# Encapsulation in OOP

Encapsulation is a fundamental concept in OOP that revolves around the idea of **bundling data and methods within a single unit, known as a class**. The primary goal is to restrict access to the internal details of an object and expose only what is necessary for the object's functionality. Imagine your phone as an encansulated object, its internals are not known to you, you just know what it does and how to use it. Think of encapsulation as a protective barrier around an object, shielding its internal workings from the outside world.

### Example

Let's consider the example of a `Car` class. In an encapsulated design, the class would encapsulate relevant data, such as the car's make, model, and year, along with methods to operate the car, such as `startEngine()` and `accelerate()`. External code should interact with the `Car` object through well-defined interfaces rather than directly manipulating its internal properties. Here's a simple implementation in JavaScript:

```
class Car {
  constructor(make, model, year) {
    // Encapsulated data
    this.make = make;
    this.model = model;
    this.year = year;
    this.isEngineRunning = false;
    this.currentSpeed = 0;
  }

  // Encapsulated methods
  startEngine() {
    this.isEngineRunning = true;
```

```
    console.log('Engine started.');
  }

  accelerate(speed) {
    if (this.isEngineRunning) {
      this.currentSpeed += speed;
      console.log(`Accelerated to ${this.currentSpeed} km/h.`);
    } else {
      console.log('Engine is not running.
      Please start the engine first.');
    }
  }
}
```

In this example, the `Car` class encapsulates data like `make`, `model`, `year`, `isEngineRunning`, and `currentSpeed`. External code interacts with the `Car` object using the `startEngine()` and `accelerate()` methods, adhering to the principle of encapsulation. This shields the internal details of the `Car` object, providing a clear and controlled interface for external use. Encapsulation promotes code organization, reduces complexity, and enhances code maintainability. As beginners, grasping the concept of encapsulation lays the groundwork for understanding how to structure code in an organized and secure manner.

---