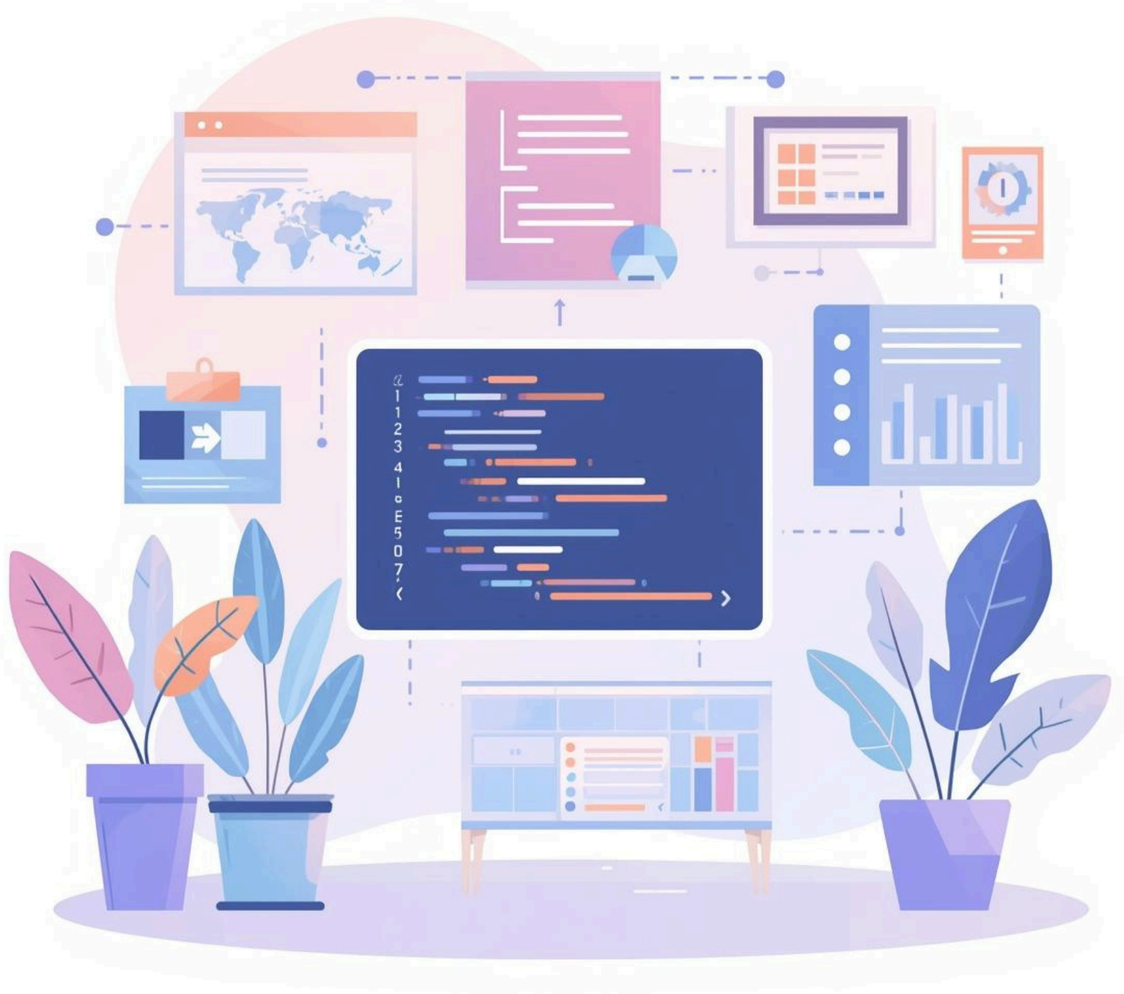
Summary of OOP Concepts

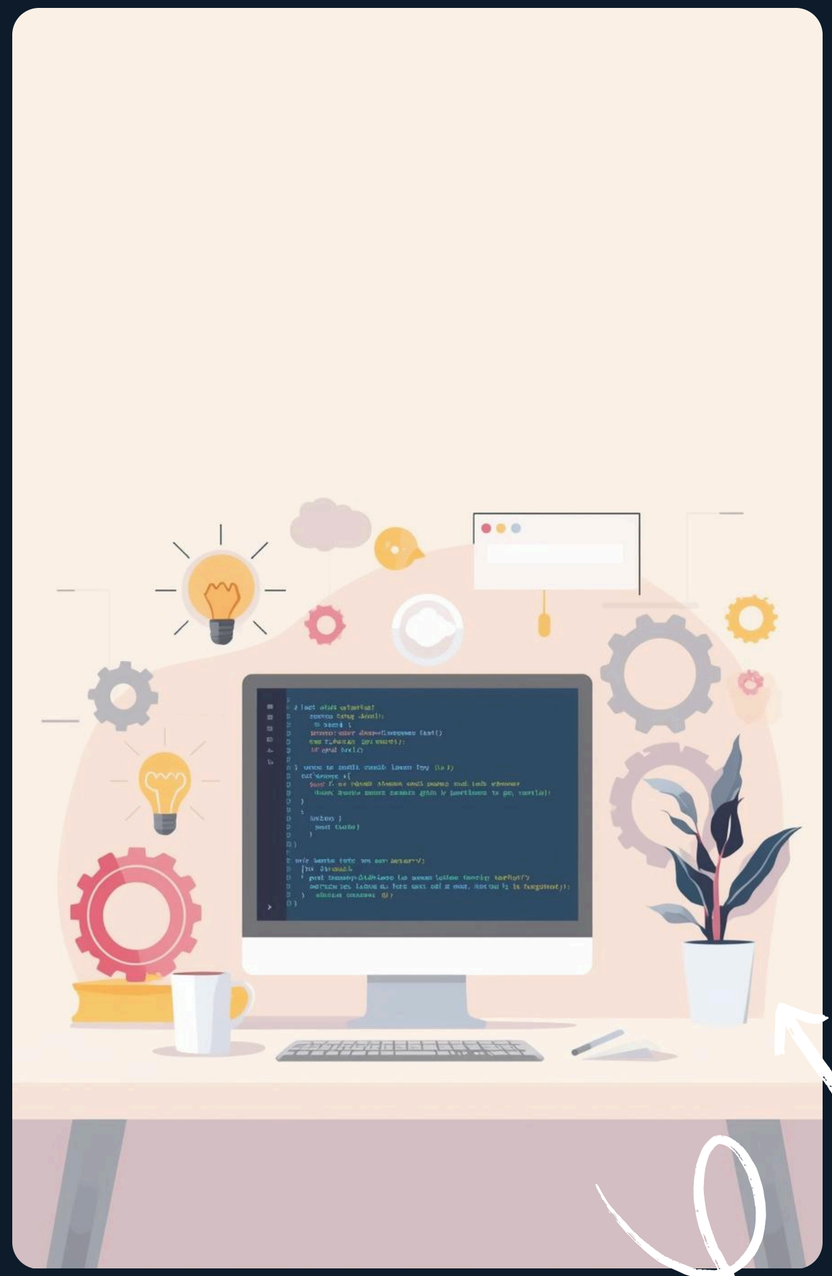
- Classes and Objects define the structure of data.
- Attributes and Methods encapsulate data and behavior together.
- Inheritance and Polymorphism enhance code reusability and flexibility.



Advantages of OOP in Python

Object-Oriented Programming enhances code organization and **modularity**, making maintenance easier and facilitating **real-world modeling**. These features streamline development and foster efficient collaboration in complex software projects.





Thank You