

## Design pattern

## Architecture pattern

### Definition

Specification that could help in implementation of a software

Fundamental Structural Organization for software systems

### Level of Focus

Design patterns are focused on solving specific coding-level problems within individual classes or components. They address common design challenges that arise at a lower level of abstraction.

Architecture patterns are focused on the overall structure and organization of a software system. They address high-level design decisions that affect the entire system.

### Granularity

Design patterns deal with the structure and interactions of objects within a single module or class. They provide solutions for implementing specific functionalities or relationships between objects.

Architecture patterns deal with the arrangement and interactions of major components, modules, and subsystems within a software system. They provide guidelines for designing the system's global structure.

### Usage

Design patterns are typically applied at a smaller scale and are used to enhance the structure, flexibility, and maintainability of code within a particular class or module.

Architecture patterns are applied at a higher scale and are used to define the fundamental structure, communication pathways, and major components of a software application.

### Examples

The Singleton pattern, Factory pattern, Observer pattern, Strategy pattern, and many others.

The Layered Architecture, Microservices Architecture, Model-View-Controller (MVC), Event-Driven Architecture, and more.

### Impact

Design patterns impact the organization, behavior, and relationships of individual objects or components. They are more concerned with the implementation details of a specific part of the software.

Architecture patterns impact the overall organization, scalability, maintainability, and performance of a software system. They guide the layout and interactions of components across the entire system.

