

Core Fundamentals

Pillar 00: Abstraction, Encapsulation, Modularity Hierarchy

Interface → Specification of an behaviour

Interface Represent some feature of a class

Kid is Jumpable he can Jump

What an object can do?

When to use? When you want to standardised and behaviour

Abstract class : Incomplete class that require further specification.

IS-A Savings Acc is a Account

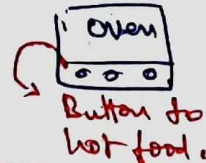
What an object is?

When to use? When you want to generalized a behaviour.

Abstraction : Design Principle of separating interface from implementation so that client only concern with the interface.

• Imp by using interface / Ab. Class

• Abstraction enforced by encapsulation.



• Encapsulation : Process of restricting access to inner implementation details of a class

Internal are not exposed to outside

= Sealed mobile ⇒ Don't open it - Warranty Voided.

⇒ Imp by private ⇒ Class Design

⇒ Ability to refactor / change ^{internal} code without breaking other code.

SOLID

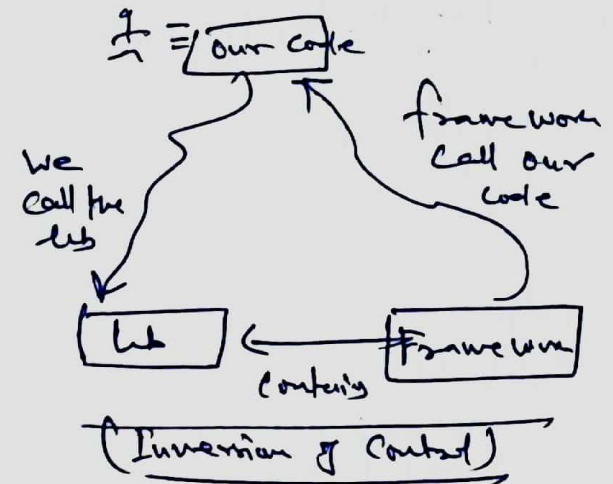
S - SRP

O - OCP

L - LSP

I - ISP

D - DIP



Smart

Generic

Design Pattern

Design Pattern?

Object oriented

* Design Pattern represent best practices used by experience SD developers

* GOF

* Design patterns are solutions to general problem that s/w dev face during s/w development

Design Pattern is a general, reusable solⁿ to commonly occurring problem within a given context in s/w design

⇒ It is an Concept ⇒ Not Readmade Code

• Creating

• Structuring

• Behaviour

Design pattern

Lib

Framework