

SOLID is an acronym representing five design principles intended to make software designs more understandable, flexible, and maintainable.

These principles were introduced by Robert C. Martin, also known as Uncle Bob.

01. Single Responsibility Principle

A class should have only one reason to change, meaning it should only have one job or responsibility.

02. Open/Closed Principle (OCP)

Software entities (classes, modules, functions, etc.) should be open for extension but closed for modification.

03. Liskov Substitution Principle

Subtypes must be substitutable for their base types without altering the correctness of the program.

04. Interface Segregation Principle

Clients should not be forced to depend on interfaces they do not use.

05. Dependency Inversion Principle (DIP)

High-level modules should not depend on low-level modules. Both should depend on abstractions.

Course Requirements

1. Flutter

2. OOP