The summary of objected oriented programming

Object-Oriented Programming (OOP) is a programming paradigm that uses "objects" to design and structure software. In simple terms:

* **Objects:** These are instances of classes, representing entities with properties (attributes) and behaviors (methods).
* **Classes:** Blueprints for creating objects, defining their attributes and methods.
* **Encapsulation:** Bundling data and methods within objects to hide the internal state and restrict access.
* **Inheritance:** Mechanism where one class inherits properties and behaviors from another.
* **Polymorphism:** Ability for different classes to be treated as instances of the same class through common interfaces.

OOP makes code more modular, reusable, and easier to manage. 🌐✨

Feel free to ask if you want to dive deeper into any of these concepts!

Object-Oriented Programming (OOP) is a method of structuring a software program by bundling related properties and behaviors into individual objects. Here's a simple breakdown:

* **Objects:** Individual instances of classes that contain data and methods.
* **Classes:** Templates for creating objects, defining their attributes (properties) and methods (functions).
* **Encapsulation:** Grouping data and methods within objects to protect the internal state.
* **Inheritance:** Allowing one class to inherit properties and methods from another class.
* **Polymorphism:** Enabling objects to be treated as instances of their parent class, making code more flexible.

This approach helps make programs more modular, easier to maintain, and reusable. If you're curious about how it applies in practice, just let me know!