OOP Introduction

What is OOP?

- **Object-Oriented Programming (OOP)** is a style of programming focused on **objects**.
- Objects represent real-world entities (car, student, bank account).
- Every object has:
  - **Attributes (data/properties)** → car has color, model, speed.
  - **Methods (functions/actions)** → car can start(), stop(), drive().

Procedural vs OOP:

Procedural Programming:

- Focus on functions and step-by-step instructions.
- Data and functions are separate.

OOP Programming:

- Combines data and functions into a single unit (object).
- More organized and reusable.

Why OOP?

- Makes code **organized** and **easier to maintain**.
- Closer to real-life thinking.
- Reduces code repetition.
- Encourages reuse (same class can make many objects).

Core Concepts of OOP in Python:

- **Class** – Blueprint/template (design).
- **Object** – Real-world instance created from a class.
- **Attributes** – Variables inside a class.
- **Methods** – Functions inside a class.