📘 Lecture 15: Object-Oriented Programming (OOP)

🧠 What is OOP?

Object-Oriented Programming is a coding style that models real-life objects.

✅ Organizes code using:

- **Class** (blueprint)
- **Attributes** (characteristics)
- **Methods** (behaviors)
- **Objects** (real items)

🔷 Key OOP Components

1️⃣ Class

Blueprint/template used to create objects.

🛠️ Example: `Car`, `Student`, `House`

2️⃣ Attributes

Data/properties of the object

Example:

```
python
CopyEdit
color, speed, engine_capacity
```

3️⃣ Methods

Functions defined inside a class (object behaviors)

Example:

```
python
CopyEdit
start(), stop(), accelerate(), get_marks()
```

---

4 Object / Instance

Real entity created from a class

 Object has specific **attribute values**
 Shares same **methods** from class

Real-Life Example: Car

Class: `Car`

Attributes:

- color = "red"
- speed = 200
- fuel_capacity = 50

Methods:

- start(), stop(), accelerate()

Object:

```
python
CopyEdit
red_car = Car("red", 200, 50)
```

Example: Student Management

Class: `Student`

Attributes:

- name, age, marks, performance

◈ Methods:

- take_exam(), get_promoted(), attend_classes()

◈ Object:

```
python
CopyEdit
student1 = Student("Harris", 25, "Average")
```

🏭 How OOP Works

1. **Define Class** – Create blueprint
2. **Add Attributes** – Describe the object
3. **Add Methods** – Define object behaviors
4. **Create Object** – Instantiate with specific values

🏠 Example: Building a House

- **Class:** House
- **Attributes:** rooms, windows, layout
- **Methods:** open_window(), lock_door()

Each house built = different object of the same class

💡 Why Use OOP?

Benefit	Description
Reusability	Use same class to create many objects
Modularity	Break code into smaller manageable units
Maintainability	Easy updates via class
Scalability	Create 1000s of objects without writing new code

🔁 Practical Summary

Concept	Description
Class	Template for object creation
Attribute	Object's properties (data)
Method	Object's behaviors (functions)
Object	Instance of a class with data

✅ Final Thoughts

✅ OOP is key for **clean**, **scalable**, and **efficient** programs.

Start practicing by:

- Creating your own `Car`, `Student`, `Laptop` classes
- Adding methods & attributes
- Instantiating objects with different values
