WELCOME TO LLD MODULE

Nikhil Jain — Sr. SDE L6 — 1 year
Amazon

Us Candau India Pricing

Walmart — Staff — 3 years

Microsoft — SDE 2 — 2.5 years
Bing

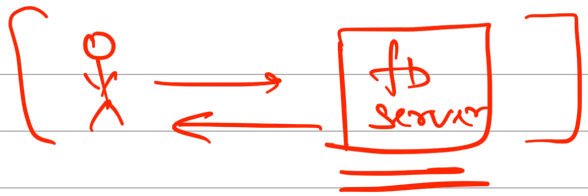
Amazon — SDE1 — 2 years
Image processig

Misys (Finstara) — 8 Months
Loan 7Q

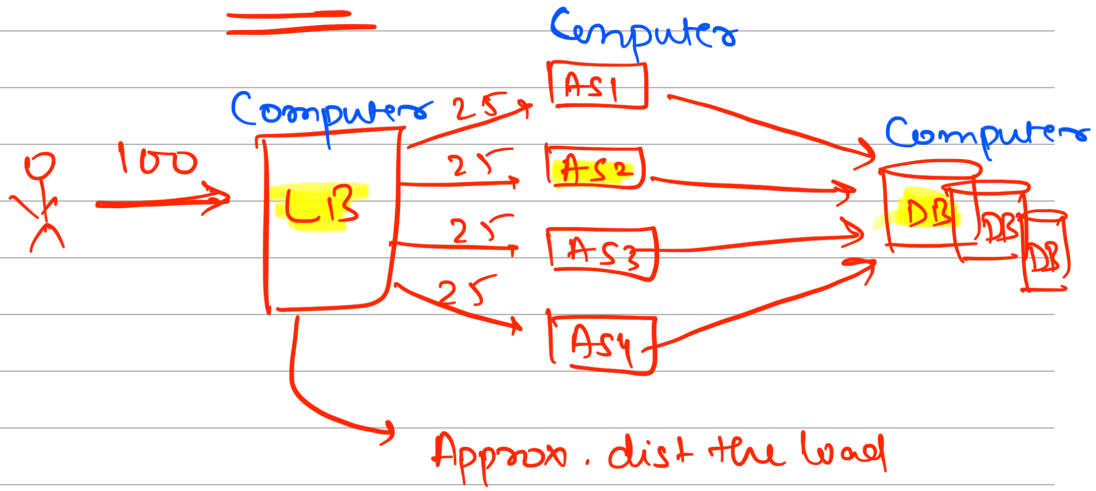
LLD (Low Level Design)

Low <=> high

High level Design



fb = 2.1 billion users



Typical HLD

HLD is study of Infra layer and way of interaction

Different infra layers are working together inorder to make entire system worke efficiently

At the end, Servers | DB | LB all are nothing but computers runing different S|W

LLD = Details of the S|W running on the Machine

↳ Details of Code → structure
↳ stack
↳ schema

↳ Object Oriented Design

On an average how much time a SWE spends in writing code

2-3 hrs

40 LPA
8 hrs
10 LPA

1) Meetings
2) Playing
3) Chai Sutta break
4) Req Gathering
5) Debugging
6) Testing
7) Code Review
8) Documentation

All ⇒ Code

∴ 12% of time is spent coding
88% about how to code

LLD helps to make the better use of 88% time

LLD
- Readable
- Understandable
- Maintainability
- Extensibility

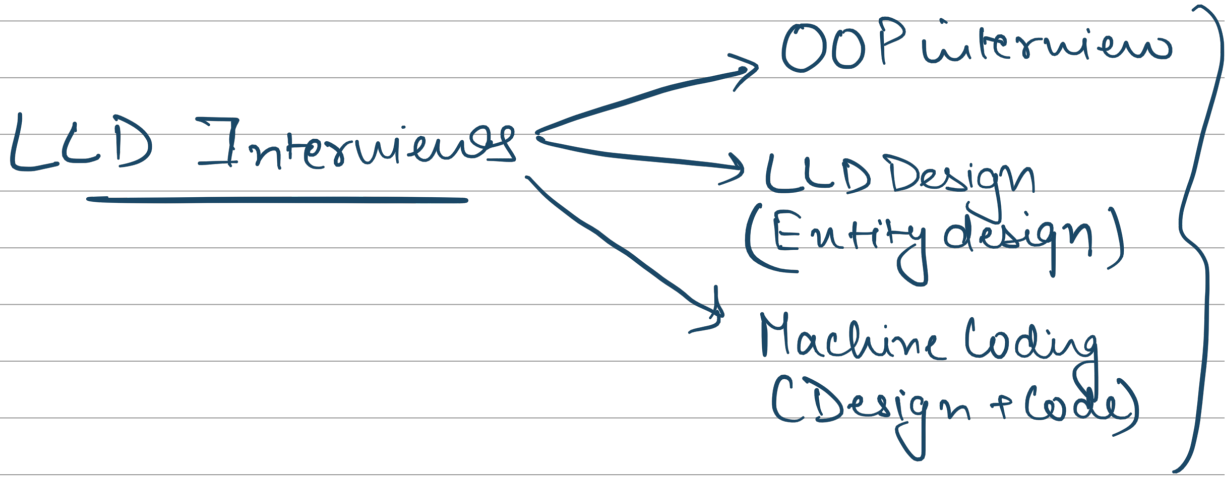
Extensible: Easy to add new features

Maintain: Easy to keep running as it is (No New feature addition)

- Bugs
- Platform update
- Library update

(Log4j) 98% → Security Vulnerability

LLD is important (Interview)



FB | Google → LLD + DSA + LLD Code

MS | Amazon | Atlassian | Adobe | MNCs → LLD or OOP → Design, No Code

{AMEX, GS}

Flipkart | Scaler | Cred | Phonepe → Machine Coding Design + Code 2.5 hrs – 3 hours

LLD Module

4 Classes OOPs
4 Multithreading | Sync

3 JAVA → Interviews
- Collections
- Generics
- Exception handling
- Reflection

Design → SOLID
- Design Patters
- Interview Case
  - Entity Design
  - Game Design
  - Real App Design
  - Infra System Design

Intro to OOPs

Programming Paradigms → Patterns or Model

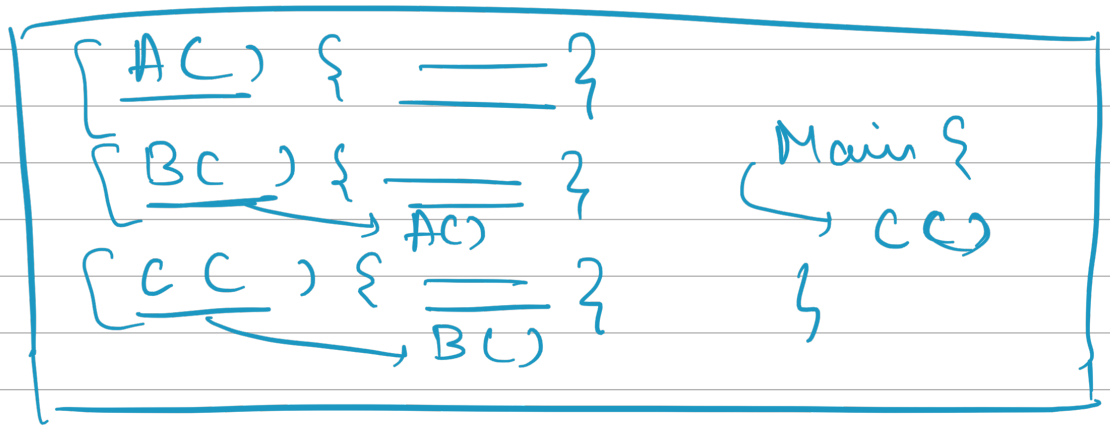
1. Procedural → C
2. Object Oriented → JAVA | C++ | C# | Python
3. Functional → Scala | Java
4. Reactive → React

Procedural

↳ Procedure : set of instruction

↳ old name of functions / Methods { (DataSet) ↓ set inst (O/P) }

1. We organise our code into bunch of procedures
2. One fun^{n} call the other fun^{n} { A() → B() → C() }
3. execution of procedures starts from special procedure that is called MAIN



1. Sneha is learning
2. Manoj is listening
3. Nikhil is teaching
4. Shailendra is watching ipl

{ Someone is doing is something }

{ Subject + Verb = Action }

```
printStudents ( String name,
                int rollno,
                String batch)
{
    print( name);
    print( roll no);
    print( batch)
}
```

Scalar

Non Readable

```
Struct Student {
    String Name
    Int Roll No
    String Batch
}
```

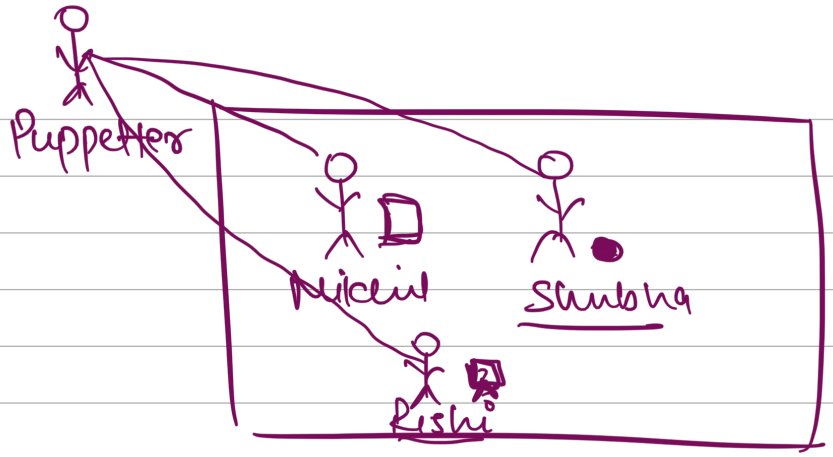
```
printStudent ( Student st)
{
    print( st.name)
    print( st.rollno)
    print( st.batch)
}
```

Procedural [Something is happening over someone

Real life [Someone doing something]

In Struct,

1. No Methods
2. All the variable are accessible to everyone



Real

Someone is doing Something

working(Rishi)

[Something happening on someone]

Bank

[Bal deduction (Rishi)]

Something → Rishi

unauth

Thing happening on Actor

OOP

[Entities should do things]

↳ Software System should have entities

Each entity can have attributes or functions

```
Class Student {
    String Name;
    int rollNo;
    String Batch

    print(Student st)
    {
        print(st.name);
        print(st.rollno);
        print(st.Batch)
    }
}
```

```
Student.print(st);

Someone is doing Something
```

OOP

⇒ Entities are core of everything

⇒ Entities → ~~Attributes~~
  → Methods

Entities → Real World Entities (Student Inst)
  → Conceptual Entities (Class Batch)

Entites → Students
  → Instructor
  → Batch
  → Class
  → Interview

How Many pillars of OOPs

1 principle.
+
3 Pillars of OOP

Java complete Reference

Principle: → Rule | fundamental

ABSTRACTION | to hide details

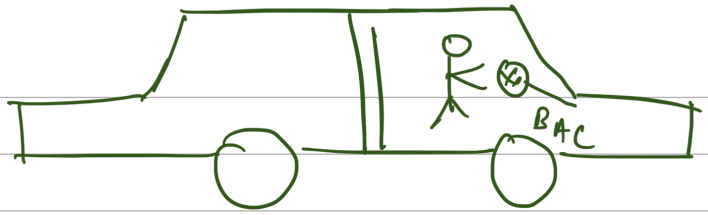
Pillars: → support the principle of OOP (ABSTRACTION)

- ENCAPSULATION
- INHERITANCE
- Polymorphism

Abstraction: hide details

Represent in terms of ideas → Real entity
→ Conceptual

Anything in S/W is a Idea → (Entity)
& Idea contains attrs or Methods or Both



Abstraction

- Ideas → Attributes
  Methods
- Others need not to know internal working.
  Only required data is exposed.

→ LLD, why it is imp
→ OOPs, Pillars / Principle of OOP

Procedural vs OOP

Shop to buy shampoo

Shopkeeper ⟶ wallet take money

Procedural

Paste
deductAmount(wallet) ← User Shubham

{

}

UNNatural

[Shubham.deductAmount( )]

Entity