

# Day 5

Object-Oriented Programming (OOP) uses "objects" to design applications and computer programs. It allows developers to structure software in a way that is modular, reusable, and easier to maintain.

## 1. Objects

### Cars

Think of each car as an "object." In the real world, every car is made based on a design or model but can have different characteristics like colour, brand, and horsepower. In OOP, objects are instances of classes, where a class can be thought of as the blueprint or design for creating objects. So, a class defines how a car should be structured and what it should be able to do, while each car (object) made from that class can have its own specific features.

## 2. Classes

### Car Models/Blueprints

A class is like the blueprint for a car. It defines the basic structure and capabilities of the car, such as having wheels, seats, and the ability to drive. However, the class itself isn't a car; it's just the plan for making cars. You can use this blueprint to create many cars (objects), each with its own unique characteristics (like a BMW M3, Audi RS6, or Porsche 911).

## 3. Encapsulation

### Car's Internal Components

Encapsulation is like keeping the car's engine and internal mechanisms hidden under the hood. You don't need to know exactly how the engine works to drive the car; you just use the steering wheel, pedals, and gears. In OOP, encapsulation means keeping some parts of an object hidden from the outside, exposing only what is necessary for the object's use. This way, you can change the internal workings without affecting how someone drives the car.

## 4. Inheritance

### Car Model Series

Imagine a basic car model that comes in different versions, like a standard, sport, and luxury edition. Each version starts with the basic features of the car but adds its own special features. In OOP, this is called inheritance. A "child" class inherits features from a "parent" class but can also have additional unique features or behaviours. For example, the sport edition might inherit the basic car structure but add a more powerful engine and sporty interior.

## 5. Polymorphism

### The Ability to Perform the Same Action in Different Ways

Polymorphism is similar to how different cars can achieve the same goal—moving forward—using different mechanisms (electric, petrol, hybrid). In OOP, polymorphism allows methods to do different things based on the object it is acting upon, even though the method might be called the same way. For instance, a "start" method might start a petrol car by igniting fuel, while for an electric car, it might just turn on the power supply from the battery.

## Object Task

- Object mapping - Pick some three things and map to OOP concepts using the above as a guide.