

Lect 4: UML Diagram

Page No.

Date 15 05 25

Why UML diagram is used?

- Suppose you have certain idea of application and you have to share it with your friends - One way is to send paragraphs (boring to read and can cause confusion) so you use diagrams to represent it.

Types of UML Diagram

Structural (static)

- ↳ how the structure of your application will be
- ↳ components & their relation
- ↳ Total 7 diagram

Behavioural (Dynamic)

- ↳ how the objects will behave
- ↳ Total 7 diagrams

We will learn only 1 structural diagram -

Class diagram

- We will learn only 1 behavioural diagram -

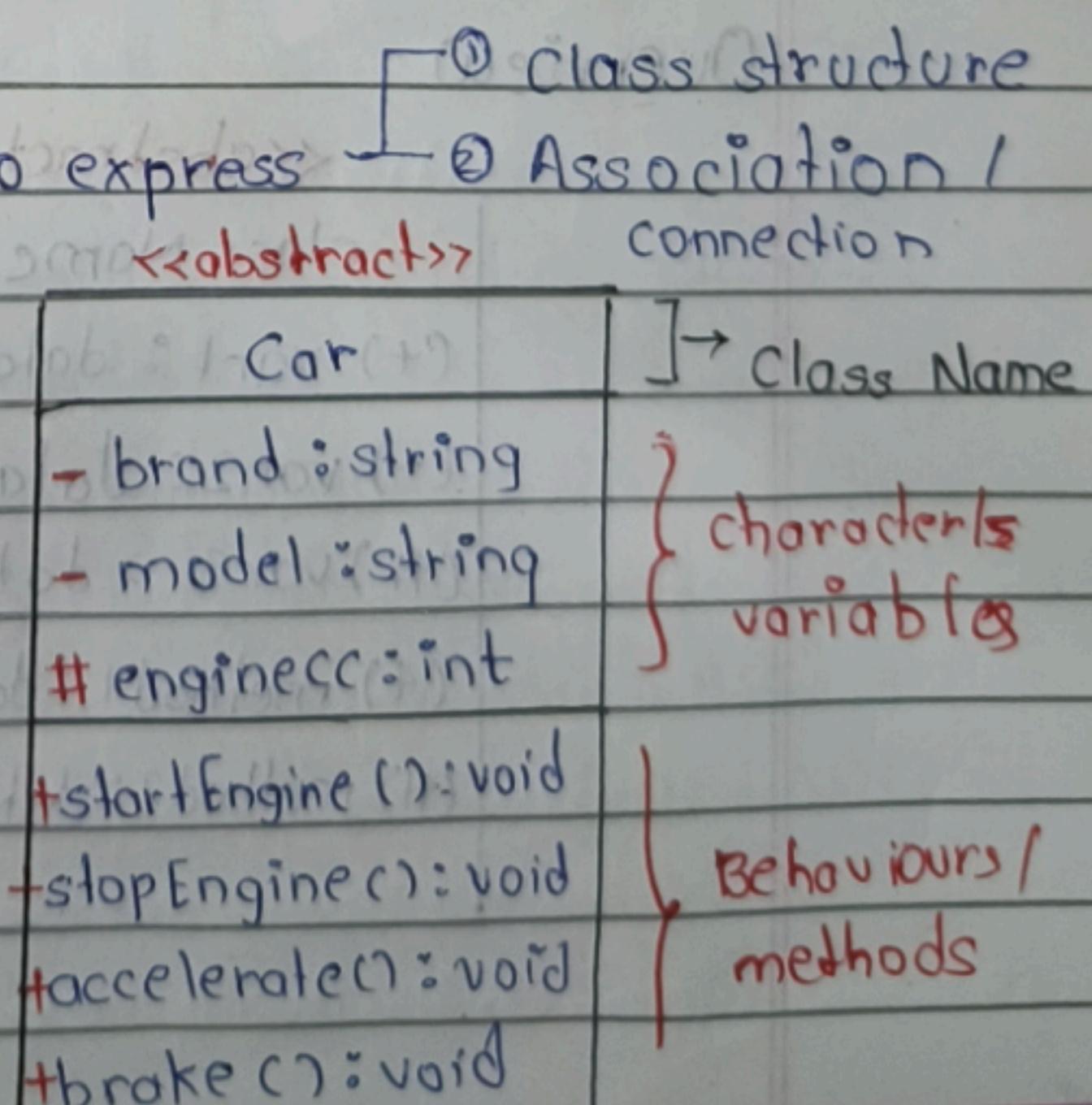
Sequential diagram

Class diagram

↳ We have to learn how to express

Ex:

```
class Car {  
    private:  
        string brand;  
        string model;  
        int enginecc;  
    public:  
        void startEngine();  
        void stopEngine();  
        void acceleration();  
        void brake();  
}
```



- How to represent access modifier?

Recap + Representation of access modifier

	Within class	From child class	outside class	Represent
public	(✓)	(✓)	(✓)	+
protected	(✓)	(✓)	(✗)	#
private	(✓)	(✗)	(✗)	-

- How to define abstract class & concrete class

Abstract class : matlab jismein virtual ~~class~~ hote hai - yaane ki aise method jinko declare class mein kiya hai par define child class kregi

↳ toh ise declare karna hai toh bass woh class ke box ke upar <<abstract>> ka lithdo

Concrete class : likhne ki jaroorat nahi kuch bhi

Overview

<<abstract>>

	class Name
(+)	var1 : datatype
(-)	var2 : datatype
(#)	var3 : datatype
(+)	method1 : datatype
(+)	method2 : datatype

How to show connection between classes:

Association

Types of Association

Class Association

Inheritance

(Is-a relation)

↳ ManualCar is
a car

Object Association

simple

Association

Aggregation

Composition

(has-a relation)

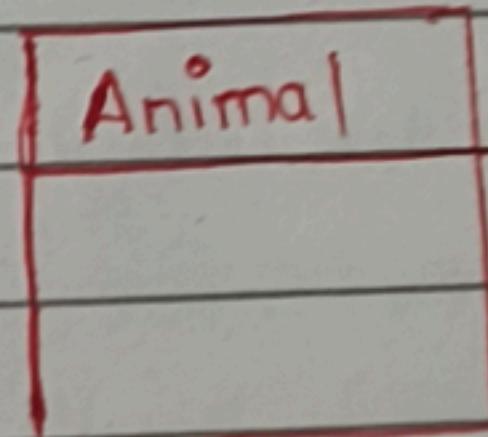
• Inheritance

How to represent it →

```
class A {  
    method1();  
}
```

```
class B : public A {  
    method2();  
}
```

```
*main()  
{  
    B * b = new B();  
    b->method1();  
    b->method2();  
}
```



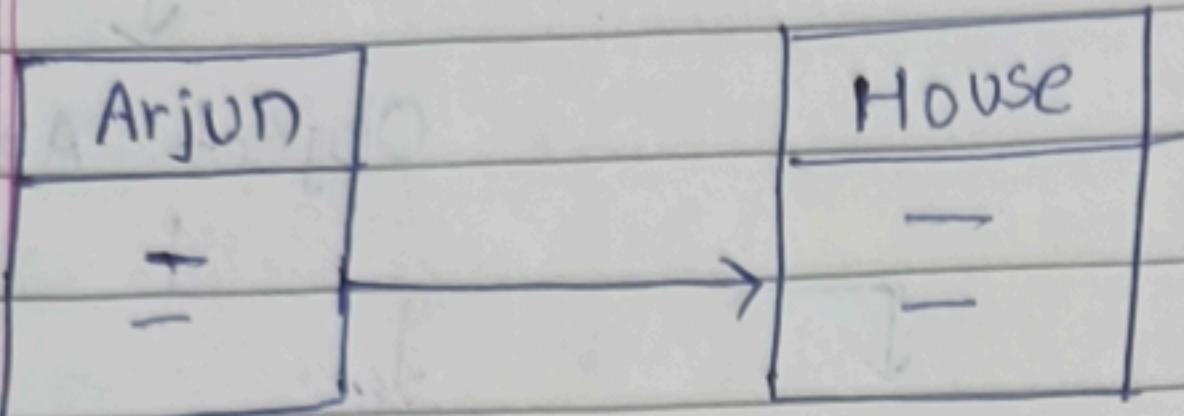
Cow

most used in LLD more than inheritance

Page No.	
Date	

- Composition
↳ (has-a) relation

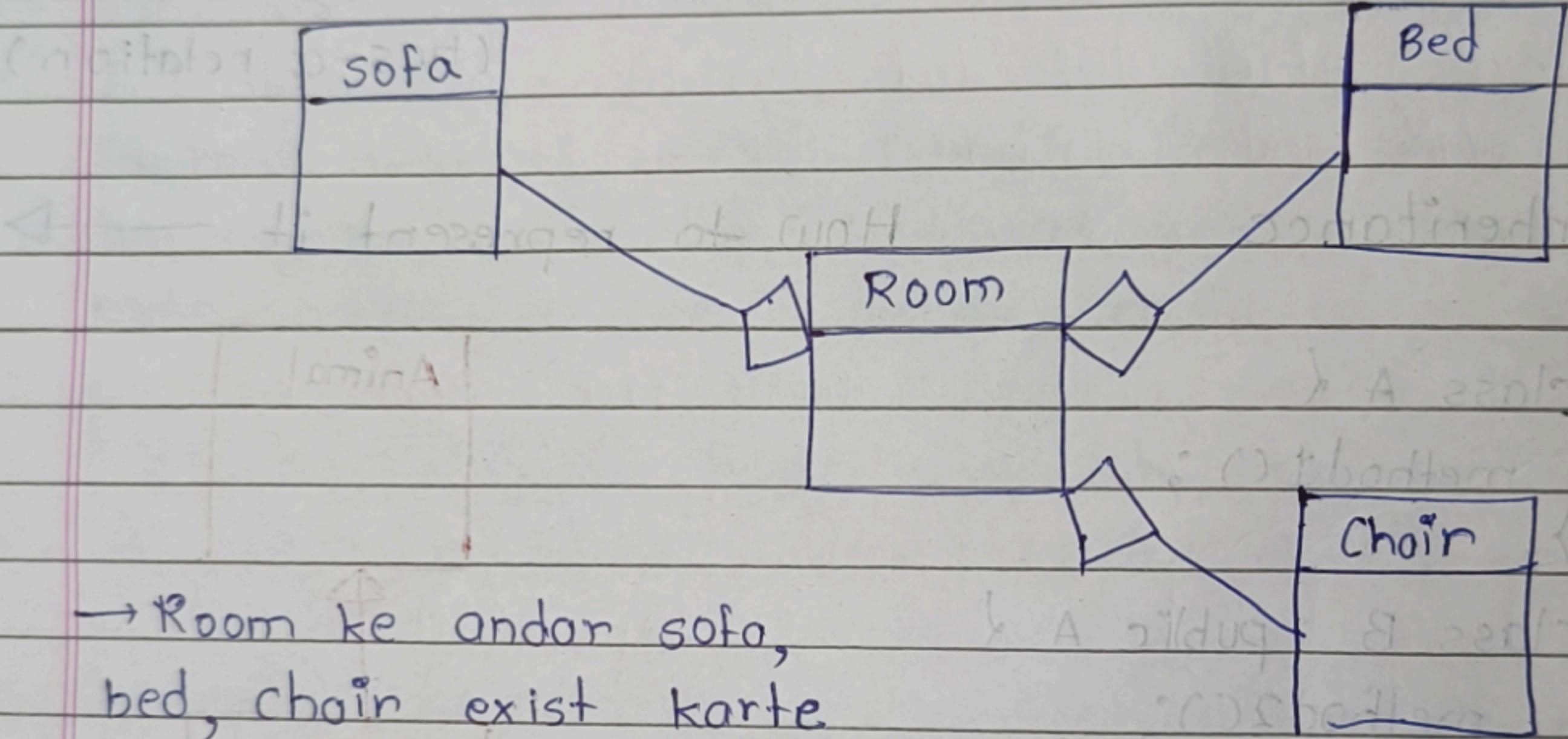
1) Simple Association - weakest/simple form of communication



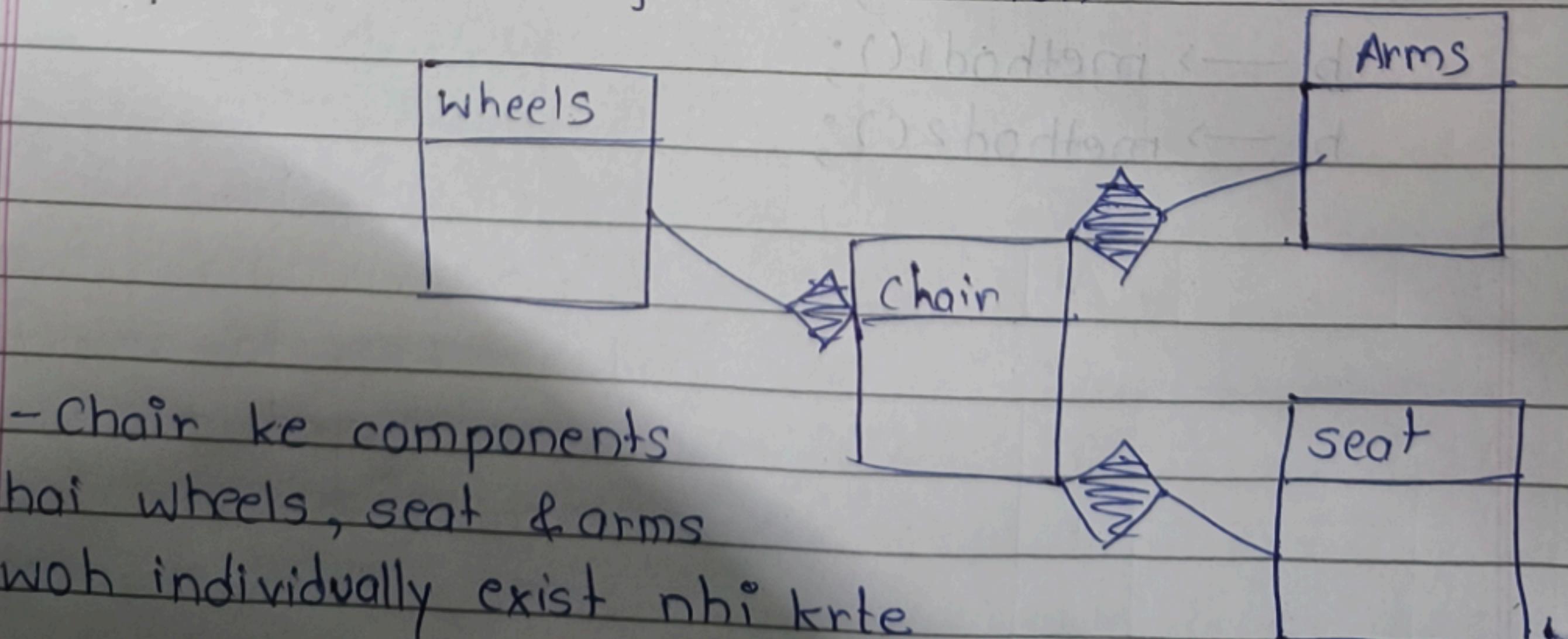
connection

Arjun lives in a House
Arjun Has House

- 2) Aggregation - more interlinked



- 3) Composition - strongest connection



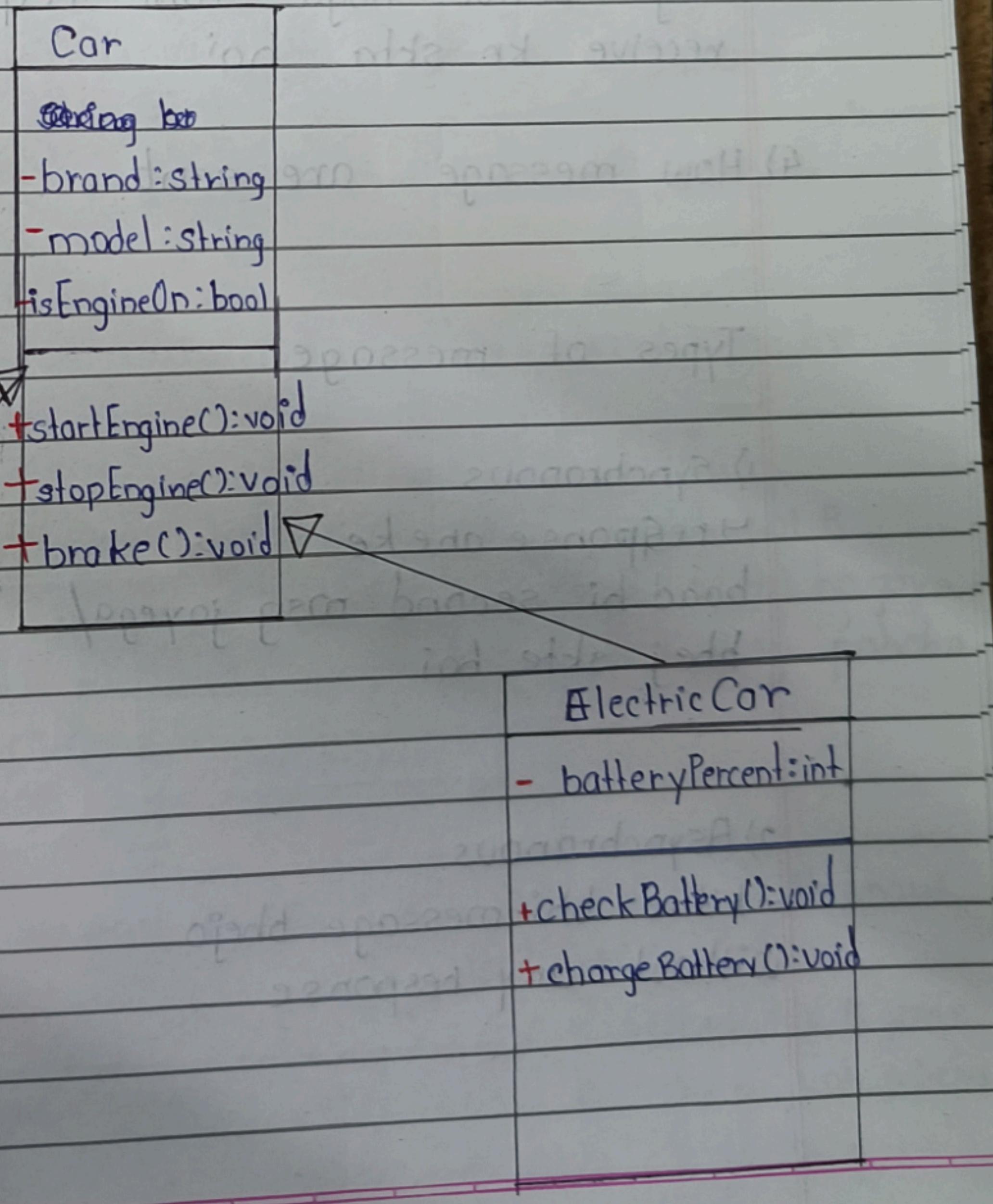
How to represent it in code?

```
class A {
    method1();
}
```

```
class B {
    A * a;
    B() { a = new A(); }
    method2();
}
```

```
main() {
    B * b = new B();
    b->method2();
    b->a->method1();
}
```

Homework : Draw UML diagram of Car - ManualCar & ElectricCar

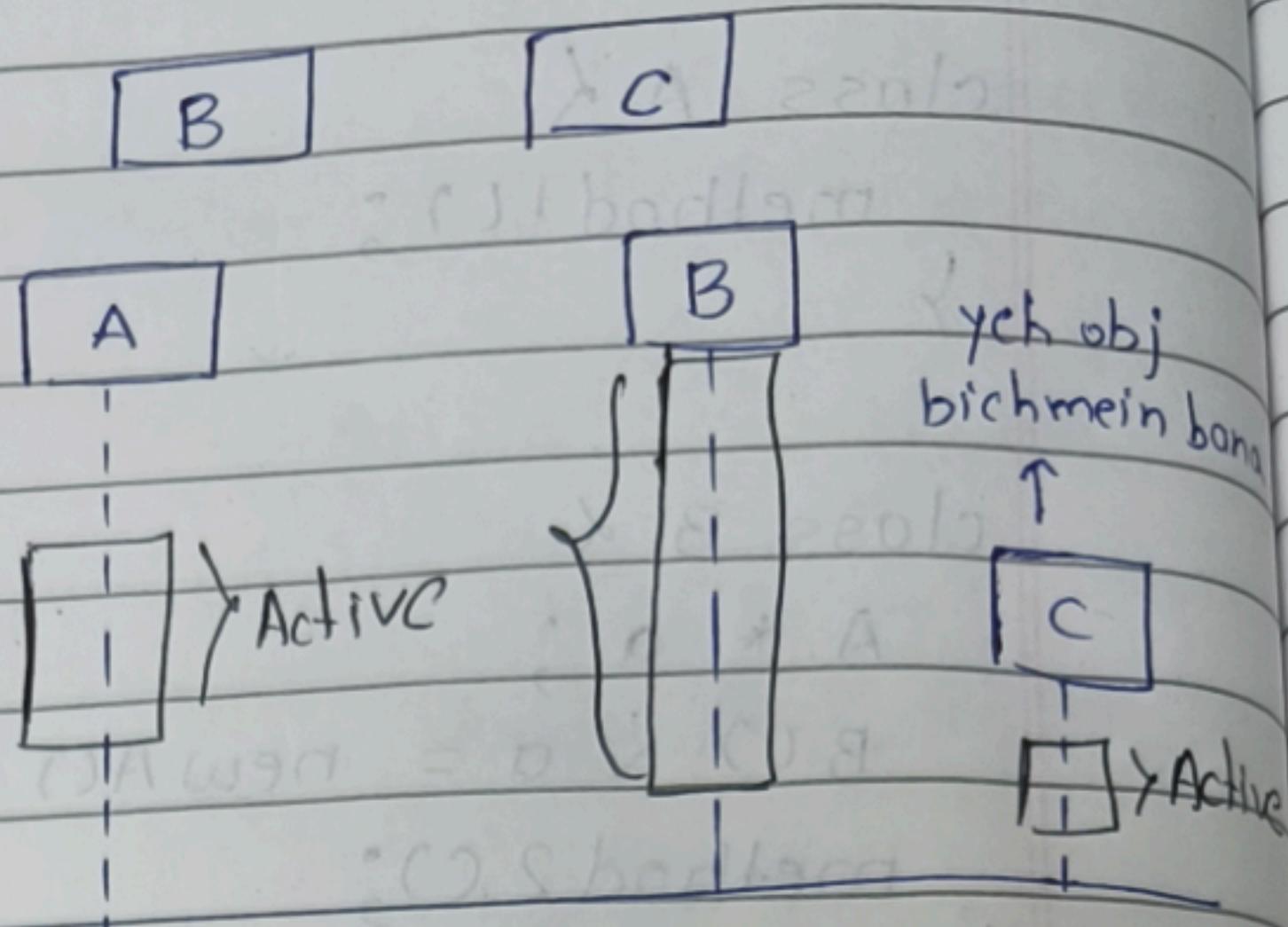


Sequence Diagram

1) Object declare: A B C

2) Lifetime of object:

↳ shows woh object
kab kab exist krega
humare object mein



3) Activation Bar - Object kab tak active hai

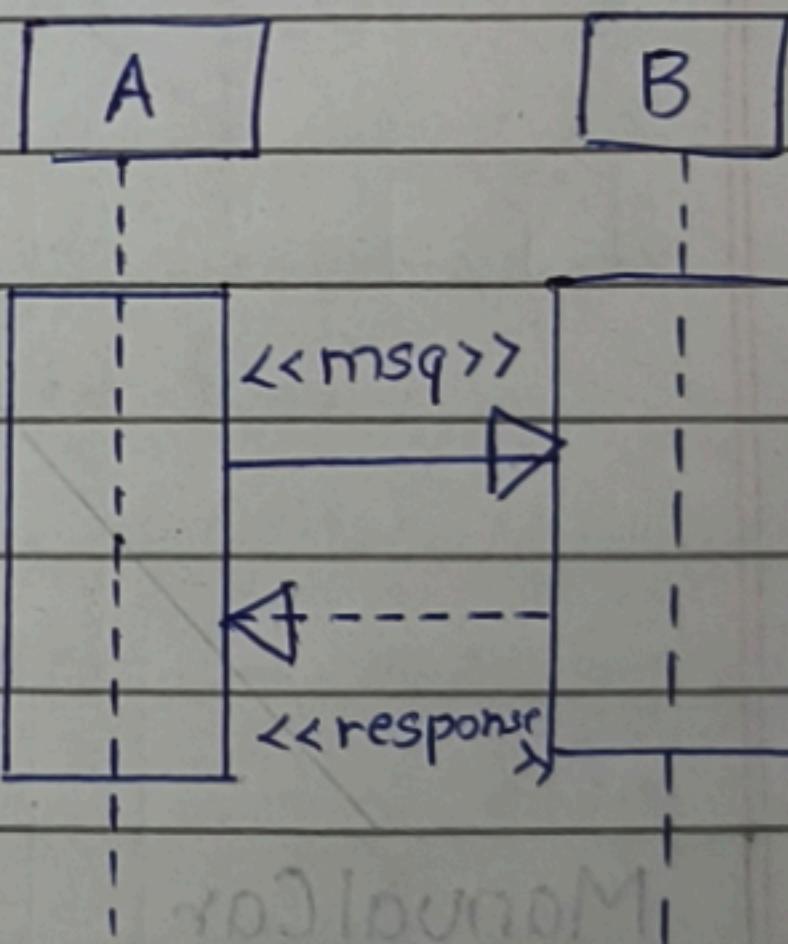
↳ jb active hoga tbhi request bhej skta hai aur
receive kr skta hai

4) How message are sent

Types of message

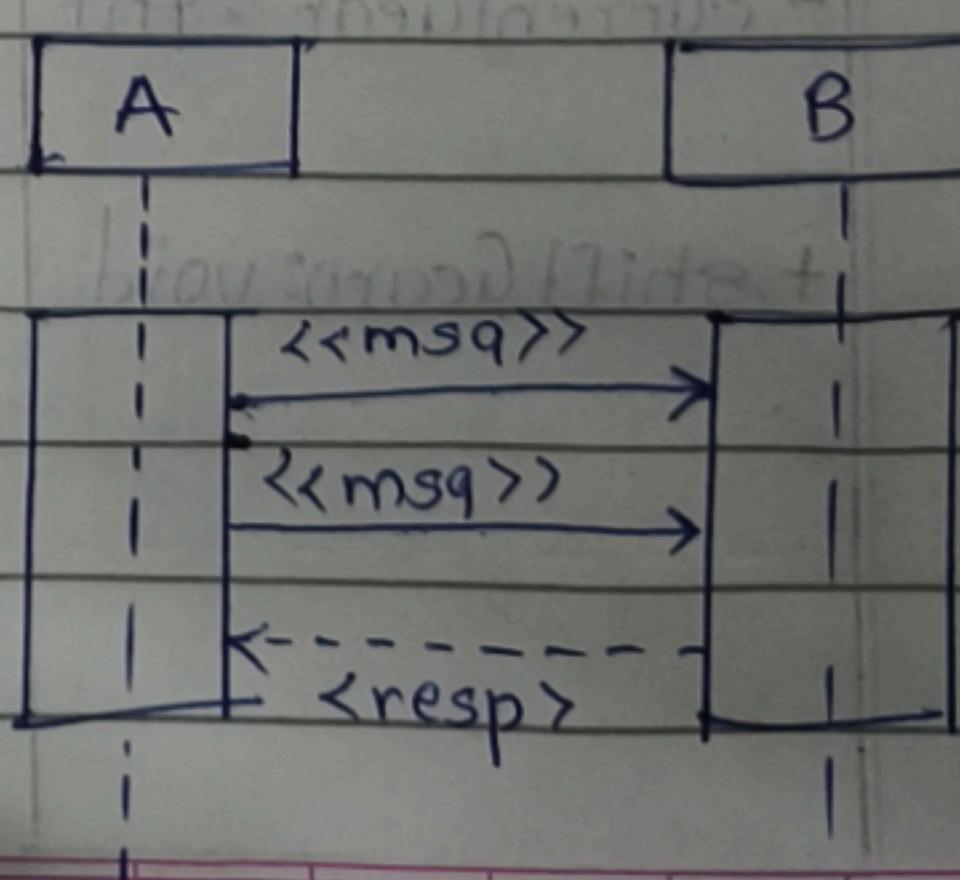
1) Synchronous

↳ response ake baad hi second msg jayega/
bhej skte hain



2) Asynchronous

- kitne bhi message bhejo
without any response



3) Create

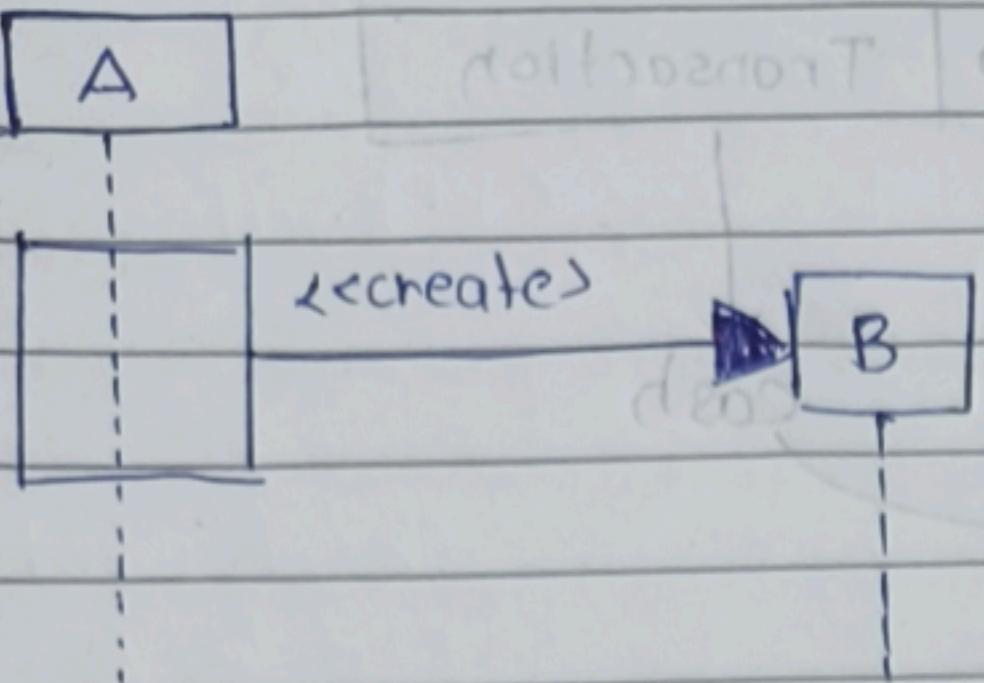
A

4) Dest

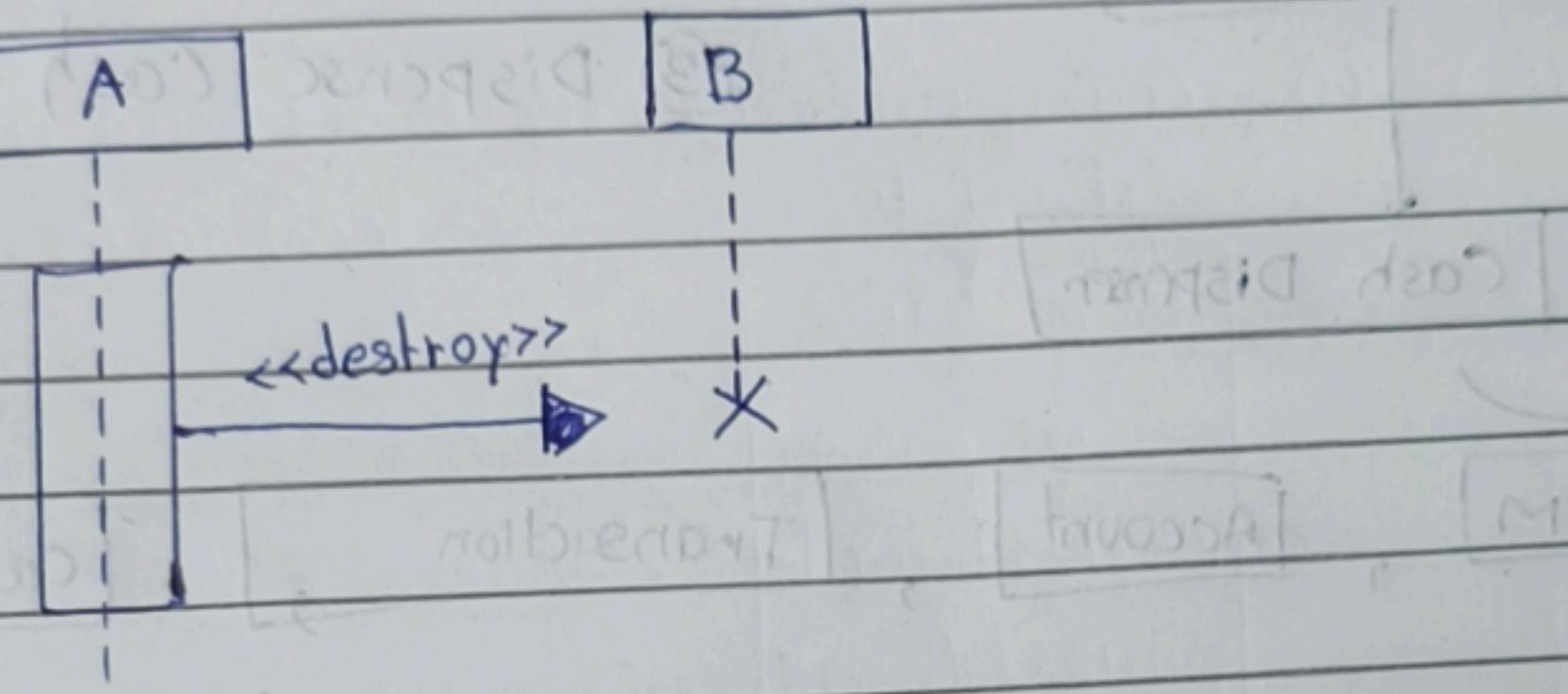
5) Es

6) F

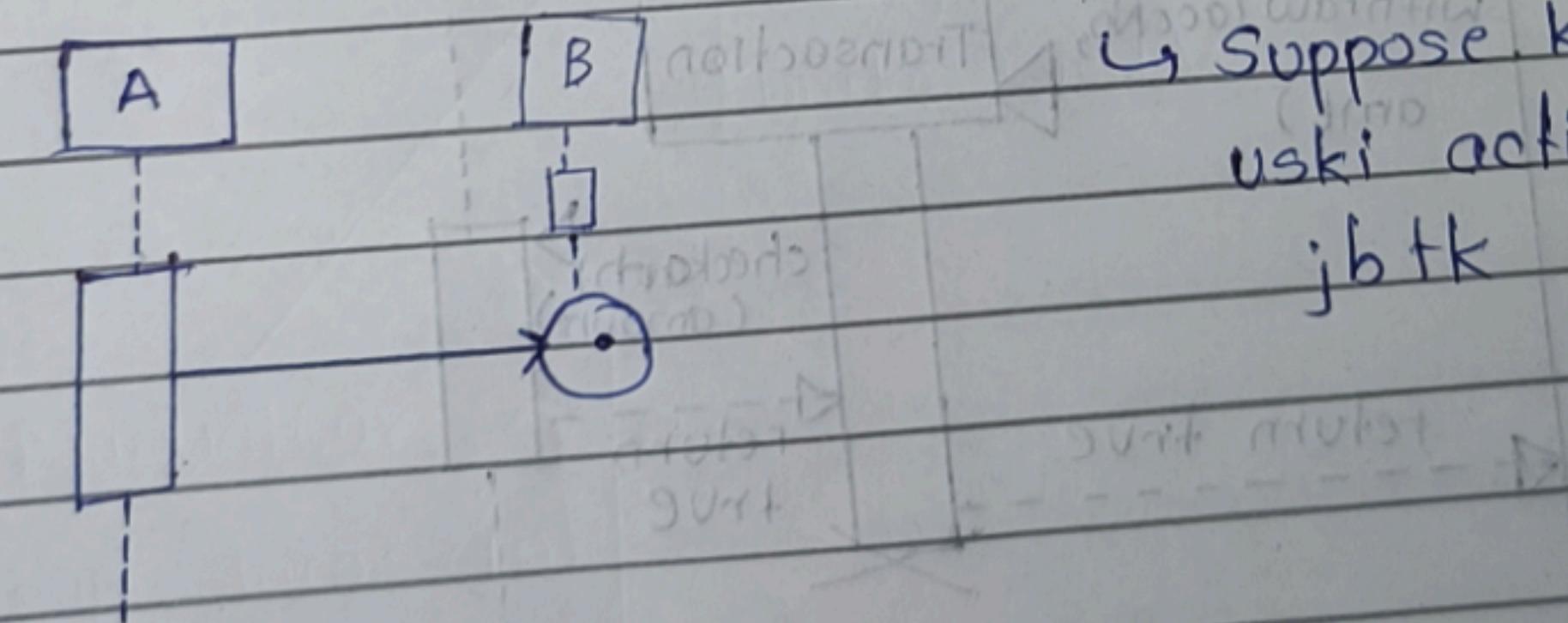
3) Create Message : Creating an object



4) Destroy Message



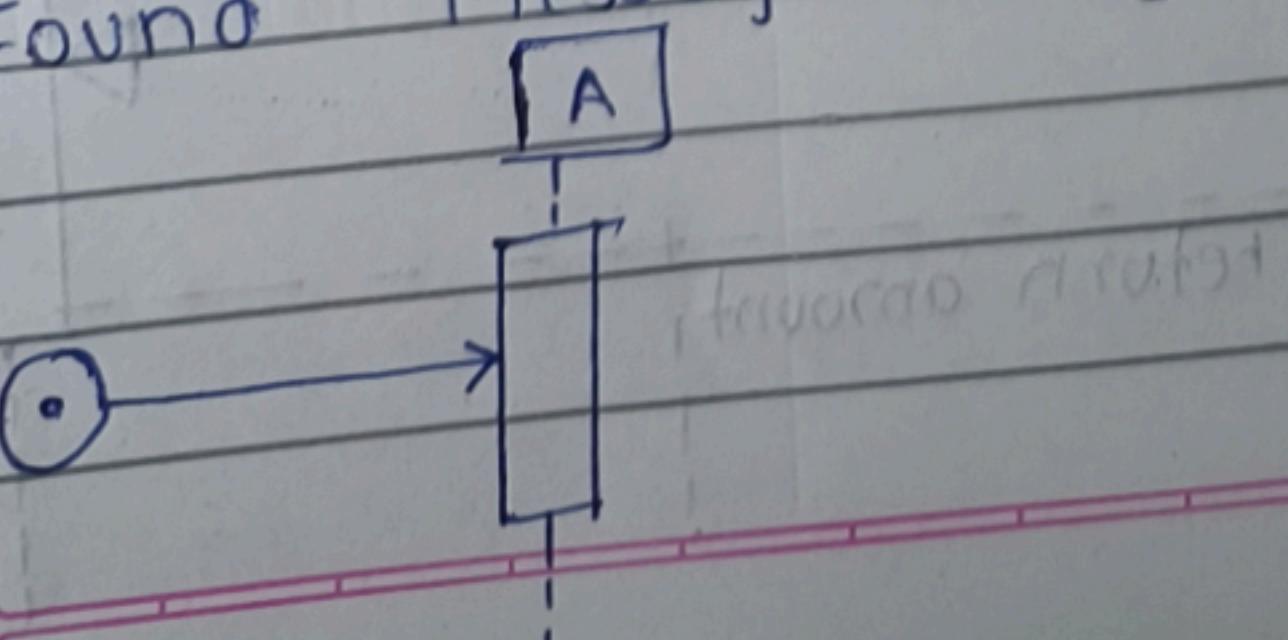
5) Lost Message - Message sent but not received by another object



kab jaa skta hai?

↪ Suppose koi object hai B
uski activation khtm ho gaya
jbt k A ka msg pohcha

6) Found Message - Msg received but who sent don't know



kab jaa skta hai?

↪ Suppose koi object hai B jisne
msg bheja pr jbt k woh pohcha
uska activation khtm ho gya