

# Software Architect

## OOP & S.O.L.I.D

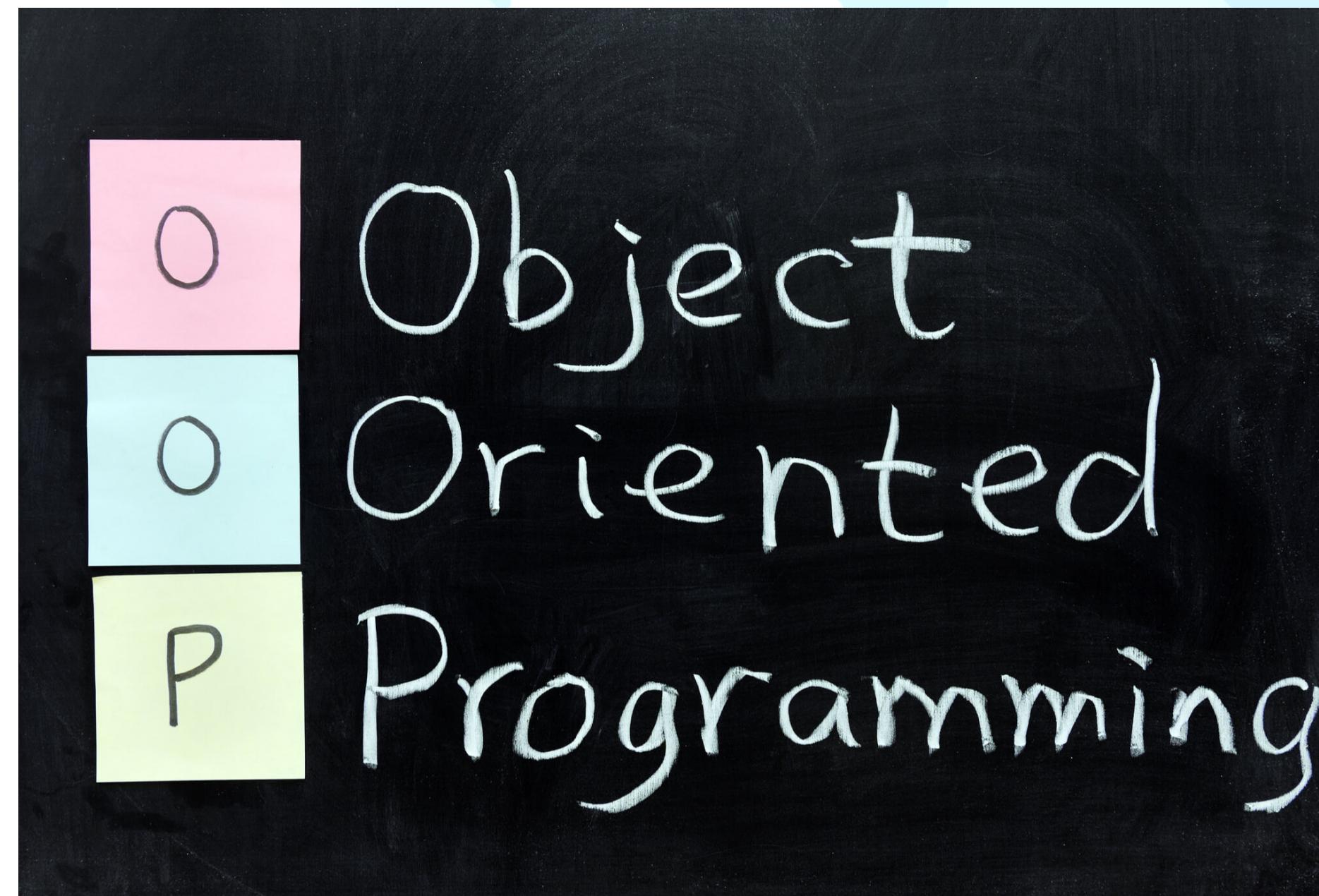


Image Source: Rober Half

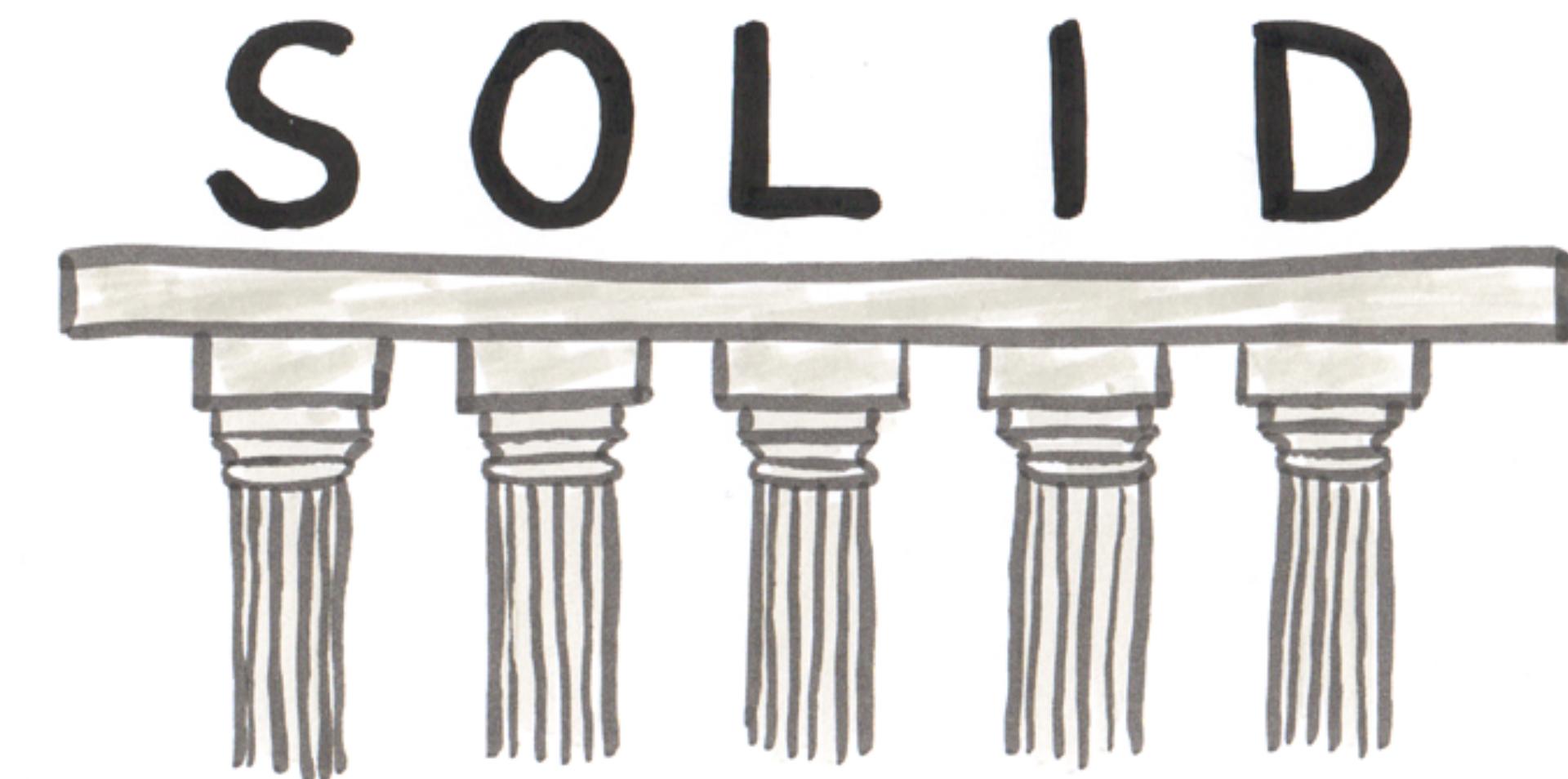


Image Source: FreeCodeCamp

We use programming to solve real-world problems, and it won't make much sense if one can't model real-world scenarios using programming languages. This is where object-oriented programming comes into play

From [educative.io](#)

# Agenda

## Module 02: OOP & S.O.L.I.D

### 1. OOP - Object-oriented Programming

- Definition
- Building Blocks: Object, Class, Attribute, Method
- Four main principles: Encapsulation, Abstraction, Inheritance, Polymorphism
- Encapsulation vs Abstraction
- Polymorphism: static vs dynamic

### 2. SOLID - Five principles of Object-oriented Design (OOD)

- Single responsibility principle (**SRP**)
- Open-Closed principle (**OCP**)
- Liskov substitution principle (**LSP**)
- Interface segregation principle (**ISP**)
- Dependency Inversion Principle (**DIS**)

# Software Architect

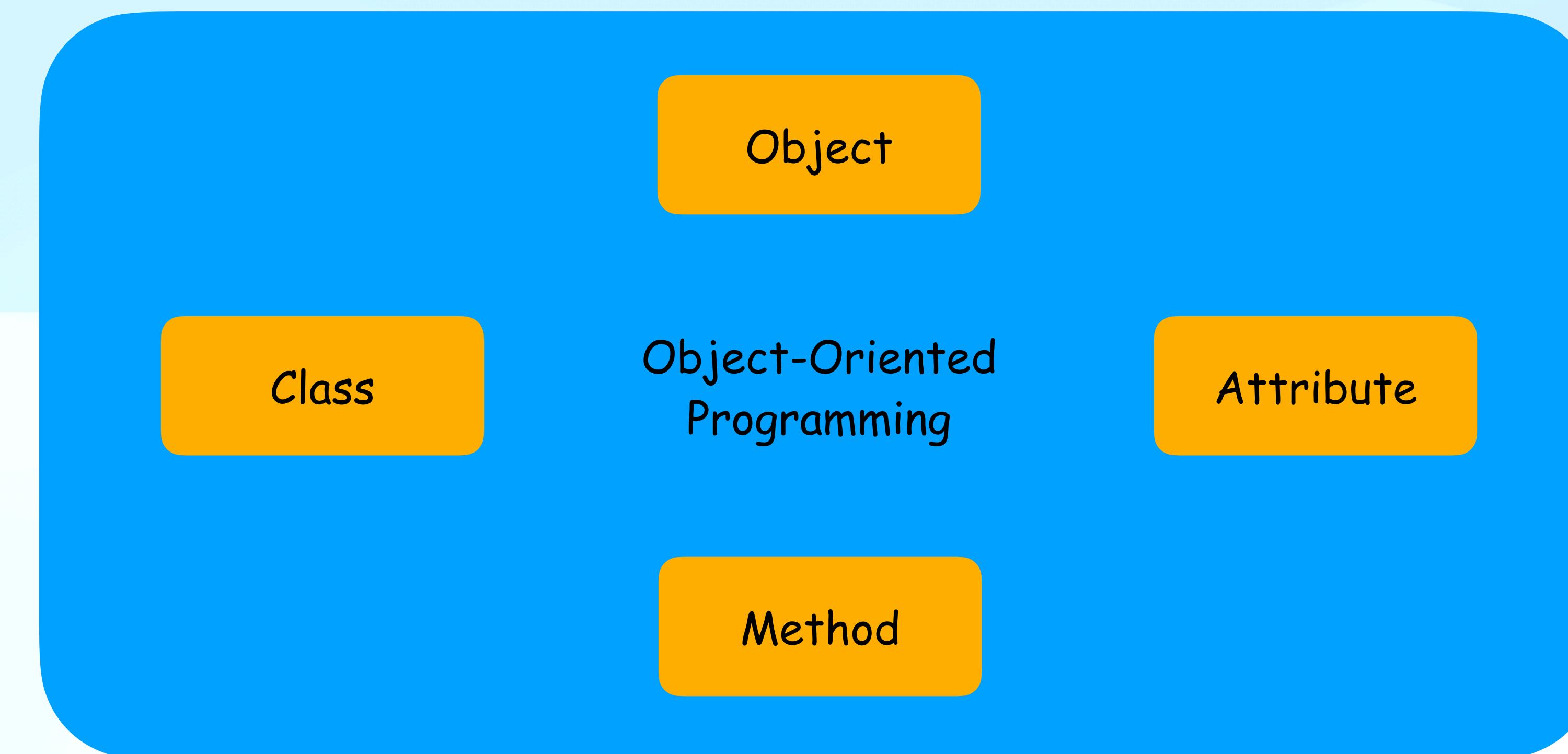
## 1. OOP (definition)

Object-oriented programming (OOP) is a **programming paradigm** based on the concept of "**objects**", which can contain **data** and **code**. The data is in the form of **fields** (often known as **attributes** or properties), and the code is in the form of procedures (often known as **methods**).

- wikipedia -

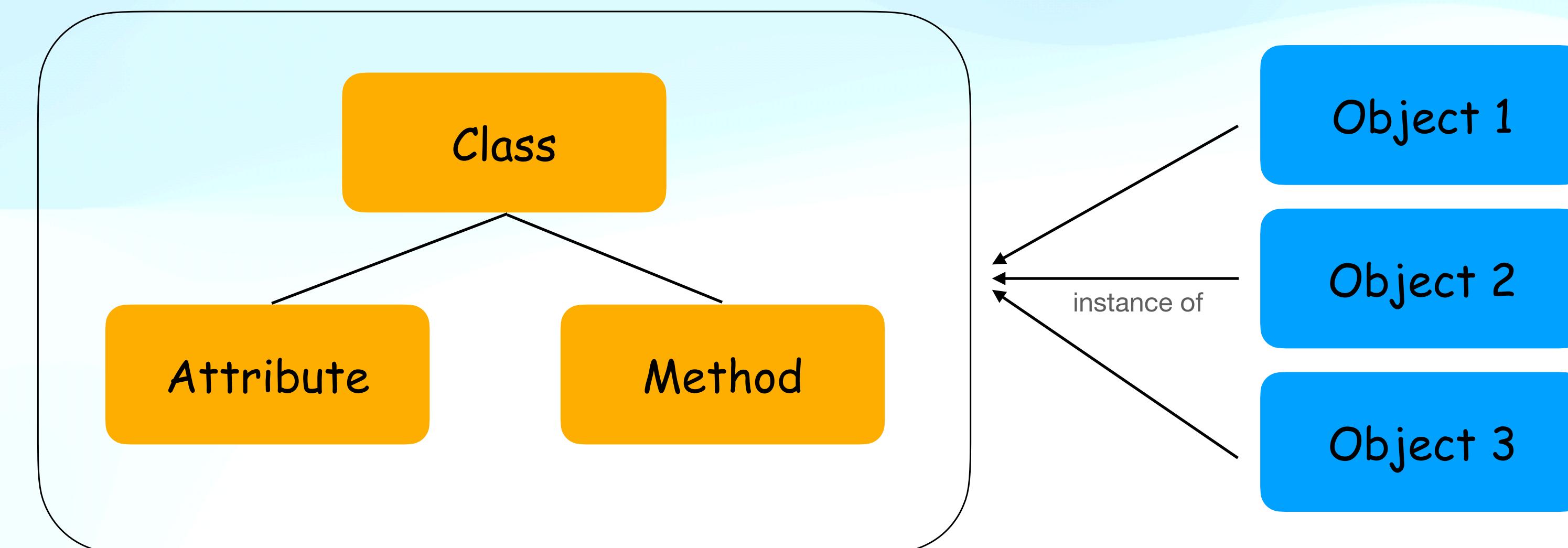
# Software Architect

## 1. OOP (building blocks) - cont.



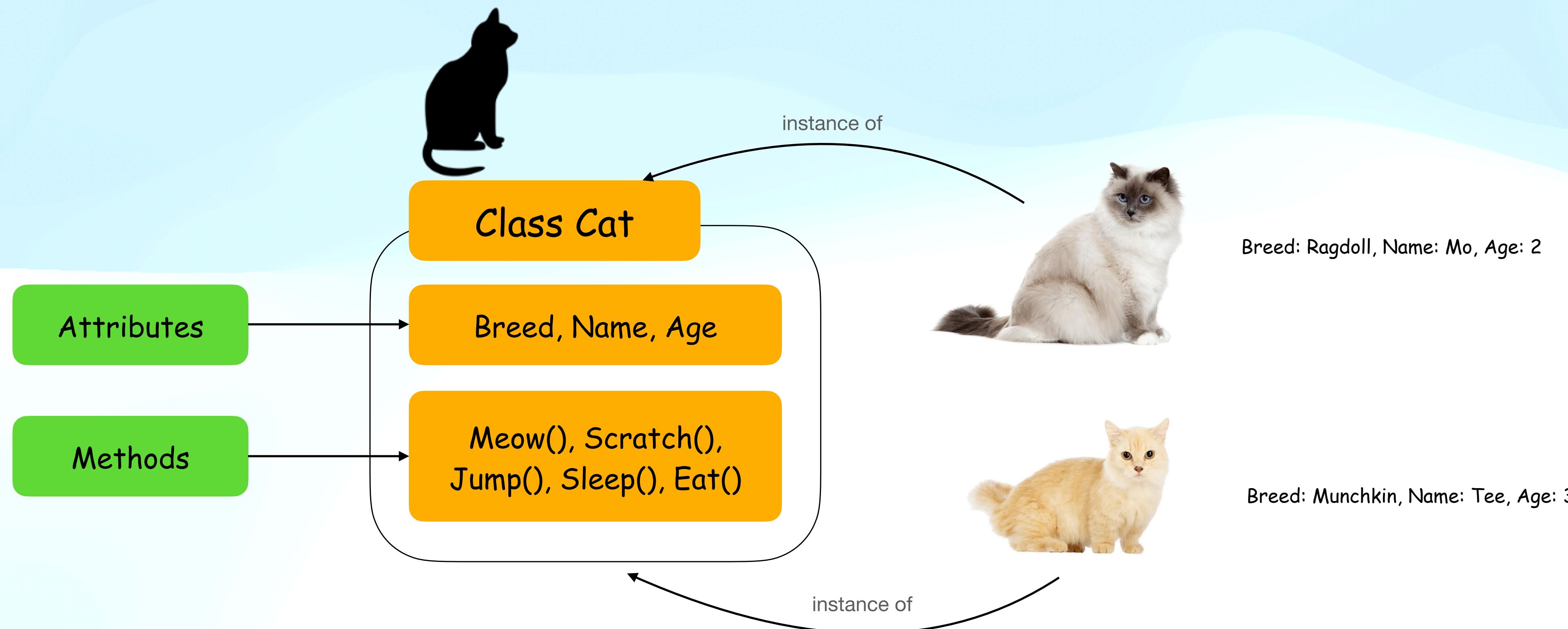
# Software Architect

## 1. OOP (building blocks) - cont.



# Software Architect

## 1. OOP (building blocks) - example.



# Software Architect

## 1. OOP - Four main principles

OOP - Object-Oriented Programming

Encapsulation

Abstraction

Inheritance

Polymorphism

# Software Architect

## 1. OOP - [Data] Abstraction

**Data abstraction** is a design pattern in which data are **visible only to semantically related functions**, so as to prevent misuse...

- wikipedia -

**Abstraction** is a technique used in object-oriented programming that simplifies the program's structure. It focuses only on revealing the necessary details of a system and hiding irrelevant information to minimize its complexity

- educative.io -

# Software Architect

## 1. OOP - Encapsulation

**Encapsulation** prevents external code from being concerned with the internal workings of an object...

- [wikipedia](#) -

**Encapsulation** is a fundamental programming technique used to achieve data hiding in OOP. Encapsulation in OOP refers to binding data and the methods to manipulate that data together in a single unit—class

- [educative.io](#) -

# Software Architect

## 1. OOP - Abstraction vs Encapsulation

Both encapsulation and abstraction are data hiding techniques in OOP. But they have some differences.

**Abstraction** is used for hiding unwanted data and giving relevant data to minimize its complexity.

=> **WHAT** object does.

**Encapsulation** is used for hiding by grouping data & code into a single unit to protect them from outside world.

=> **HOW** object does.

# Software Architect

## 1. OOP - Advantages of Abstraction & Encapsulation

- It helps classes simpler to modify and maintain.
- It reduces the complexity of the system from a user's perspective.
- It makes the code extendable and reusable

# Software Architect

## 1. OOP - Inheritance

In **object-oriented programming**, **inheritance** is the mechanism of basing an **object** or **class** upon another object (**prototype-based inheritance**) or class (**class-based inheritance**), retaining similar **implementation**.

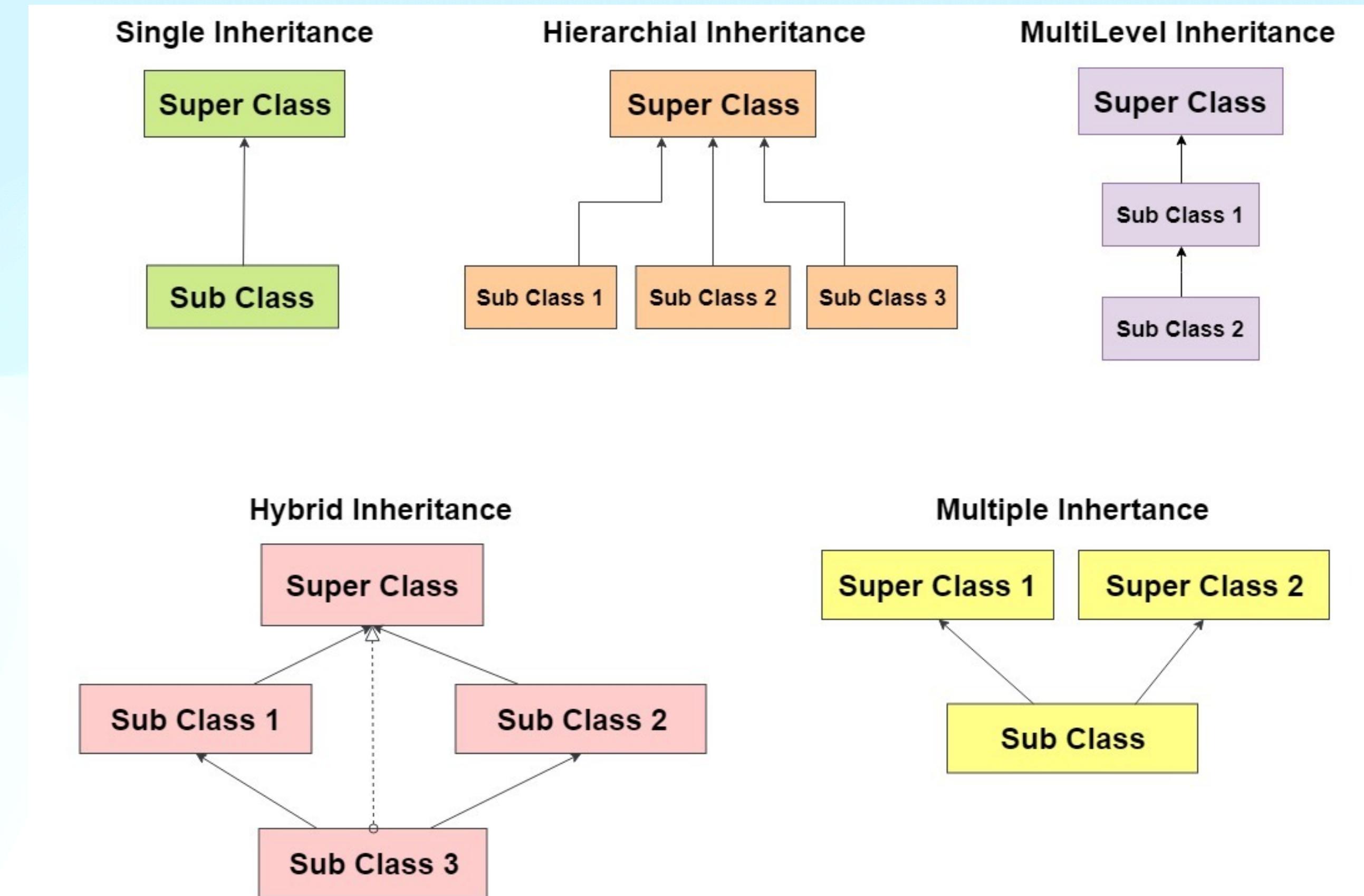
- wikipedia -

**Inheritance** provides a way to create a new class from an existing class. The existing class is used as a starting point or base to create the new class.

- [educative.io](#) -

# Software Architect

# 1. OOP - Inheritance (Types)

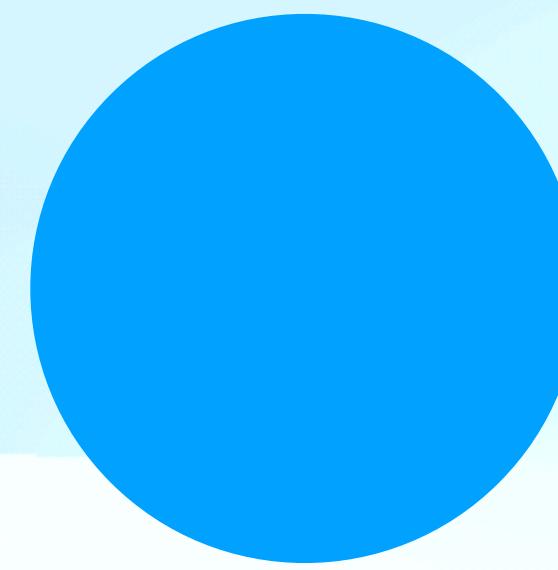


To reduce the complexity and simplify the modeling, I would like to ignore "multiple inheritance".



# Software Architect

## 1. OOP - Inheritance ("Is-a relationship")



Circle - is a Shape



Cat - is a Pet



Car - is a Vehicle



Tom - is a Cat

# Software Architect

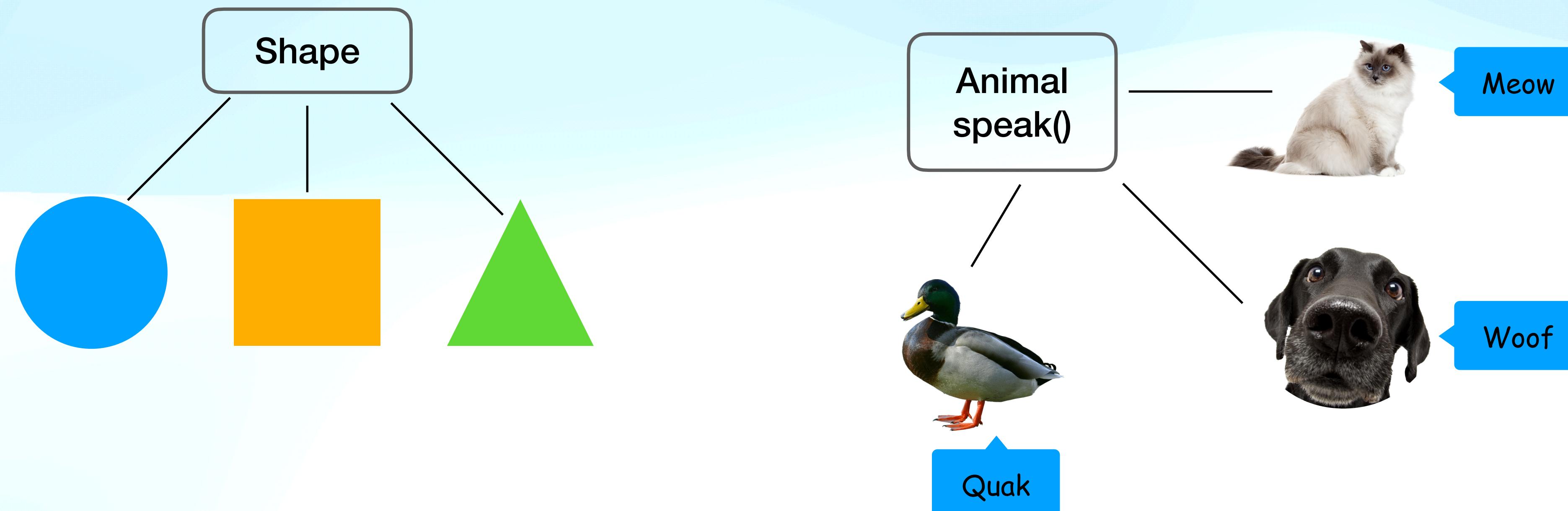
## 1. OOP - Polymorphism

The word **polymorphism** is a combination of two Greek words, “poly” meaning many, and “morph” meaning forms. In programming, polymorphism is a phenomenon that allows an object to have several different forms and behaviors.

- [educative.io](#) -

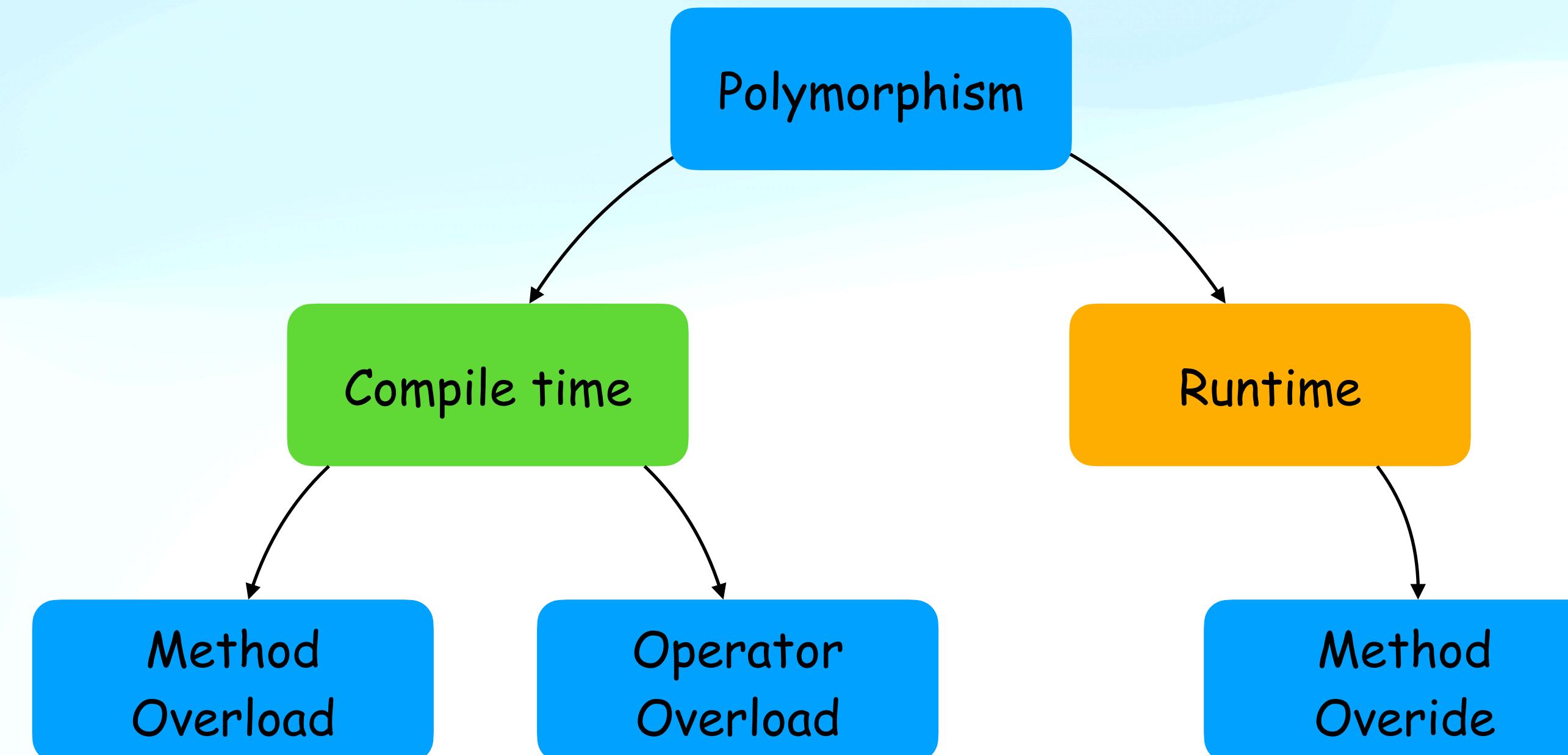
# Software Architect

## 1. OOP - Polymorphism (cont.)



# Software Architect

## 1. OOP - Polymorphism (static vs dynamic)



# Software Architect

## 1. OOP - Four main principles (recap)

OOP - Object-Oriented Programming

Encapsulation  
(grouping/hiding information)

Abstraction  
(hiding information)

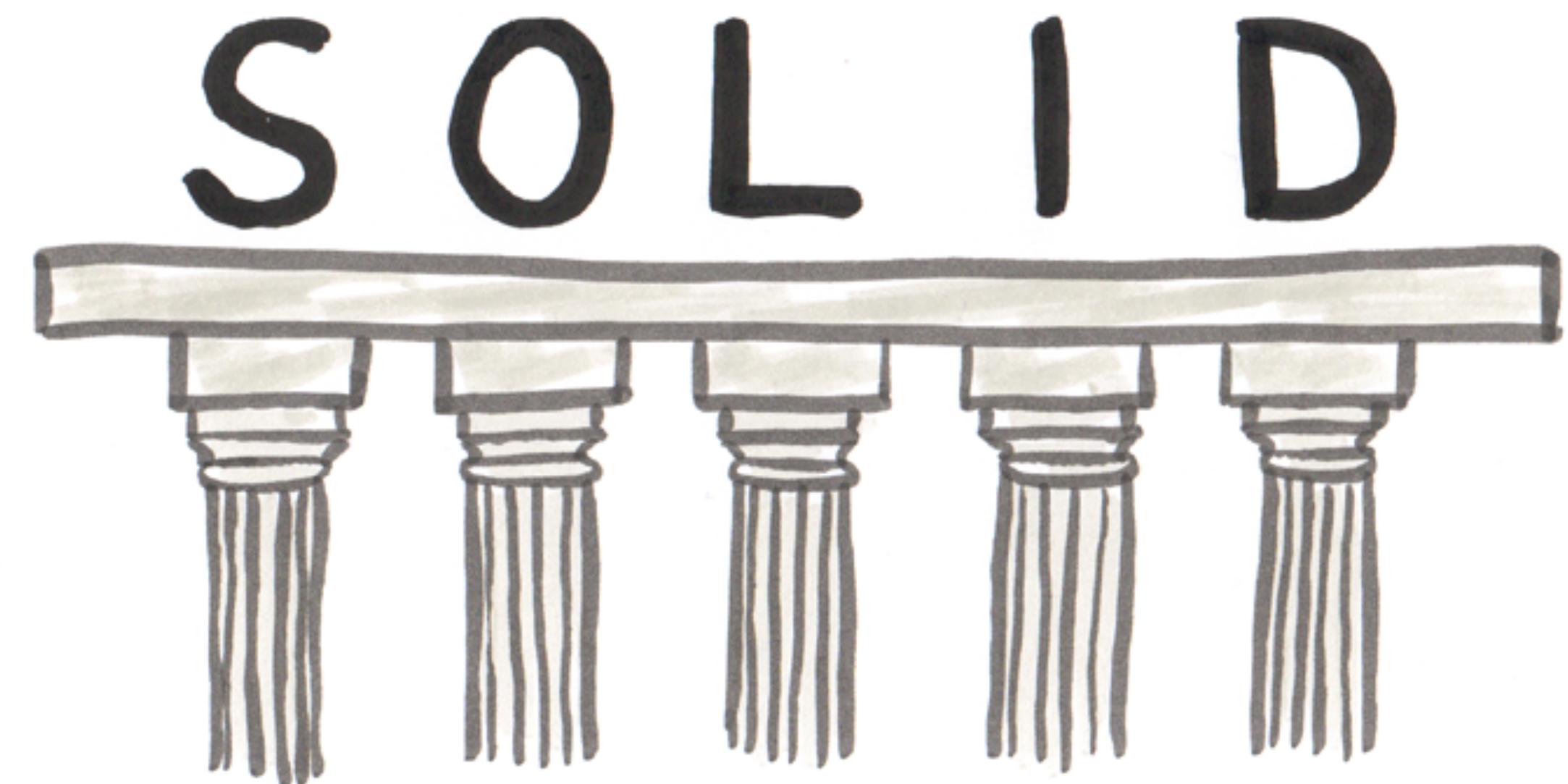
Inheritance  
(sharing information)

Polymorphism  
(redefine information)

# Software Architect

## 2. SOLID - Five principles of Object-oriented Design

1. Single responsibility principle (**SRP**)
2. Open-Closed principle (**OCP**)
3. Liskov substitution principle (**LSP**)
4. Interface segregation principle (**ISP**)
5. Dependency Inversion Principle (**DIS**)





Thank you