🧱 Class and Object (OOPS Concepts)

📘 Theory

🧩 1. What is a Class?

A **class** in Python is a **blueprint or template** for creating objects.
It defines **attributes (variables)** and **methods (functions)** that describe the behavior and properties of the objects.

Syntax Example:

```
class ClassName:
    # attributes (variables)
    # methods (functions)
```

🧍 2. What is an Object?

An **object** is an **instance** of a class.
It represents a **real-world entity**, created using the class.

Example:

```
obj = ClassName()
```

⚙️ 3. Attributes and Methods

- **Attributes** → Variables inside a class that hold data.
- **Methods** → Functions inside a class that define behaviors.

Example:

```
class Car:
    def __init__(self, brand, color):
        self.brand = brand      # Attribute
        self.color = color      # Attribute

    def display_info(self):     # Method
        print(f"Brand: {self.brand}, Color: {self.color}")
```

```
# Creating an object (instance)
my_car = Car("Tesla", "Red")
my_car.display_info()
```

Output:

```
Brand: Tesla, Color: Red
```

🌍 4. Local vs Global Variables

Type	Definition	Scope
Local Variable	Declared inside a function or method.	Accessible only within that function/method.
Global Variable	Declared outside all functions.	Accessible throughout the program.

Example:

```
x = 100  # Global variable


class Example:
    def show_value(self):
        y = 50  # Local variable
        print("Local variable (inside method):", y)
        print("Global variable (accessible):", x)


obj = Example()
obj.show_value()
```

Output:

```
Local variable (inside method): 50
Global variable (accessible): 100
```

🧠 Summary

- A **class** defines structure; an **object** represents a real instance of that structure.
- **Attributes** store data; **methods** define behavior.
- **Local variables** exist only inside functions or methods.
- **Global variables** can be accessed anywhere in the code.