# Four principles of OOP:

**Encapsulation:**

Def: Each object keeps its state private, other object can use methods to call.

E.g., A Cat class can have private variables mood, energy, hungry. It also has a private method meow().

Other classes cannot tell the cat what to do.

In the public method you can modify the inner state using private variables and methods.

**Abstraction:**

Def: natural extension of encapsulation

Help to maintaining large codebase. We only use what we need(methods) without knowing about all the works(code) inside.

E.g., A phone can be used by touching buttons, we do not need to know what happens inside the phone.

**Inheritance:**

Def:  reuse the common logic and extract the unique logic into a separate class

Child classes can use fields and methods from parent class and can have its own method.

E.g., A Teacher class can be derived from Person class and have all public fields and methods of Person

**Polymorphism:**

Def: a way to use a class exactly like its parent so there's no confusion with mixing types. But each child class keeps its own methods as they are

E.g., Inherited from parent class Shape, we can have a list of mixed Triangles, Circles, Squares. All can use the method calculateArea() or calculatePerimeter(), but the actual method is defined in each child class.

# OOP Explained

**What is OOP:**

Def: a programming paradigm that relies on the concept of **classes** and **objects.**

**Benefits of OOP:**

Simplify structures, reusable, easier to debug, secure information with encapsulation.

**Building blocks of OOP:**

Consist of classes, objects, attributes, methods

Class: contain attributes (data, variable) and methods (behaviors, changing the data)

Objects: instances of classes, have specific attributes and can use methods.