Object-Oriented Programming (OOP)

Why Do We Use OOP?

OOP is used to **organize code around real-world entities**. It helps in managing **complex programs** by grouping related data and functions together.

Example (Simplified)

Imagine a **person**. If we want to store their **salary**, we can structure it as:

- person.salary instead of using separate variables.

This makes the data **easy to access** and **logically connected** to the person.

Key Concepts in OOP

1. **Classes & Objects**
    - **Class** is a **blueprint** (e.g., "Car" defines properties like color, brand).
    - **Object** is an **actual instance** of that class (e.g., "MyCar" is red and a Toyota).

2. **Encapsulation**
    - Protects data by keeping it private within an object.
    - Example: A **bank account** where balance can only be accessed via proper methods.

3. **Inheritance**
    - A new class can **inherit** features from an existing class.

- Example: **ElectricCar** inherits general properties of **Car** but adds its own features.

4. Polymorphism

- One function can work in different ways for different objects.
- Example: **A person can speak, and a robot can also "speak", but differently.**

5. Abstraction

- Hides unnecessary details, showing only the important ones.
- Example: **When using a mobile phone, you dial numbers but don’t see the internal working of circuits.**

OOP makes programs **organized, reusable, and scalable**, which is why it is widely used in modern software development.