# SOLID

S → Single Responsibility Principle (SRP)

O → open / closed Principle (OCP)

L → Liskov Substitution Principle (LSP)

I → Interface Segregation Principle (ISP)

D → Dependency Inversion Principle (DIP)

## Single Responsibility Principle

One person → One job

SRP reduces the impact of change.

## Open / closed Principle

Don't break old things when adding new features.

OCP avoids modifying stable code.

## Liskov Substitution Principle

If it looks like a thing, it should behave like that thing.

Sub classes must not break parent behavior.

## Interface Segregation Principle

Don't force someone to do things they don't need.

Small interfaces reduce coupling.

## Dependency Inversion Principle

Depend on Contracts, not concrete things.

DIP improves testability and flexibility.

### ·> Spring Framework

Spring is built around Dependency Inversion using IoC containers.

### ·> Netflix

Netflix applies SOLID at microservices boundaries.

### ·> Amazon

Amazon's payment system follows Open-closed principle to support rapid extension.

### ·> Android

Android APIs follow Interface Segregation with listner-based design.