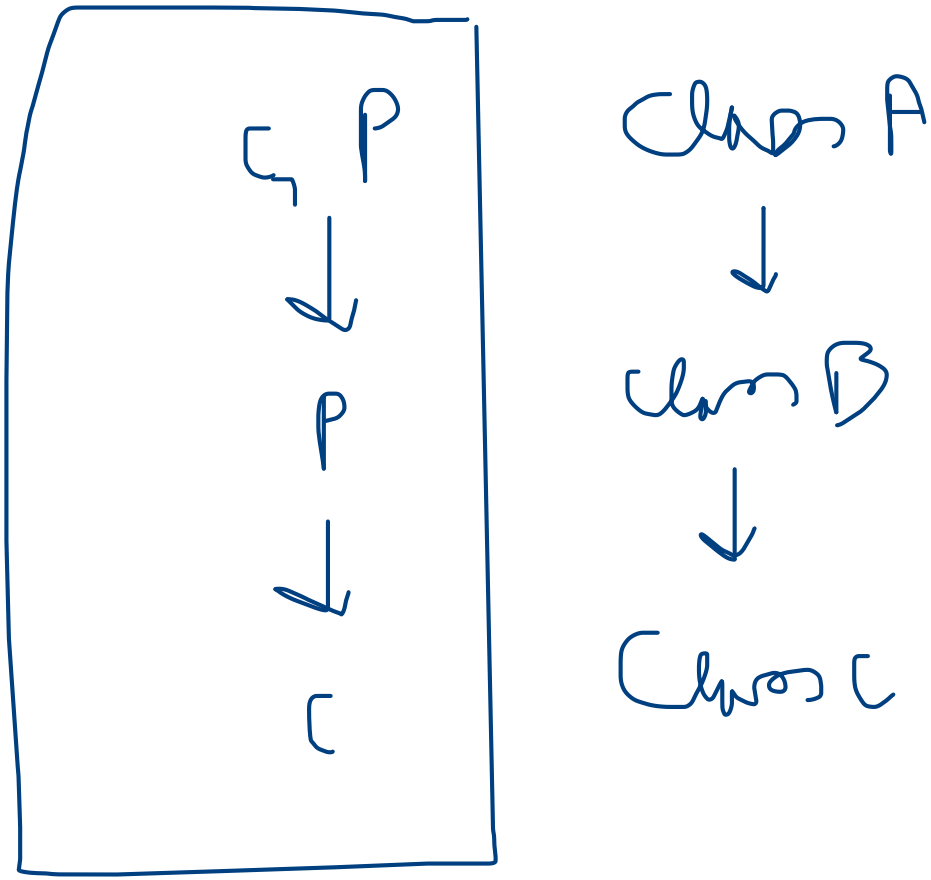
Pillars of OOPS (Rules)

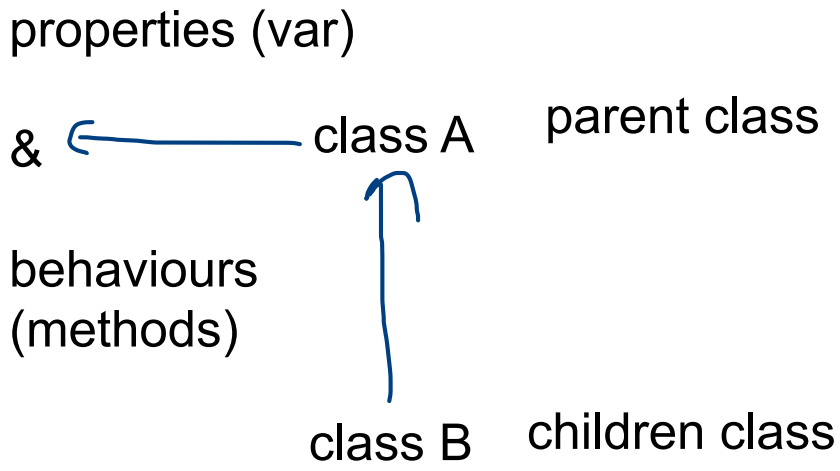
1. Encapsulation
2. Inheritance
3. Polymorphism
4. Abstraction

Advance of Abstraction --> interface

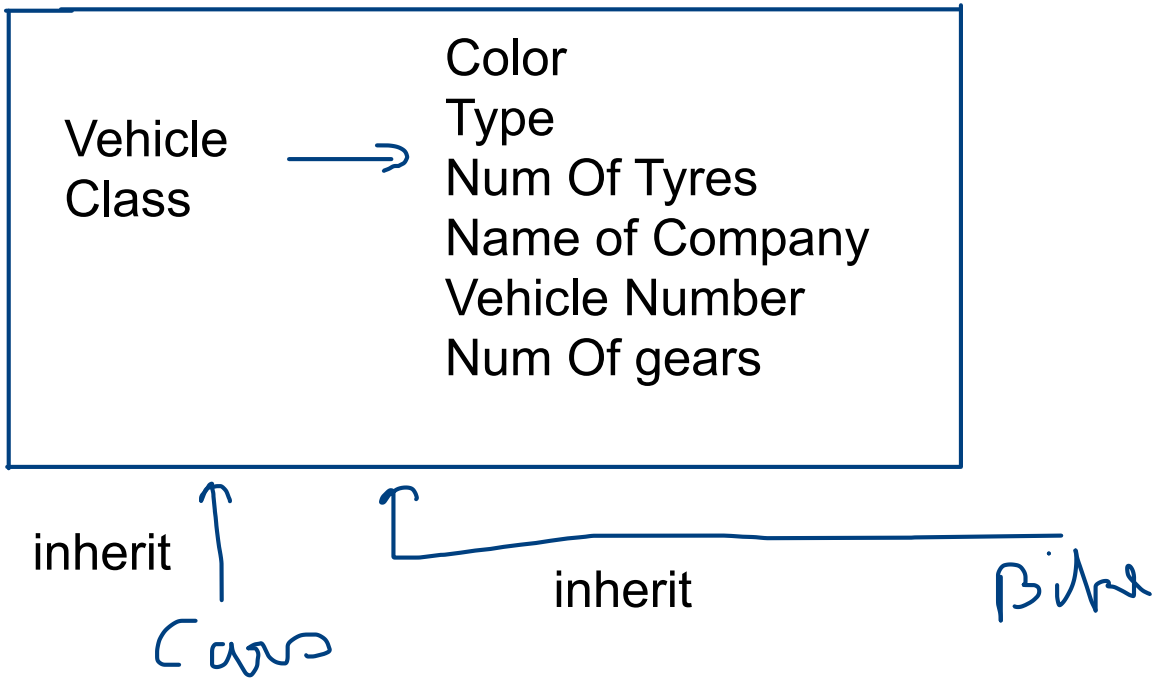
Inheritance



inheritance is a mechanism in which one object aquires all properties and behaviour of its parent object



IS-A relationship



gear system --> Automatic / Manual

General Terms

Class --> blueprint

Sub-Class / Child Class --> class inherit the other class

Super Class / Parent Class / base class --> class from where subclass inherit

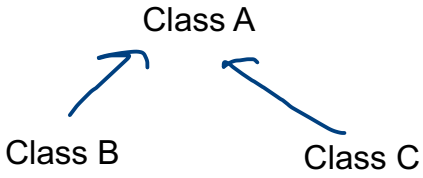
Types of Inheritance

1

Single Inheritance

3

Hierarchical Inheritance

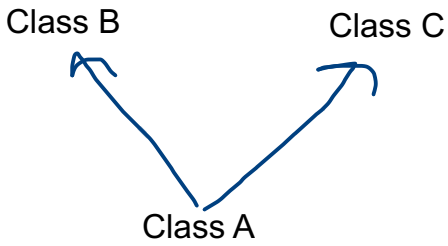


2

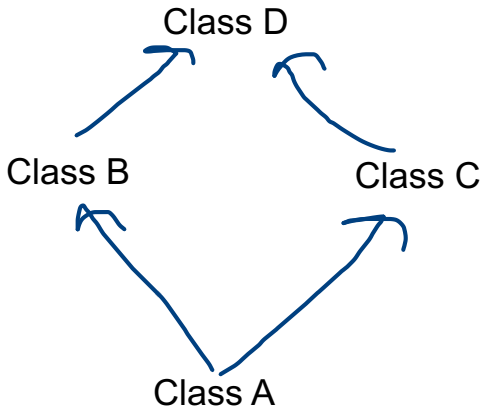
Multilevel Inheritance

4

Multiple Inheritance --> not supported by java

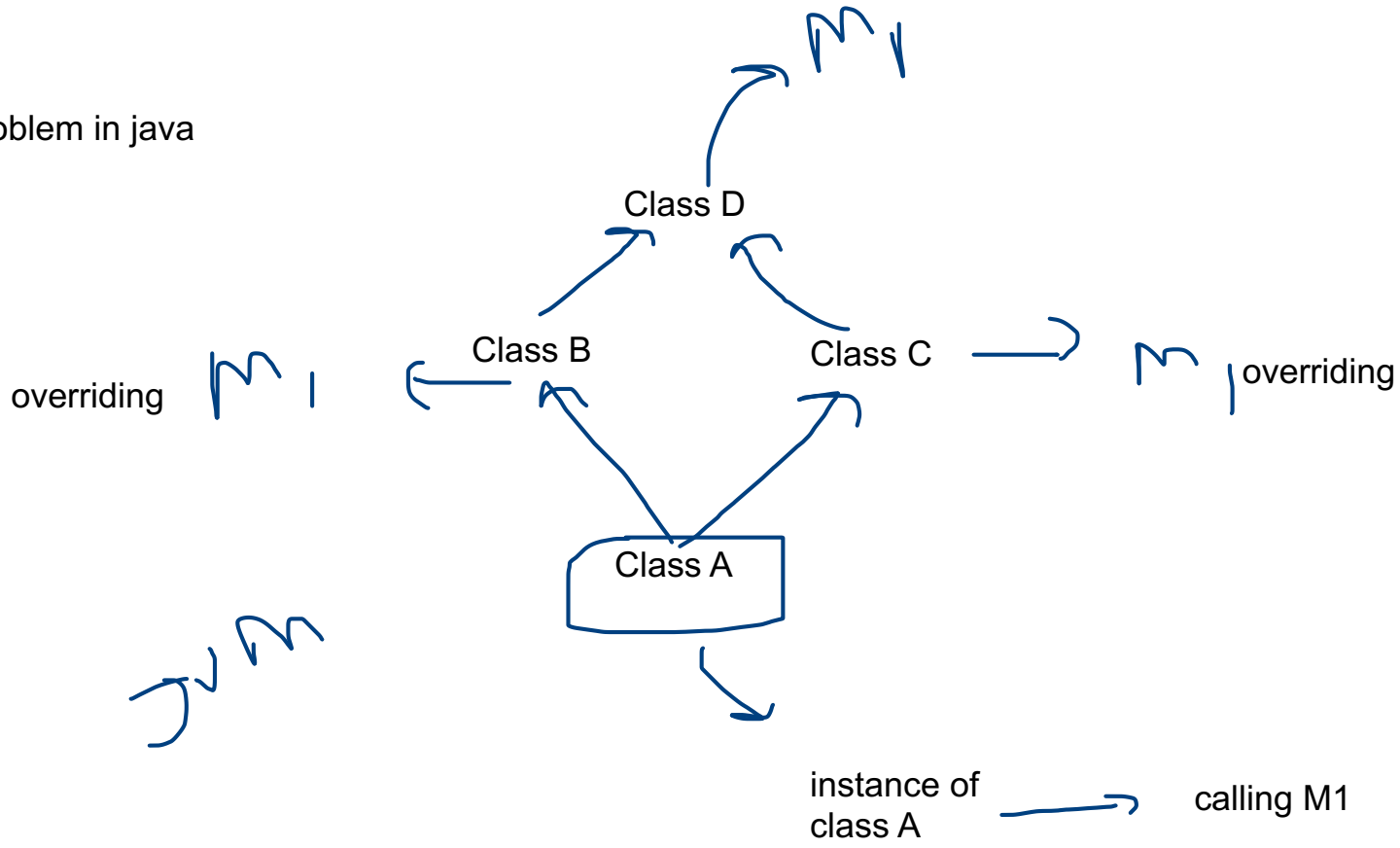


5 Hybrid Inheritance

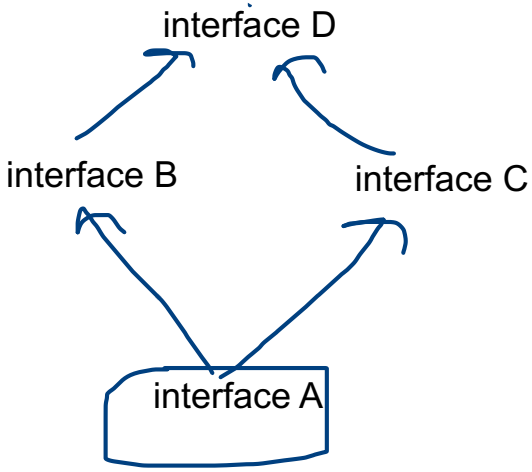


why multiple inheritacne is not allowed in java ?

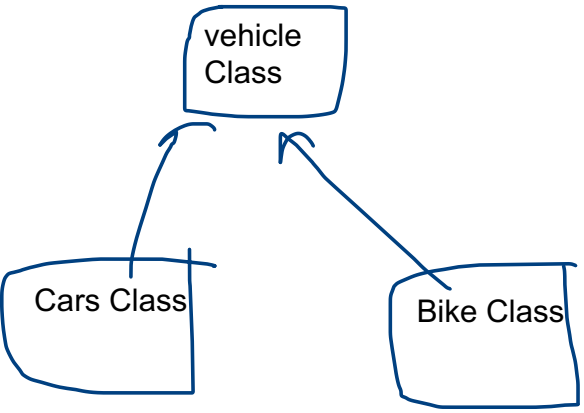
Diamond Problem in java

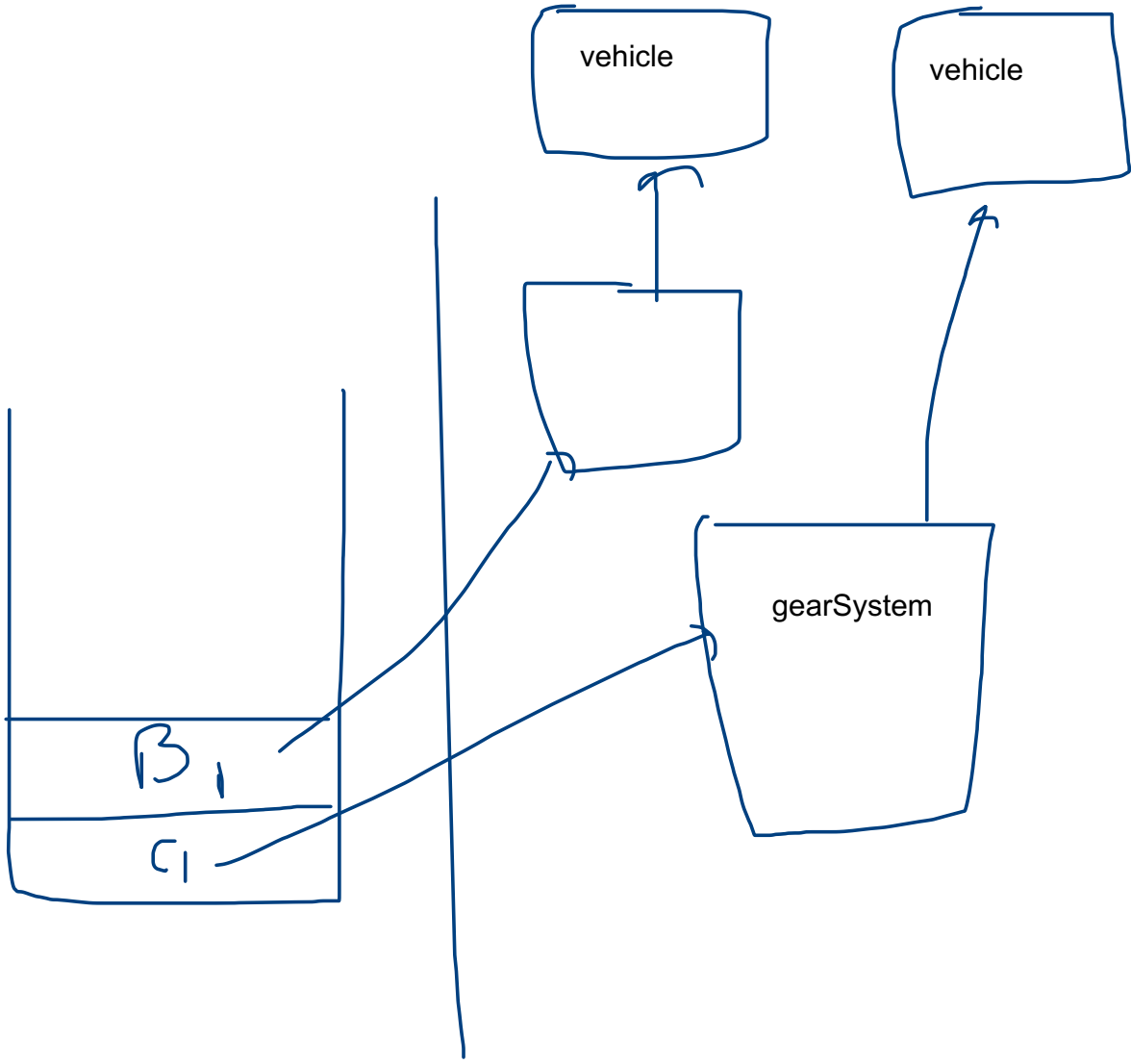


allowed in java --> multiple inheritance of interfaces

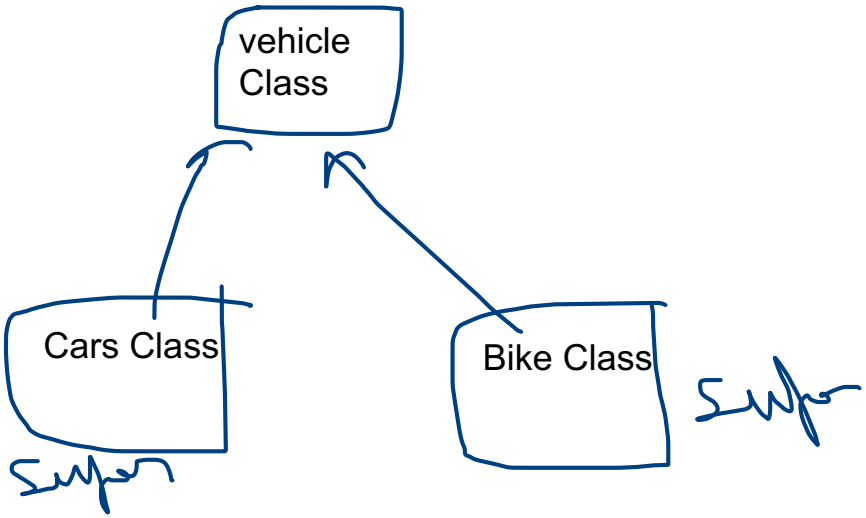


Memory Diagrams of inheritance





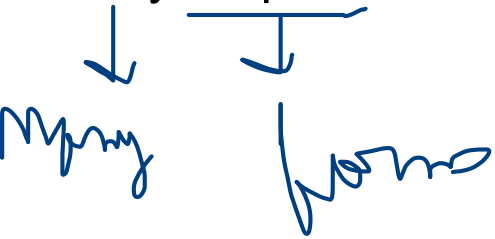
this and super keyword



this --> represents current class

super --> whatever class it is used on super refer to its immediate parent class

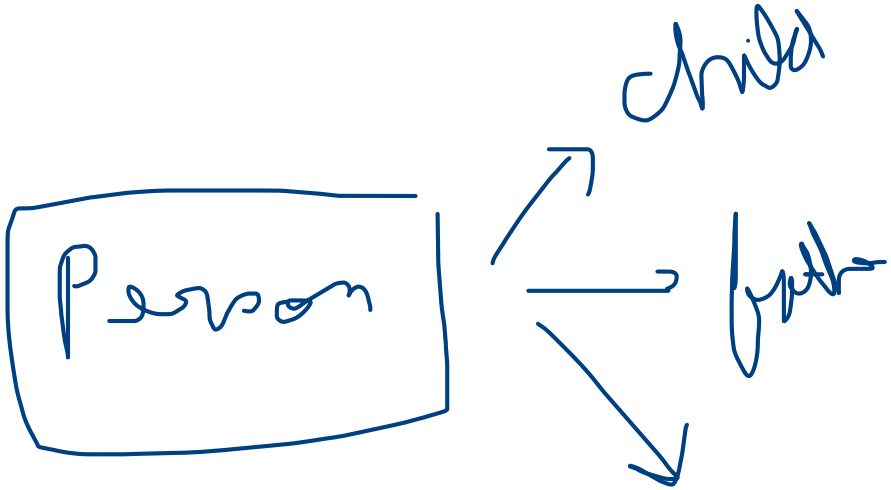
PolyMorphism same name different meaning

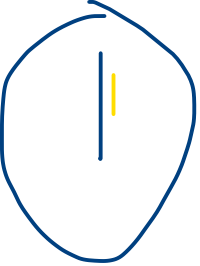


Types of Polymorphism

1. Static Polymorphism / compile time Polymorphism

2. Dynamic Polymorphism / Run-Time Polymorphism


Person



Static Polymorphism / compile time Polymorphism

those Polymorphism in which

1. decision is happen during compile time

2. it is achieved by Method overloading or operator overloading

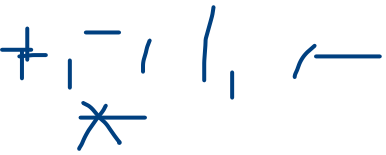
Method Overloading --> is happens in same class

int sum (int a, int b) --> a + b

float sum(flaot a, float b) --> a + b

int sum(int a, int b, int c) --> a + b + c

same name diff mean



Operator Overloading --> java do not support it

Strings java support it

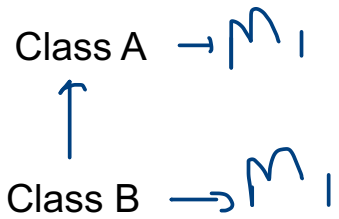
```
String str = "abcd" + "efgh";
System.out.println(str);
```

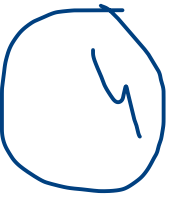
Runtime Polymorphism / Dynamic Polymorphism / Dynamic Method Dispatch Polymorphism

1. Its decision happen at runtime only

2. it is achieved using Method Overriding

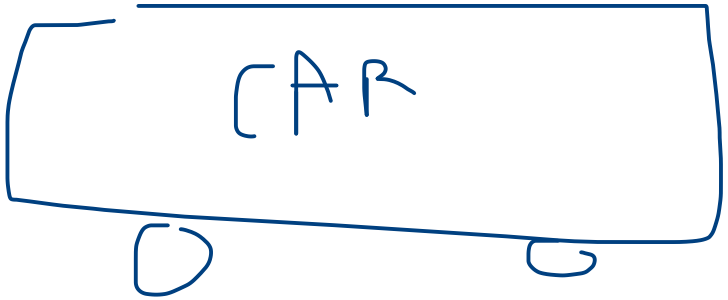
Method Overriding --> It happens in different class





Abstraction

it is a process of hiding the implementation details and showing only functionality to user



How to achieve abstraction in java?

1. Abstract classes (0% to 100%)

2. Interface (100 % only)

abstract class --> using abstract keyword

1. abstract class can have abstract / non-abstract methods

abstract methods --> Methods which do not have body only have defination

2. You cannot create the instance of abstract class

3. It can have constructors and static methods

4. It is always mandatory to implement all of the internal details of abstract class inside of its child class

Manager --> handle devs

Manager --> write Cars class

dev1 --> implement Maruti class
dev2 --> implement Tata class
dev3 --> implement Honda class

Interfaces

1. Achieve 100 % abstraction

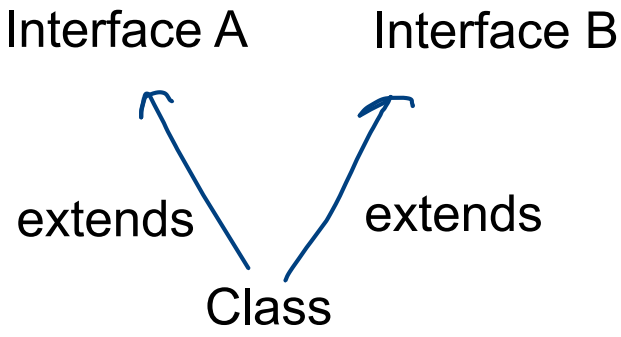
2. It cannot contian non abstract methods and instance variables, only abstract methods

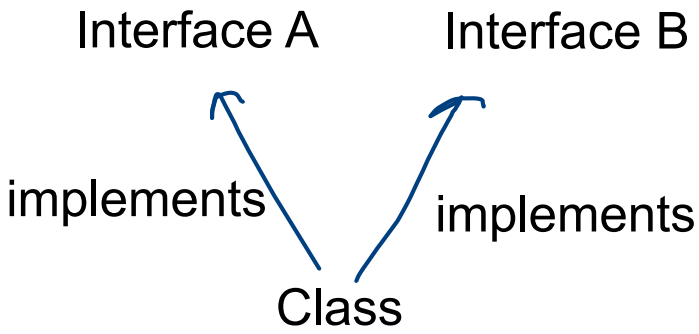
3. Instance making not allowed

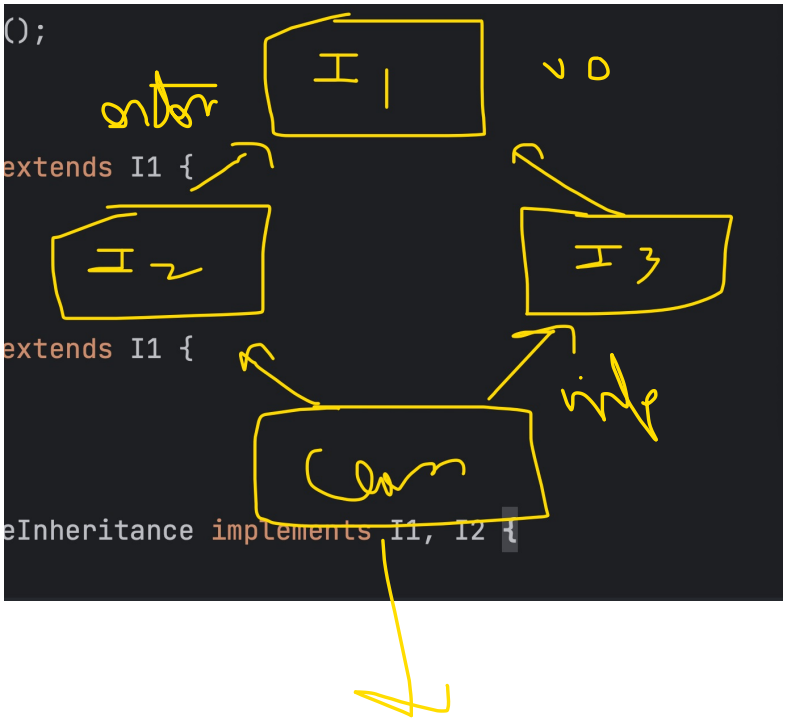
4. Achieve Multiple inheritance

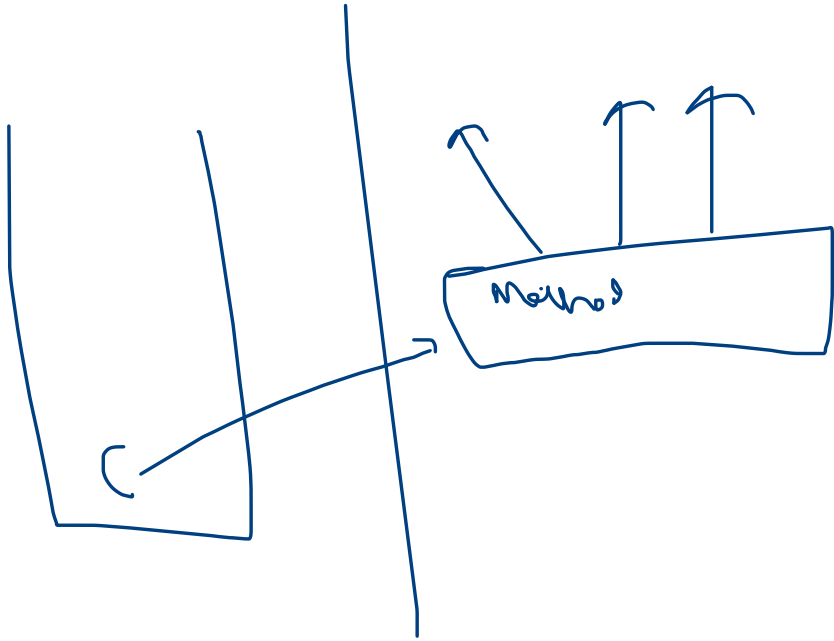
5. It is always mandatory to implement all of the internal details of all the methods of an interface inside its child class

Multiple Inheritance









need to define all methods defination in this class only