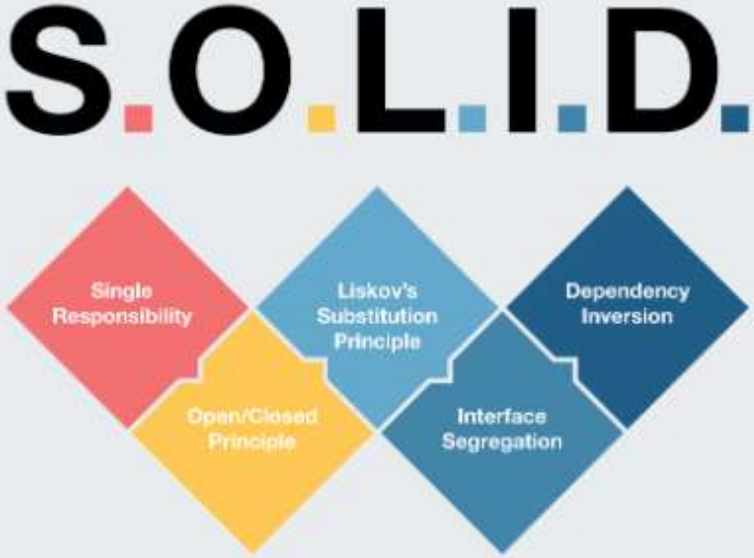
Open-Closed Principle

Part 3



Visit GitHub @BCAPATHSHALA

1. What is Open-Closed Principle (OCP)

1. An entity `(e.g., classes, modules, functions, etc.)` should be open for extension but closed for modification.
   This means you should be able to add new functionality without changing the existing code.
2. Extend functionality by adding new code instead of changing existing code.
3. **Goal:** Get to a point where you can never break the core of your system.
4. **Importance:** `OCP` encourages a more stable and resilient codebase. It promotes the use of `interfaces` and `abstract classes` to allow for **behaviours to be extended without modifying existing code.**
5. Writing code structure in such a way new functionality can be added by adding new code not by modifying existing code.
6. **OPEN** for extending and **CLOSE** for modification

In One Statement:
The Open-Closed Principle states that classes should be open for extension but closed for modification. This means that the behaviour of a class should be extendable without modifying its source code.

Key Idea:
Once a class is written, it should be closed for modifications but open for extensions.

Real-Time Examples:
1. Your smartphone – you don't open it up to add features; you just download apps to extend its capabilities.

How can Open-Close Principle be applied?

Visit GitHub: @BCAPATHSHALA

Practical Coding Examples in Java #1
Practical Coding Examples in Java #2
Practical Coding Examples in Java #3
Much more about ***Open-Close Principle***