



What is Class and Object in OOP?

✔ Definition of Class:

A **class** is like a **blueprint or template** used to create objects. It defines **properties (variables)** and **methods (functions)** that the object will have.

🧠 Think of a class like a **car design** — it describes what a car should have (engine, wheels, etc.), but it's **not a real car**.

✔ Definition of Object:

An **object** is a **real-world instance** of a class. When you create an object using a class, it has its **own values** for the properties defined in the class.

🧠 Using the same analogy: A real **car you drive** is an **object** created from the **car design (class)**.

✔ JavaScript Example (Beginner Friendly):

```
// Define a class
class Person {
  constructor(name, age) {
    this.name = name; // property
    this.age = age;
  }

  // Method
  greet() {
    console.log(`Hello, my name is ${this.name} and I'm ${this.age} years old.`);
  }
}

// Create an object (instance) of the class
const person1 = new Person("Milan", 25);
person1.greet(); // Output: Hello, my name is Milan and I'm 25 years old.
```

✔ Explanation (Point-by-Point):

- class Person defines a **blueprint** with properties name, age, and a method greet().
- new Person("Milan", 25) creates a real **object** with its own values.
- person1.greet() calls the method using the object's data.

✔ Summary Table:

Term	Meaning
Class	Blueprint/template for creating objects
Object	Real-world instance created from a class