⊘ Lecture 04 : What is UML Diagrams | Class & Sequence Diagrams with Real Examples **≥**

X Introduction - UML Diagrams Kya Hain?

Basic Concept

- UML (Unified Modeling Language) ek visual representation hai 🜎 software design ideas ka.
- Ye ek **diagrammatic approach** deta hai jisse long paragraphs ki jagah cheezein easily samajh aati hain .
- Dikhata hai components, objects, entities aur unke interactions 🖸.

OUNL Diagrams ka Purpose

- Apne dost ya team tak idea pahunchane ka better tareeka 🕮
- Boring paragraphs likhne ki jagah visualization 🚫
- Diagrammatically samjhana 📊
- Components aur interactions ko visually represent karna @

1 UML Diagrams ke Types

■ Do Main Categories:

Structural Diagrams (Static) 🍙 🛮 Behavioral Diagrams (Dynamic) 🖸

Object Diagram

∑ State Diagram

Component Diagram

Deployment Diagram
Output
Use Case Diagram



| Important Facts:

- Total 14 UML Diagrams hote hain 1 4
- LLD Interviews ke liye mostly 2 diagrams important hain 6
 - Class Diagram (Structural)
 - Sequence Diagram (Behavioral)
- 99% interviews me Class Diagram pucha jata hai 📊
- Sequence Diagram specific use-cases ke liye <a>
 <a>

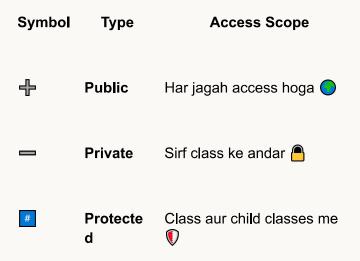
🏛 Class Diagrams (Structural) 锅

Basic Class Representation

```
Example: Car Class
```

```
срр
class Car {
private:
   string brand;
    string model;
   int engineCC;
public:
   void startEngine() {}
   void stopEngine() {}
   void accelerate() {}
   void brake() {}
};
UML Representation:
        Car
   - brand: string |
   - model: string |
   - engineCC: int
   + startEngine() |
   + stopEngine()
   + accelerate()
   + brake()
```

Access Modifiers



Abstract Classes

- Class ka naam italic hota hai ya abstract keyword likha jata hai.
- Abstract classes instantiate nahi hoti
- Ye abstract methods contain kar sakti hain.

Class Relationships (Associations)

III Relationship Hierarchy

```
Associations Class Level

Class Level

Inheritance (Is-A Relationship)

Object Level

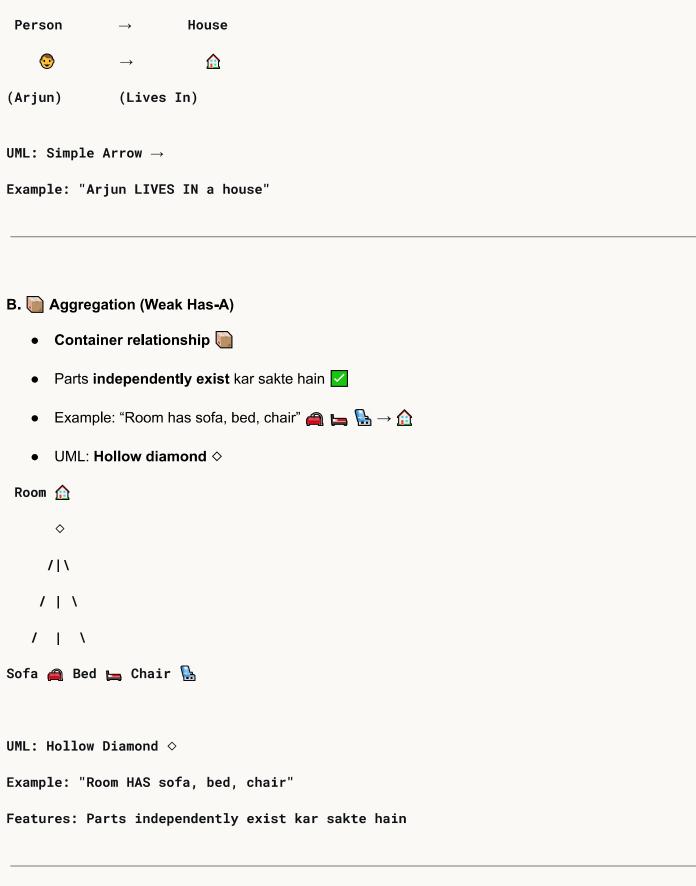
Simple Association

Aggregation (Weak Has-A)

Composition (Strong Has-A)
```

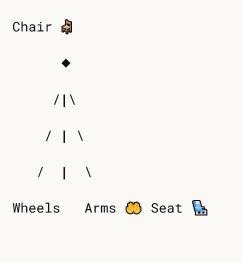
Inheritance (Is-A Relationship) 🍱 Real-Life Examples: "Cow is an Animal" ¬ ♥ "Tiger is an Animal" $_{f m} ightarrow \overline{f W}$ C++ Code Example: срр class Animal { public: void breathe() { } }; class Cow : public Animal { public: void eatGrass() { } }; **III** UML Notation: Animal 🐾 Δ Cow 😘 → Solid line with hollow arrowhead (♠) Composition Relationships (Has-A Relationship) A. Simple Association ● Sabse weak relationship □

UML: Simple arrow →



C. S Composition (Strong Has-A)

- Sabse strong relationship
- Parts independently exist nahi kar sakte X
- Example: "Chair has wheels, arms, seat" \square \bigoplus \bigoplus \longrightarrow \clubsuit
- UML: Filled diamond ◆



UML: Filled Diamond ◆

Example: "Chair HAS wheels, arms, seat"

Features: Parts independently exist NAHI kar sakte

🖸 Sequence Diagrams (Behavioral) 🔯

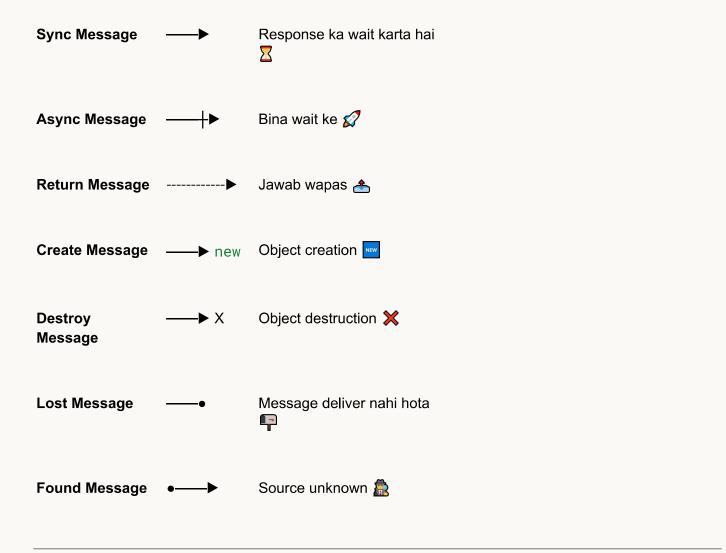
Purpose aur Importance

- Dikhata hai objects ke interactions time order me
- Communication/Interaction flow visualize karta hai
- Objects kaise messages exchange karte hain

Main Components

- 1. **Objects** Rectangles ke form me top par Example: User, ATM, Transaction
- Activation Bars Lifeline par small rectangles
 Dikhata hai object kab active hai
- 5. Message Types

Type Symbol Description



ATM Example - Sequence Diagram

Involved Objects:

- 🙎 User
- ■ ATM
- = Transaction
- 🔡 Account
- @ CashDispenser

Flow:

- 2 ATM → Transaction: new Transaction() WW
- 3 Transaction → Account: checkAmount(amount)
- 4 Account → Transaction: return true ✓
- $\hline \textbf{5} \ \textbf{Transaction} \rightarrow \textbf{CashDispenser: withdrawCash(amount)} \ \underline{\textbf{60}}$
- 6 CashDispenser → User: return amount 6

Control Structures

- $\bullet \quad \text{alt} \rightarrow \text{if-else conditions} \ {\color{red} \ } {\color{red} \ } {\color{black} \ }$
- loop → loops (for/while) □
- opt → optional interactions

COMPLETE ATM Sequence Diagram - Step by Step

text

[Figure 13: ATM Withdrawal Complete Sequence Diagram]

Objects: [User] [ATM] [Transaction] [Account] [CashDispenser]

User	ATM	Transaction	Account	CashDispenser	
1	I	1	I	I	
—withdraw(a	amount, accN	No) - ▶	1	I	
1	1	I	1	1	
1	—create-	-▶[Transaction]	1	1	
1	1	I	1	1	
I	1	-checkAmou	ınt(amount)-	→	
1	1	I	1	1	
I	1	∢ —return †	true—	I	
1	1	I	1	1	
I	1	—withdrawC	ash(amount)	→	
1	1	I	1	1 1	
I	1	I	1	—dispense()—▶	
I	1	I	1	1 1	
∢ —return amount———					
1	1	I	1	1	
I	I	[X]	I	1 1	

6 Interview Focus

- 99% LLD interviews me Class Diagram pucha jata hai 📊
- Sequence Diagram tab useful jab flow complex ho
- Dono ka **practice karna** zaruri hai 🟆

Real-World Applications

- Inheritance = "is-a" relationship
- Composition = "has-a" relationship 🐒
- Sequence diagrams = workflow understanding

Quick Revision Chart

Concept	Relationship	UML Symbol	Example
Inheritance	ls-A	^	Cow → Animal 🏠 → 😈
Simple Association	Has-A		Person → House ⊕→♠
Aggregation	Weak Has-A	>	Room → Furniture 🏚 → 👜
Composition	Strong Has-A	-	Chair → Wheels 🖨 → 🗆

Practical Tips:

- Class Diagram code likhne se pehle zaroor banayein
- Sequence Diagram complex flows ke liye use karein 🖸
- Composition ko inheritance ke upar prefer karein 6

6 Yaad Rakhne Wali Baatein:

- 14 UML Diagrams me se sirf 2 main hain 6
 - Class Diagram → Structure dikhata hai 🖀
 - \circ Sequence Diagram \rightarrow Behavior dikhata hai \square
- Interview me 99% Class Diagram hi pucha jata hai IIII