

# LLD Interview Notes: DRY, Rule of Three & Abstraction

## 1. What is DRY (Don't Repeat Yourself)?

DRY means maintaining a single source of truth for knowledge in a system. Knowledge includes business rules, validation logic, message formats, calculations, and configurations. DRY is not about removing duplicate lines, but about avoiding multiple places that represent the same decision or rule.

### ***Example (Conceptual)***

If tax percentage is written in multiple services, changing tax requires multiple updates. Instead, tax should exist in one place so the change happens once.

## 2. What is Abstraction?

Abstraction means hiding implementation details and exposing only what is necessary. In LLD, abstraction is used to group common behavior so that multiple components depend on a shared concept rather than duplicated logic.

### ***Example (Conceptual)***

Instead of each service formatting messages differently, a formatter abstraction defines how messages should look, while services simply request a formatted message.

## 3. What is the Rule of Three?

The Rule of Three states that abstraction should be introduced only after the same logic appears three times. One occurrence may be accidental, two may still be coincidental, but three indicates a stable and repeatable pattern.

### ***Why the Rule of Three Exists***

It prevents premature abstraction and over-engineering. Creating shared components too early can lock the design into assumptions that later turn out to be incorrect.

## 4. Why Abstraction Can Fail (If Done Too Early)

Early abstraction often combines logic that only looks similar but has different business meaning. When requirements diverge, the abstraction becomes complex, fragile, and harder to maintain.

### ***Typical Failure Scenarios***

- Different rules evolve independently but are forced into one abstraction
- Excessive parameters added to support edge cases
- Reduced readability due to generalized logic

## 5. Correct Approach in LLD (Interview Perspective)

First, observe duplication and identify whether the duplicated code represents the same knowledge. Second, allow duplication until the pattern stabilizes. Third, once the third occurrence appears, extract the logic into a dedicated class or abstraction that becomes the single source of truth.

### ***Summary for Interviews***

DRY reduces maintenance risk by centralizing knowledge. Abstraction is the tool used to achieve DRY, but only when repetition is proven stable. The Rule of Three ensures abstractions are created at the right time, not too early and not too late.