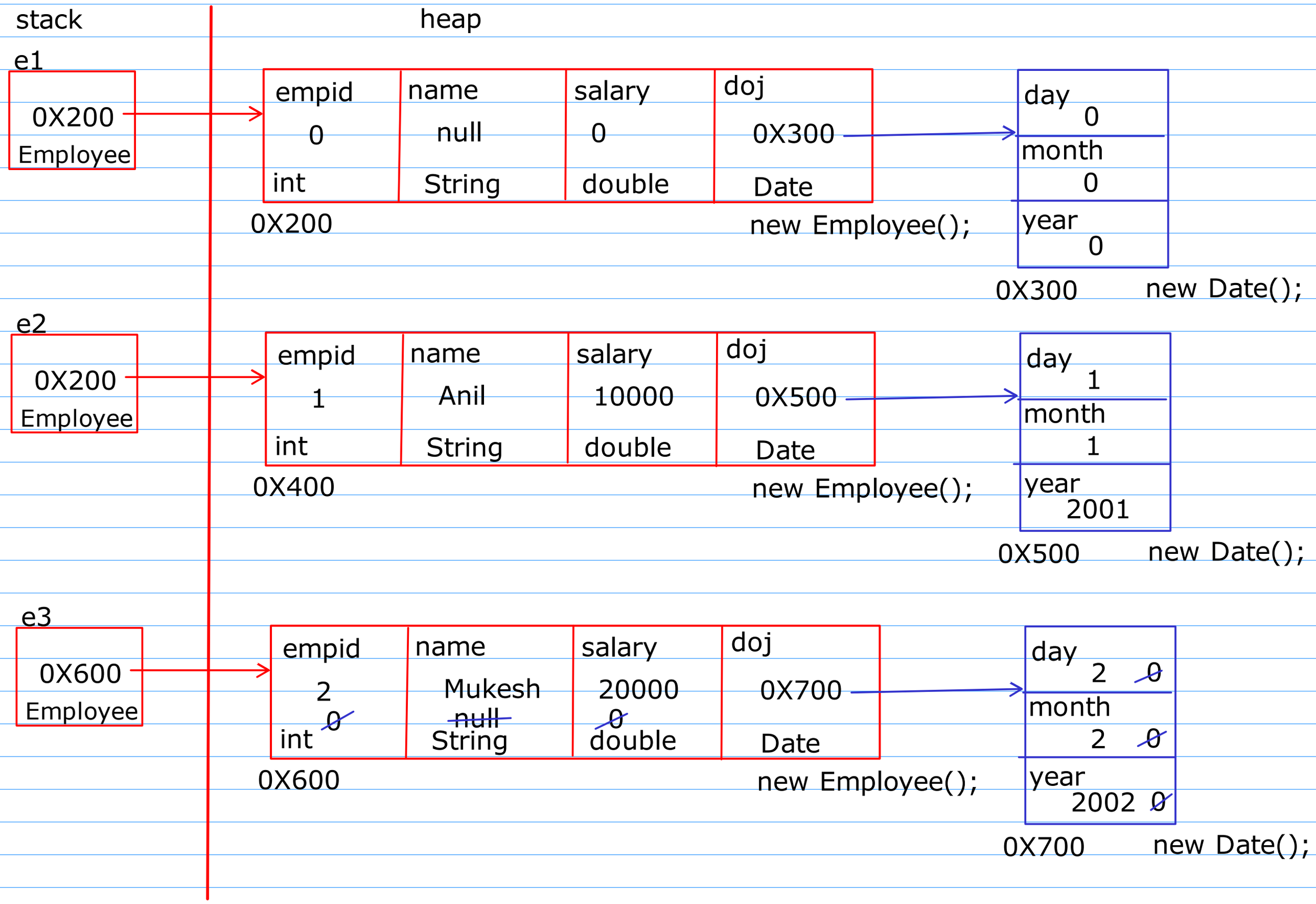
Hirerachy
- Reusability
- has-a(Association), is-a(inheritance)

Association
- Composition
- Aggegration

Human(Dependent) has-a Heart(Dependency)
Employee(dependent) has-a DateofJoining(dependency)



stack

heap

e1

0X200

empid	name	salary	veh	doj
0	null	0	null	0X300
int	String	double	Vehicle	Date

0X200

day 0
month 0
year 0

0X300

e2

0X400

empid	name	salary	veh	doj
1	Anil	10000	~~null~~ 0X600	0X500
int	String	double	Vehicle	Date

0X400

day 1
month 1
year 2001

0X0500

veh_name
bmw
veh_number
mh101234

0X600

e3

0X700

empid	name	salary	veh	doj
2	Mukesh	20000	null	0X800
int	String	double	Vehicle	Date

0X700

day 2
month 2
year 2002

0X800

Composition

Aggegration

Day07 - Demo01/src/com/sunbeam/p2/Program01.java - Spring Tool Suite 4

File Edit Source Refactor Source Navigate Search Project Run Window Help

Package Explorer

- Demo01 [cj-o-16 main]
  - JRE System Library [JavaSE-1.8]
  - src
    - com.sunbeam.p1
      - Date.java
      - Employee.java
      - Program01.java
    - com.sunbeam.p2
      - Date.java
      - Employee.java
      - Program01.java
      - Vehicle.java

Vehicle.java | Employee.java | Program01.java

```
1  package com.sunbeam.p2;
2
3  import java.util.Scanner;
4
5  public class Program01 {
6
7      public static void main(Str
8          Scanner sc = new Scanne
9
10         Employee e1 = new Emplo
11         e1.displayEmployee();
12
13         Employee e2 = new Emplo
14         e2.addVehicle(sc);
15         e2.displayEmployee();
16
17 //      Employee e3 = new Emplo
18 //      e3.acceptEmployee(sc);
19 //      e3.displayEmployee();
20     }
21
```

Problems | Javadoc | Declaration | Console

<terminated> Program01 (1) [Java Application] D:\Softwares\sts-4.19.0\plugins\org.eclipse.justj.openjdk.hotspot

```
Empid - 0
Name - null
Salary - 0.0
Date of Joining - 0/0/0
---------------------------
Enter the vehicle name - bmw
Enter the vehicle number - mh1212
Empid - 1
Name - Anil
Salary - 10000.0
Date of Joining - 1/1/2001
Vehicle name - bmw
Vehicle number - mh121234
---------------------------
```

ENG IN 10:09 AM 26-07-2025

Employee has-a Date

```
class Date{

}
```

System has-a PrintStream

```
System.out
```

```
class Employee{
int id
Date doj;
Date dob;
Date dot;
Vehicle bike;
Vehicle car;
}
```

OOA

```
class A{

}
```

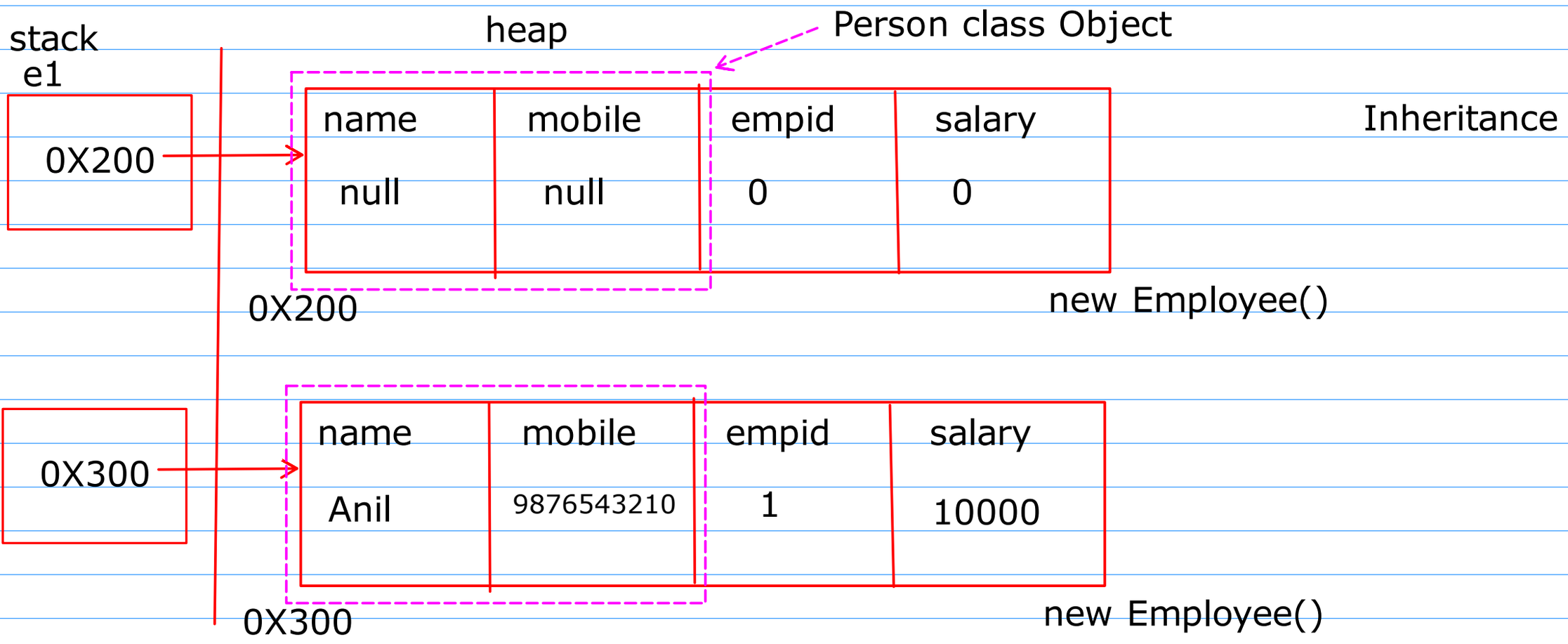
```
class B{

}
```

Inheritance

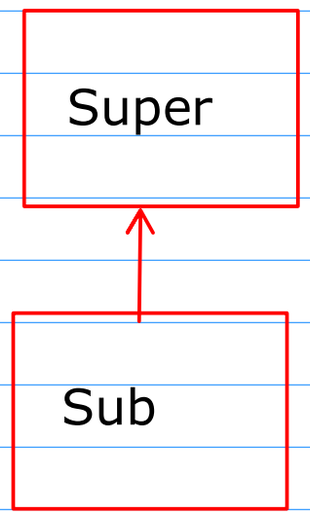
- Their is-a relationship between 2 entities
- eg Employee(Child) is-a Person(Parent)

Apple(Subclass) is-a Fruit(Super)
Mobile is-a Device

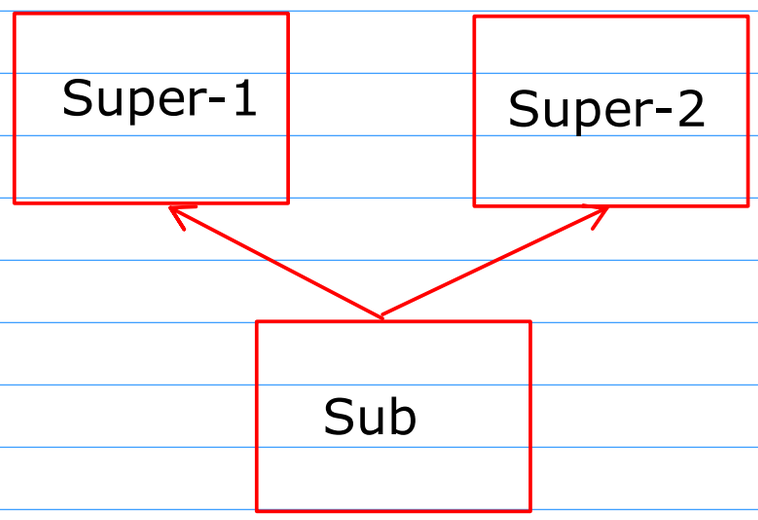


// Complete the diagram even for the Student

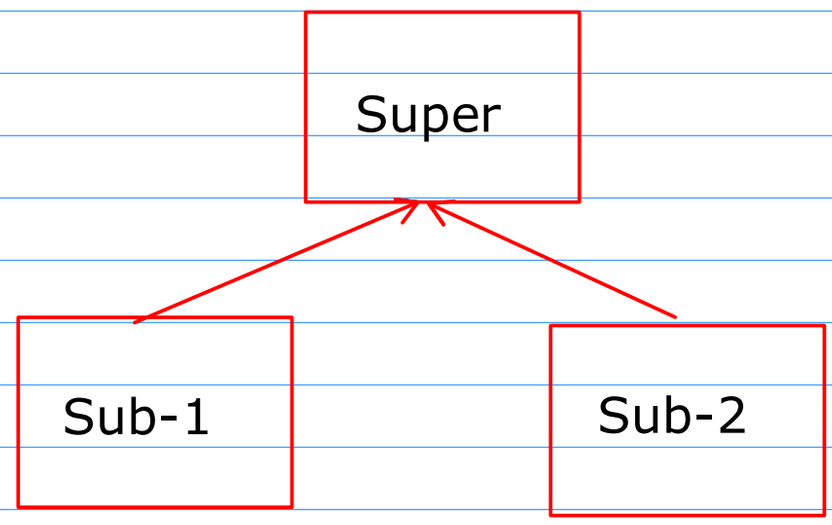
Types of inheritance



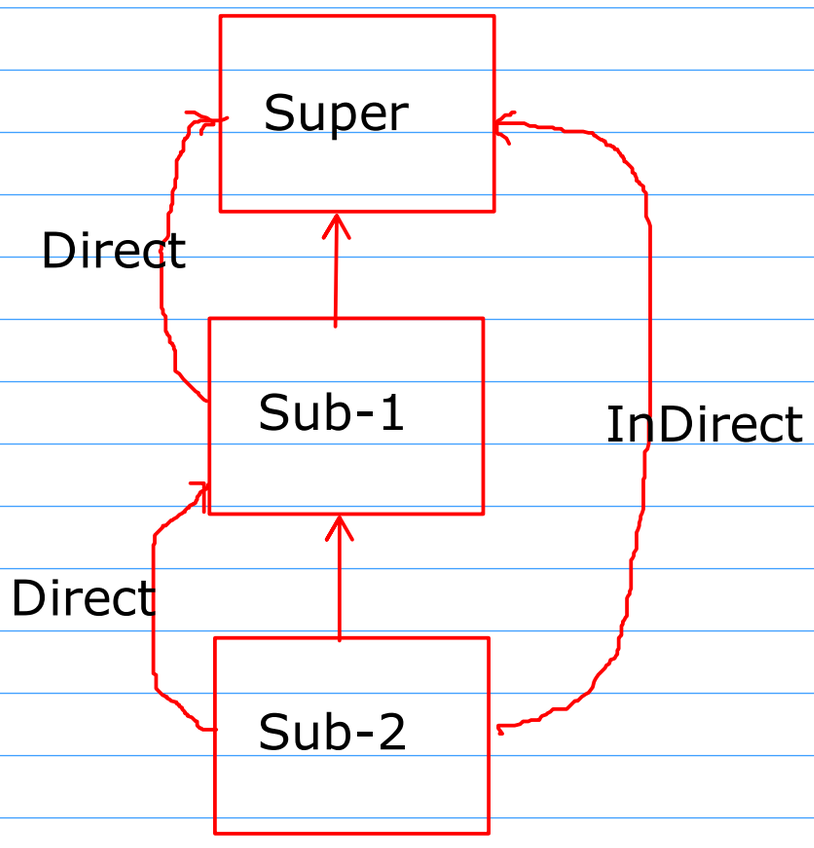
1. Single Inheritance



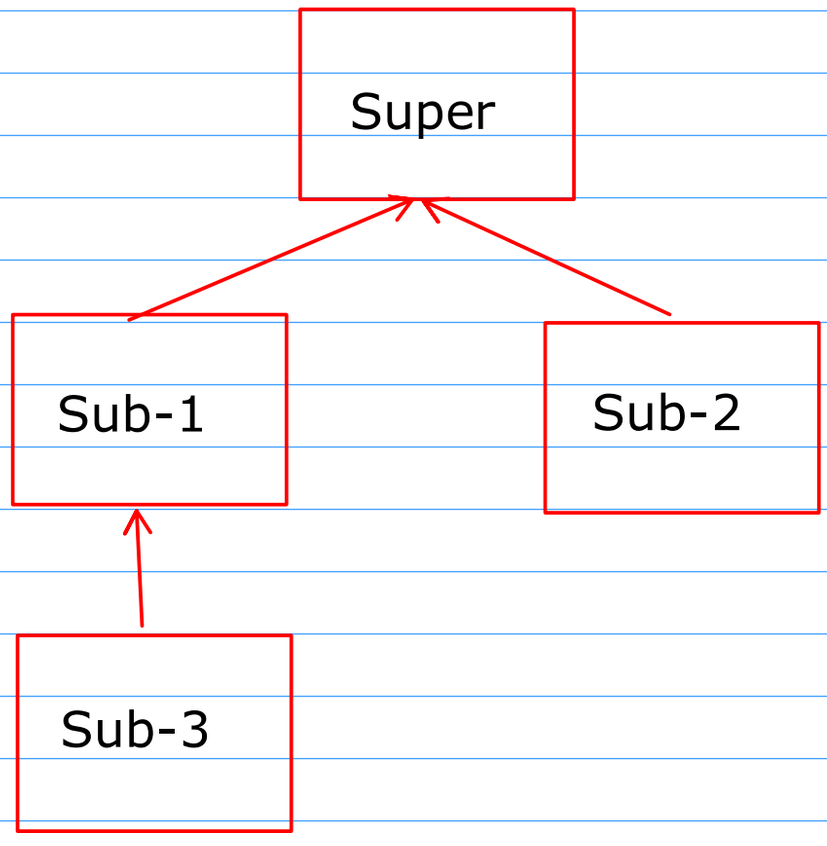
2. Multiple Inheritance



3. Hirerachical Inheritance



4. Multilevel Inheritance



5. Hybrid Inheritance - Mix of any 2 inheritances

- Java Supports all the types of inheritances.
- Java does not support multiple class inheritance however it do supports multiple interface inheritance

Method Overriding

- Defining the method of the super class once agin with same name and signature in sub class is called as method overriding
- Why to perform method overriding ?

1. If the implementation of the super class method is partial complete.
2. If the implementation of the super class method is 100% incomplete.
3. If the implmentation in the subclass we want totally different from the super class method

Employee
manager
salesman