

Class starts in 5 minutes

Ankit

- ankit.aerona-1@escaler.com
- 7337057594 ←

Agenda

- ↳ What is LLD? -
- ↳ Why is LLD important? -
- ↳ Structure of LLD module
- ↳ Expectation setting for LLD
 - ↳ Language
 - ↳ Types of assign
 - ↳ amount of theories
 - ↳ CS fundamentals

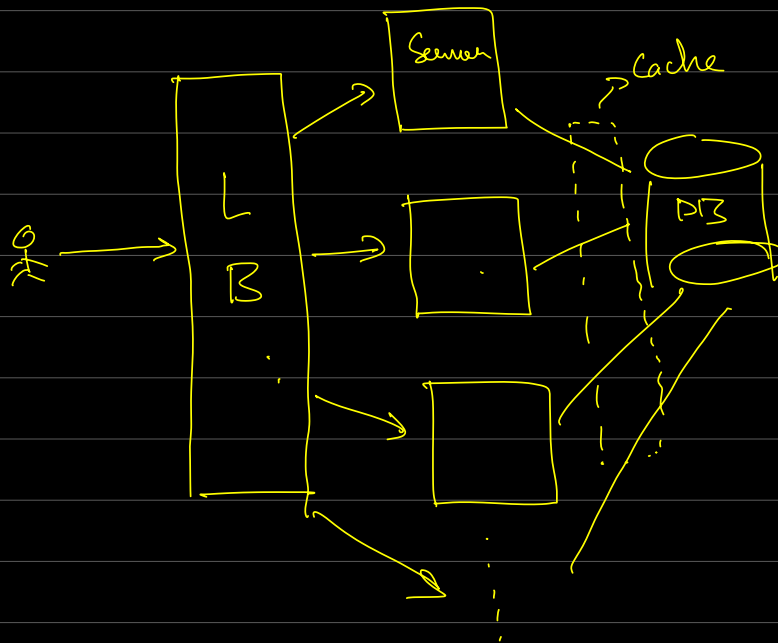
What is LLD

- ↳ Low Level Design

HLD ⇒ High Level Design
↓
Overview

- ↳ An overview of how a software system will look like.

- ① Latency
- ② Crash
- ③ Single point of failure
- ④ User experience will be bad.



Overview of different infrastructure layers that work together to serve an app at scale at desired efficiency

Each system is nothing but a computer which acts per what program is running on them.

Details of a software system will describe the code/program that is running.

↳ L2D => Details of the code that you write.
 --- => Design of " " " " " "

Why LLD?

- ↳ 12% of their time writing code.
- ↳ 1 hour

- Documentation (Design, research) \Rightarrow Design
- Debugging (or optimizing) \Rightarrow understanding around code.
- meetings \Rightarrow req, gathering, discussion
- Testing \Rightarrow
- Code reviews \Rightarrow Read & understand code

helps us
LLD, writes code :

- ↳ understandable -
- ↳ reusable \Rightarrow DRY
- ↳ extensible
- ↳ maintainable
- ↳ modular -

Extensible Code

- ↳ How easy is it for your s/w system to be modified to be able to handle new req.

Maintainable Code

- ↳ bugs fixes -
- ↳ optimizations -
- ↳ dependency updates
- ↳ platform update

Interviews

Exp.

Type of companies

SDE1 0-2

2 2-5

3 5+

4

Startups → Flipkart, CRED, Swiggy.

MNC → Cisco, Paypal, Adobe

AAN

FG

LLD	↓ Startups	↓ MNC	↓ AAN	↓ FG	DSA HLD
SDE1	Machine Coding	Theoretical	Design. Ho Code	X	
SDE2	MC	"	"	X	
SDE3	MC	"	"	X	

Structure of LLD module

4 parts :

LLD1

↳ Foundational design / language concepts

OOPs ⇒ LLD.

3-4
↳ Object Oriented Programming
OOPs Abstract Classes
interface

5 → Concurrency (Java)
Java lambda
Collecting

(1) Day to day work
(2) Imp for interviews

LLD2 : Design Principles & Patterns

↳ SOLID - 2

↳ Design Patterns - (10 D.P.) - 3

↳ UML diagrams

LLD3 :

↳ Mini Case studies

↳ Design BMS

Splitwise

Tic Tac Toe

Parking Lot

MVC ✓

Spring Boot ✓

LLD4 : Project Module ✓ — All web ←

↳ Ecommerce website

Auth
Implement APIs

→ Cloud deployment

SQL DB

GIT ⇒ VCS

Spring Boot

UI ✗

What?

Kafka
Redis Cache

8:39 → 8:44 (5 minutes)

Why Java

name private String name;

[No prior Java knowledge is expected.]

Intelli IDEA ← Community version
Scaler support for paid version

Assignments

↳ Assignments are in Java.]

Amount of theory

CS Fundamentals

DBMS, OS, CN → LLD
↓
SQL
Concurrency
[CPU Scheduling and memory management]
↳ Support Scaler

Job Readiness

SDE1 \Rightarrow DSA + Lang + LLD^{SQL} mock interview
SDE2+ \Rightarrow " + HLD

Int/Adv Batch vs Beginner

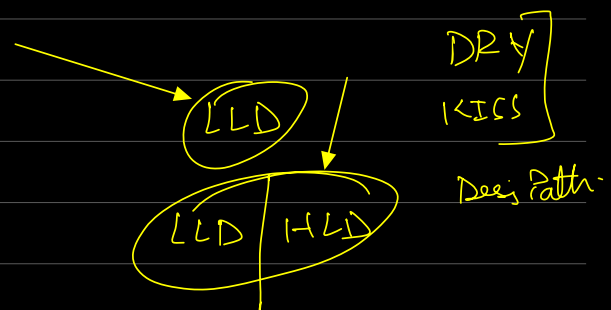
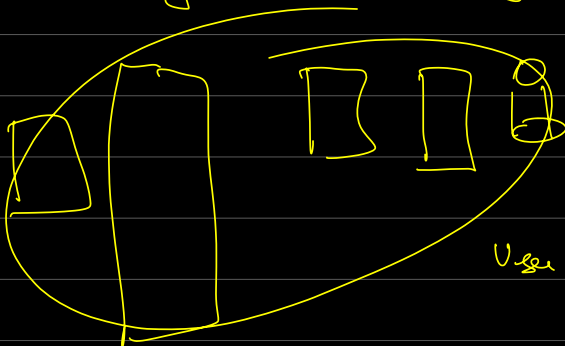
2:30

\hookrightarrow Prep. + Extra classes

\hookrightarrow prior dev experience or
if you have a comprehensive project in
your portfolio

Doubts

Infrastructure layers



$\circ \rightarrow \circ \rightarrow \circ \rightarrow$

User entry

\downarrow

verification

\downarrow

HLD

\Rightarrow LLD

BMS
↳ ticket bn.

Ticket → Book