Relationships

is-a relationship:- one class is using a properties and beh
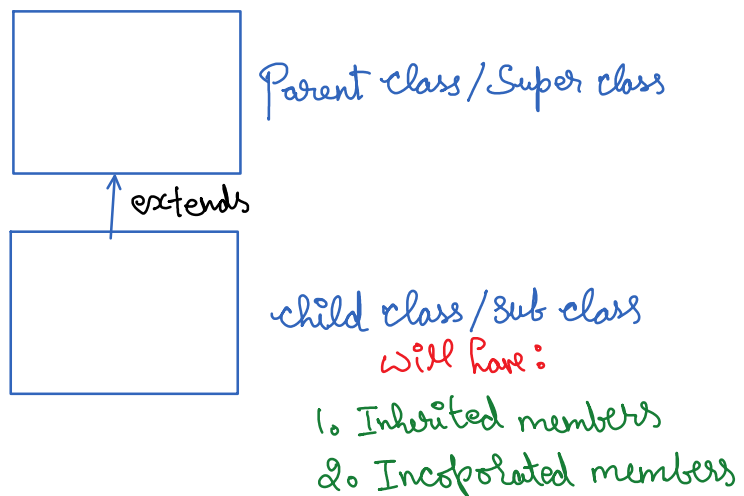of another   class with inheritance
ex:-all the properties of emp class inherited to testing
class
has-a relationship :-one class is using a properties and
behavior of another   class without inheritance
ex:-encapsulated class and driver class

Inheritance
Taking all the properties and behavior of super
class to a sub class by using extends keyword is
called as inheritance

Parent class/Super class

extends

child class/sub class
will have:
1. Inherited members
2. Incorporated members

super class:-class which is providing all the properties and behaviors to a
child class

sub class:-class which is inheriting members from a super class

we are inheriting by using extends keyword

extends

syntax

class sub_class_name extends super_class_name

explanation :- all the properties and behavior of super class is inherited to sub class

NOTE:-if class doesn't extends any class then by default compiler will
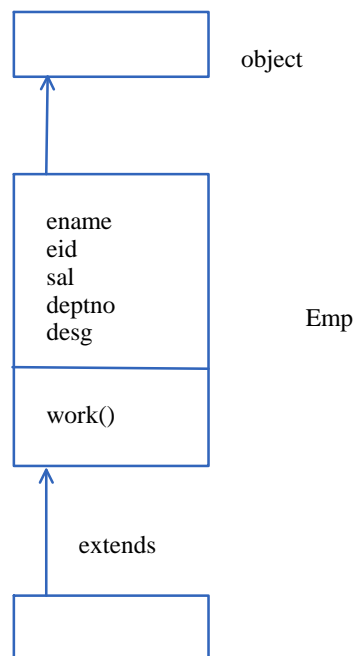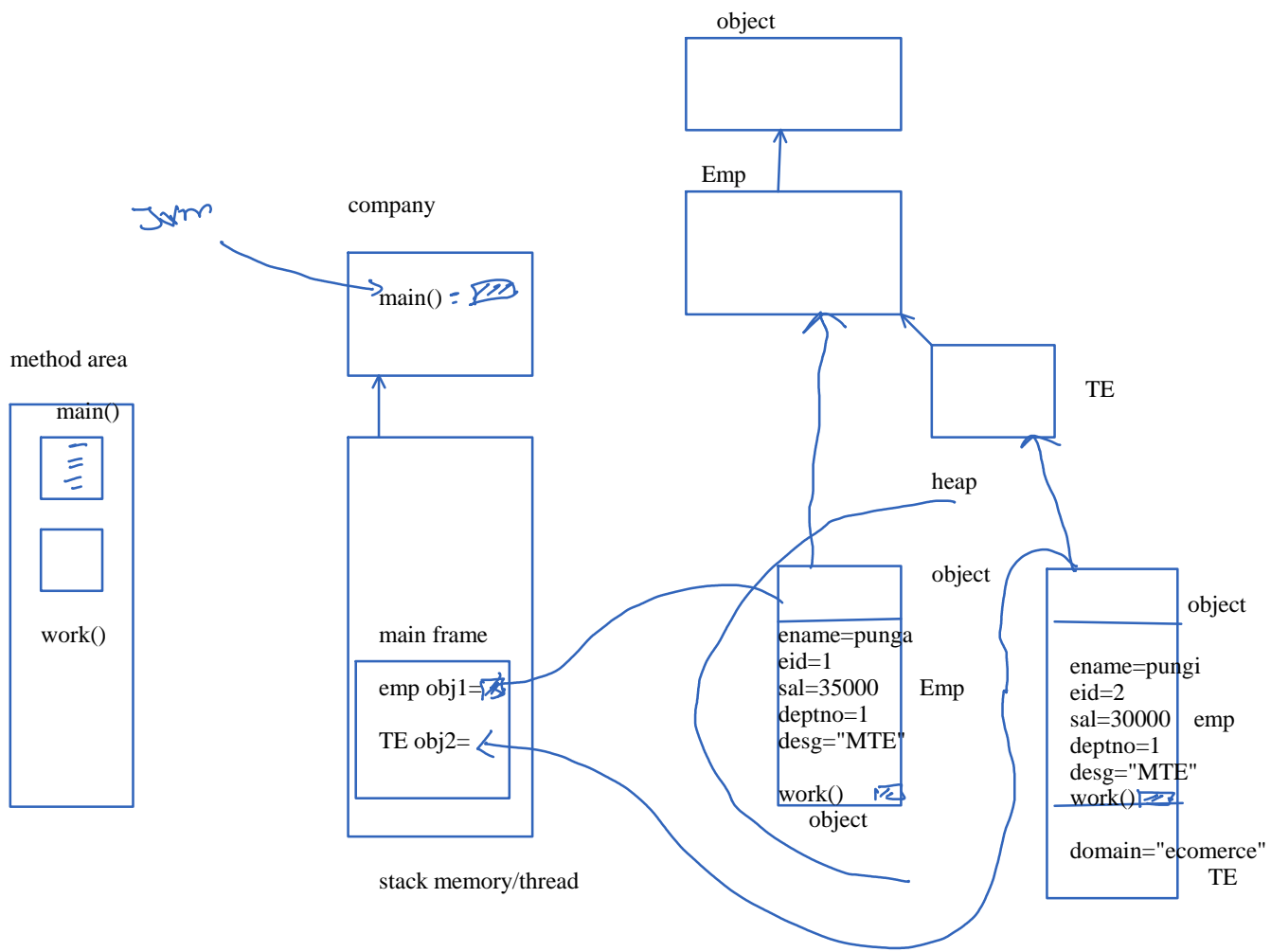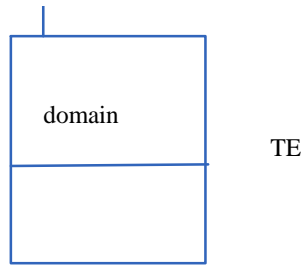**extend Object class**

note
 constructer calling statement
this():-to call the constructer of same class
super():-to call the constructor immediate parent class constructor

   1)one constructor can contain either one this() or one super()
   2)CCS should be declared at the beginning
   3)when there is no CCS then by default compiler will add super()
   4)to use super() extends is mandatory(to use super()-classes should
   follow is-a relationship)
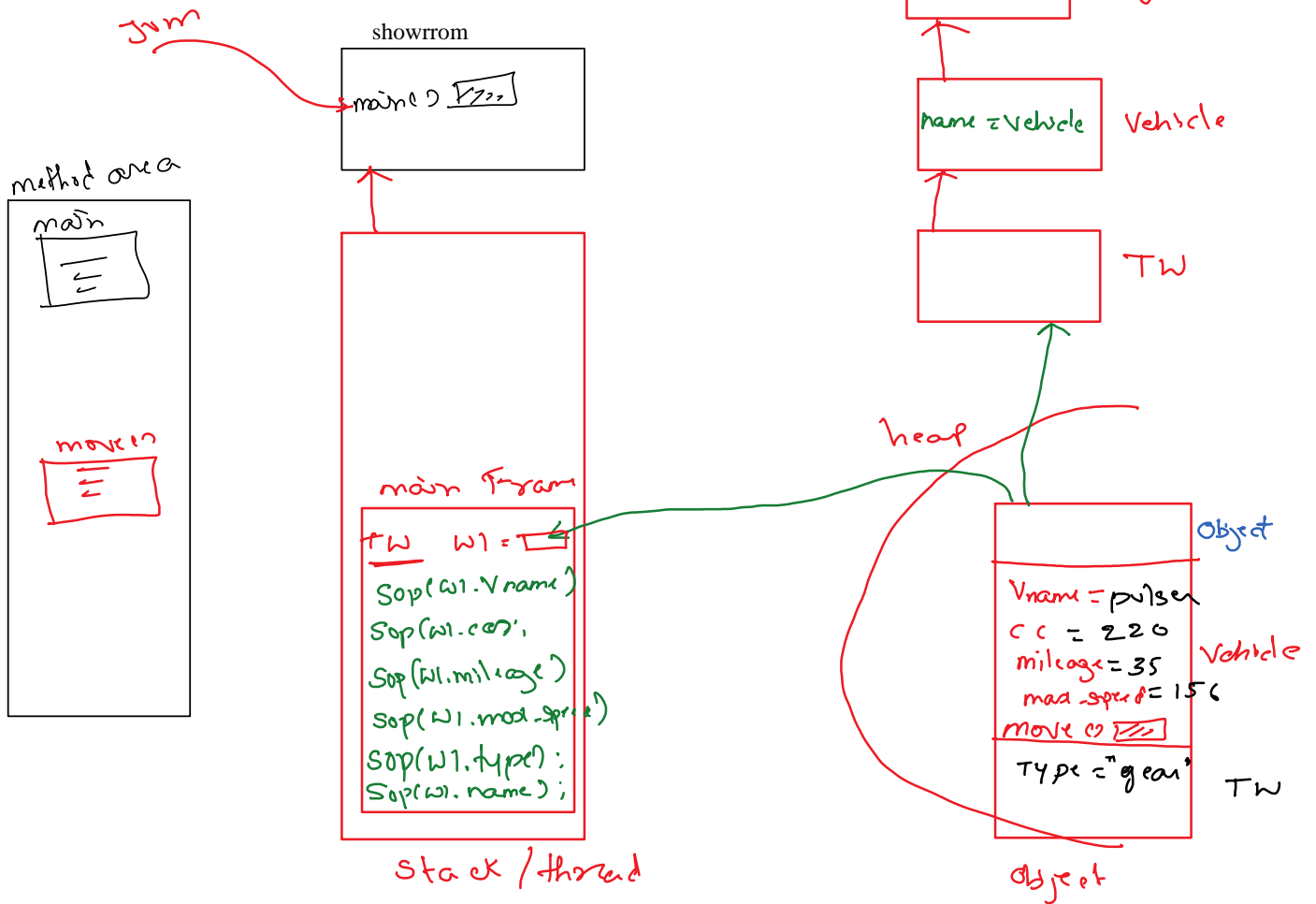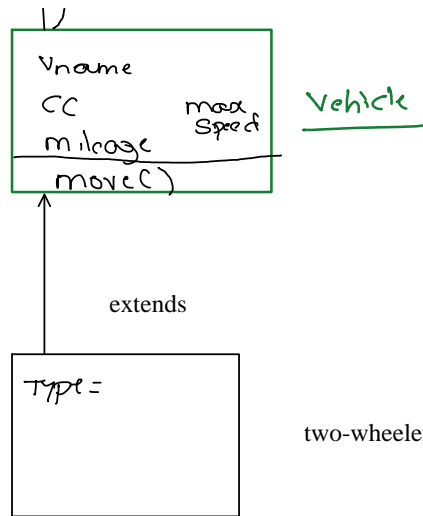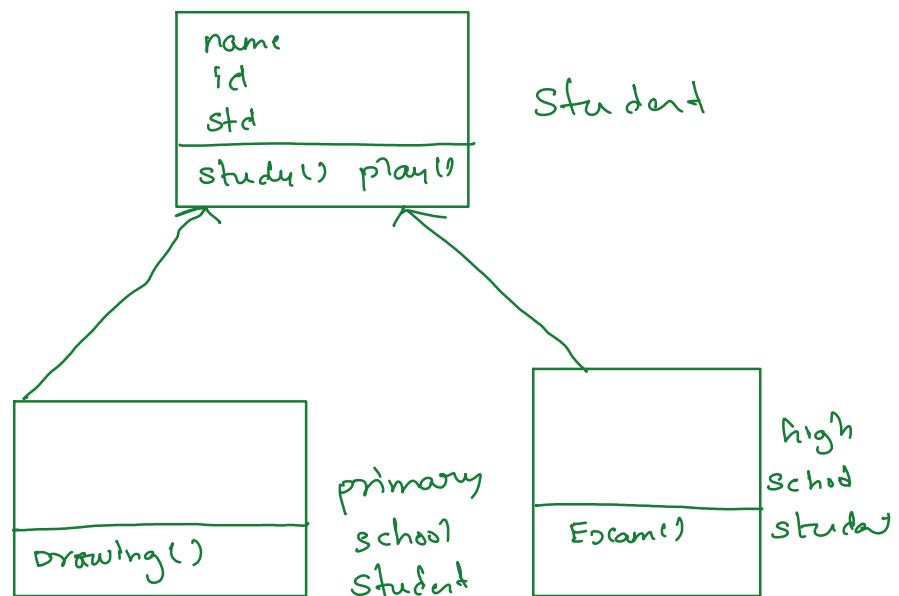
Ex

object

ename
eid
sal
deptno
desg

Emp

work()

extends

domain

TE

object

Emp

JVM

company

main() = ✍

method area

main()

work()

main frame

emp obj1 = 🔑

TE obj2 =

stack memory/thread

TE

heap

object

ename=punga
eid=1
sal=35000   Emp
deptno=1
desg="MTE"

work() 🔑
object

object

ename=pungi
eid=2
sal=30000   emp
deptno=1
desg="MTE"
work() 🔑

domain="ecomerce"
TE

Cx -2

object1

Vname

Vehicle

V

Vname
CC   max
Speed
mileage
move()

Vehicle

extends

Type =

two-wheeler

Jvm

showrrom

main()

method area

main

move()

Object

name = Vehicle   Vehicle

TW

main Frame

TW   W1 =
Sop(W1.Vname)
Sop(W1.cc);
Sop(W1.mileage)
Sop(W1.mod-speed)
Sop(W1.type);
Sop(W1.name);

stack/thread

heap

Object

Vname = pulser
CC = 220
mileage = 35   Vehicle
max speed = 156
move()
Type = "gear"   TW

Object

o/p

pulser

220
35
156
gear
Vehicle

is it possible to inherit static members ? yes we can

extends keyword can inherit static and non static members

imp
prove that static members also inheritable by using extends?

Assignment - 1

```
name
id
std
study() play()
```
Student

```
Drawing()
```
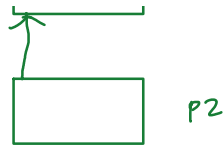primary
school
Student

```
Exam()
```
High
school
Studat

driver:- school

create an object for HSS and PSS

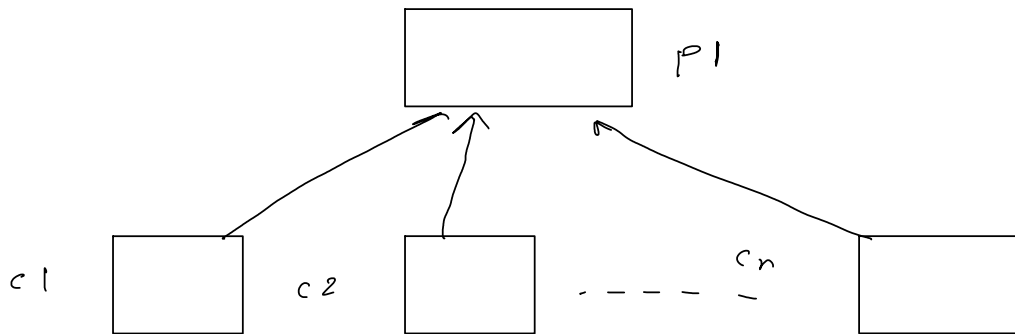**Type of inheritance by using extends keyword**

single level inheritance

P1

multi level inheritance
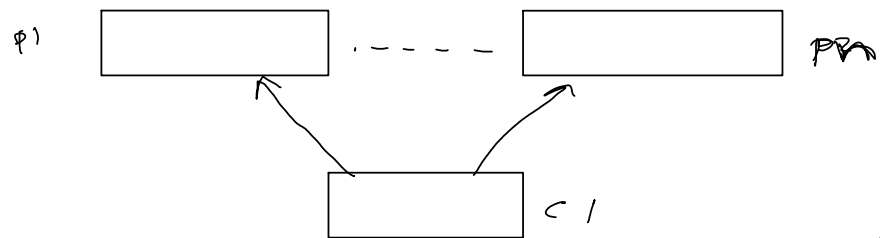
P2

P1

P2

P3
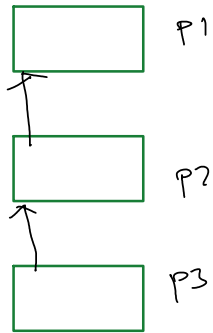
hierarchical inheritance

P1

C1   C2   Cn

multiple inheritance

P1   PPn

C1

not possible by using extends → implements
                                    interface

hybrid /derived inheritance
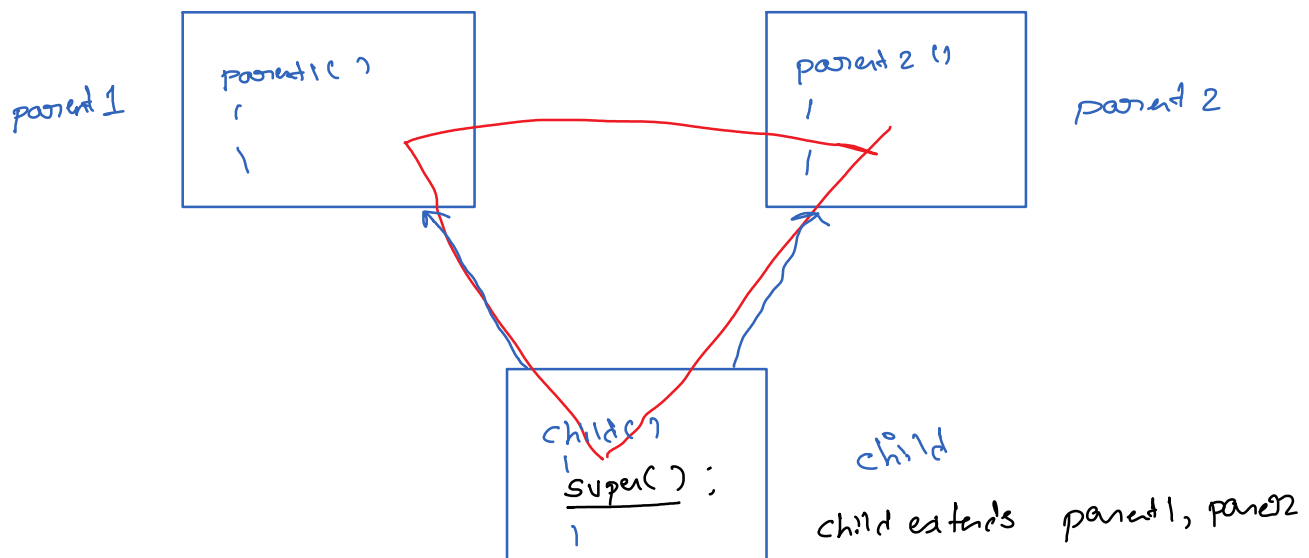
Supermost   P1

Super   S1   S2

C1

Sub
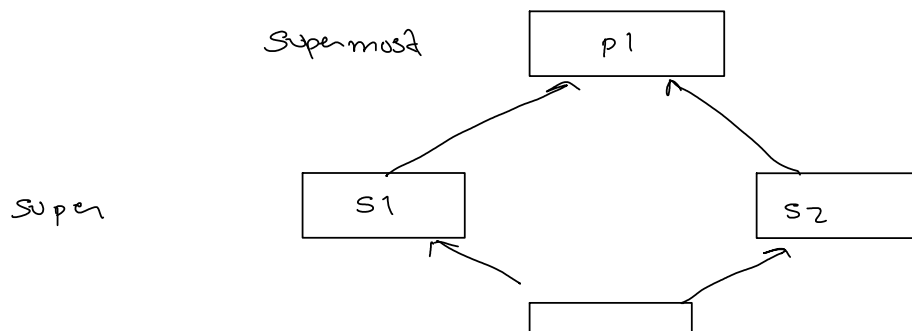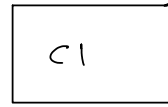
by using eaterds we can't achive hybrid
we can achieve through implements (interface)

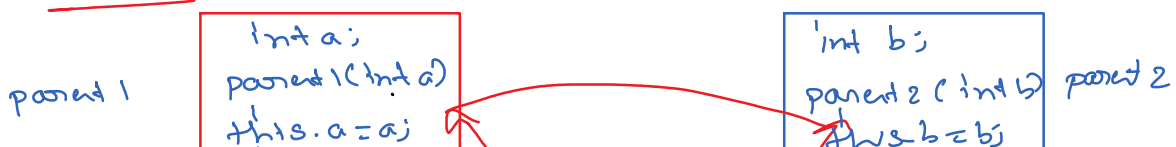super keyword : it is used to access a members of super class

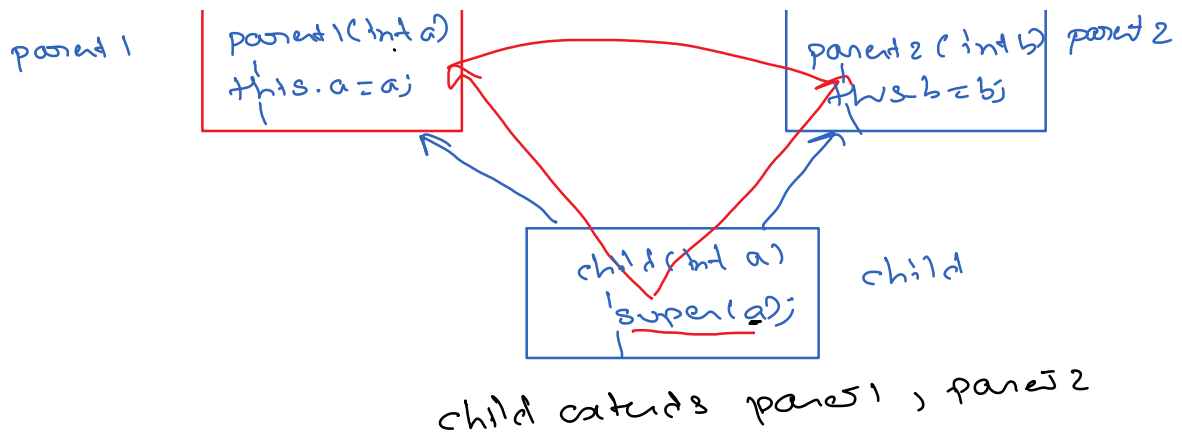we can't use super inside static context ,

Diamond problem

Reason 1   super();

parent 1
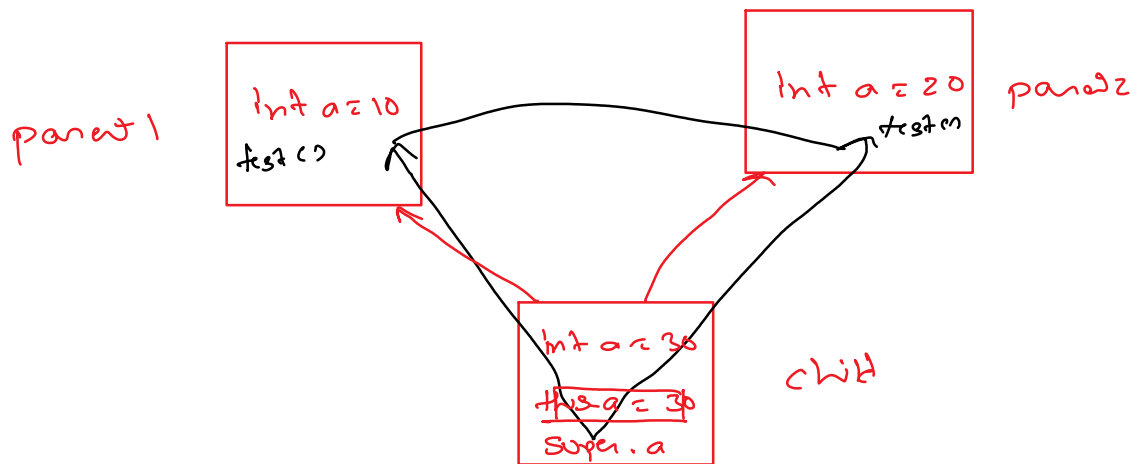
parent1( )
{
}

parent 2 ()
{
{

parent 2

child()
{
super( ) ;
}

child

child eaterds parent1, parent2

Reason -2   same constr in a parent classes

parent 1

int a;
parent1(int a)
this.a = a;

int b;
parent2 (int b)   parent 2
this.b = b;

parent 1

parent1(int a)
this.a=a;

parent2(int b)
this.b=b;   parent 2

child(int a)
super(a);   child

child extends pares1, pares2

Reason - 3

parent 1

int a = 10
test()

int a = 20   parot2
test()

int a = 30
this.a = 30
super.a   child

we can
solve
by using

implements ⟶ interfaces
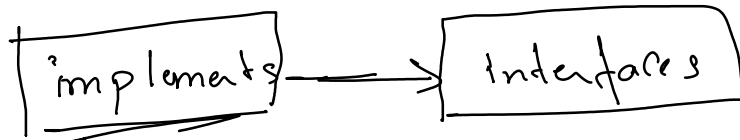
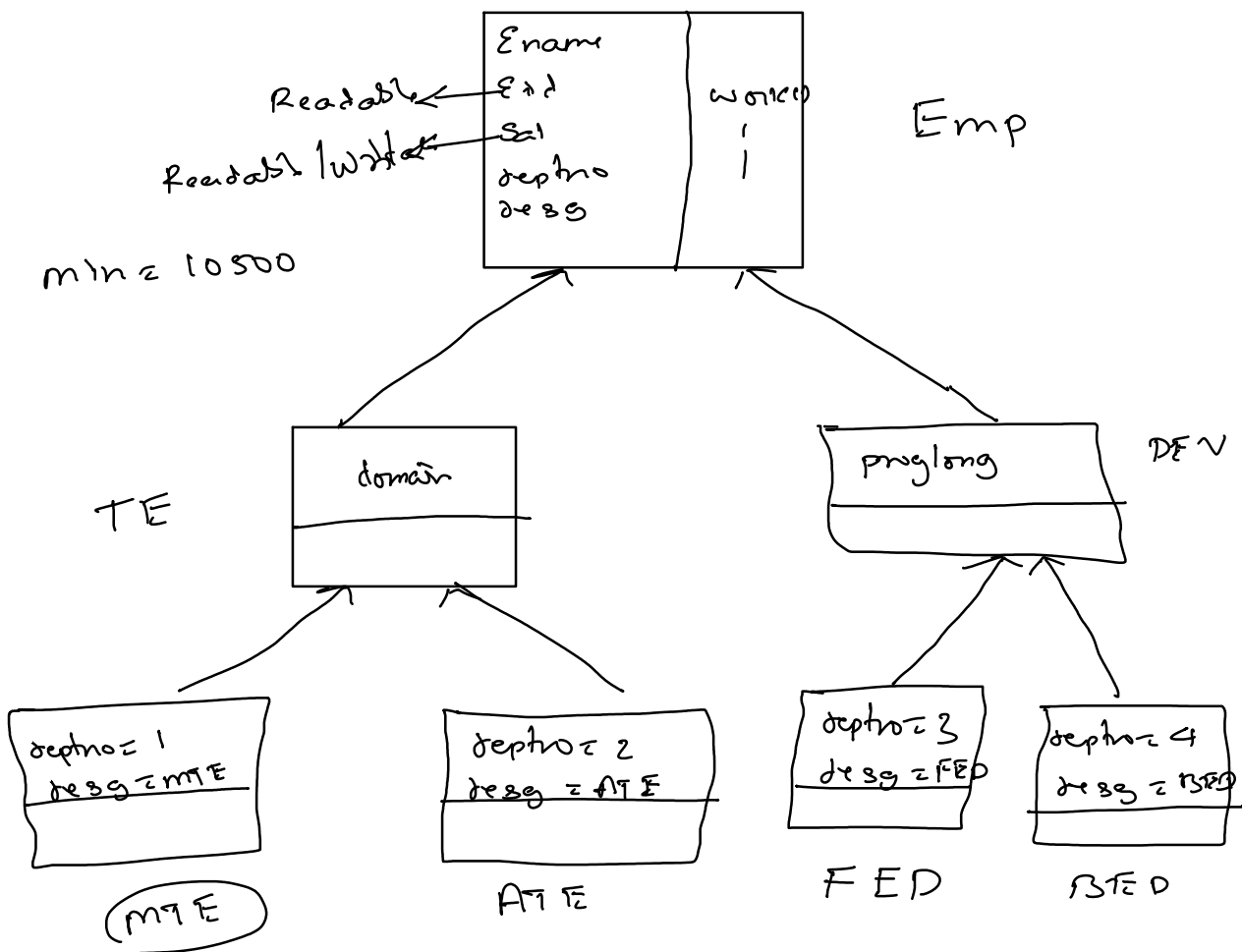Diamond problem :- it is a major problem accurse when we try to inherit more than one parent/super classes

problem-1 if both parent class contains default constructor and if sub class calling parent class constructor with super() then it will get confused whether to call which parent class

problem-2 if both parent/super class contains a parameterized constructor with a same type of Formal Arg and from a sub class if you try to call a parent class constructor by using super() with a same actual arg -->again it will get confused whether to call which parent class
problem-3 if both a parent class contains a members

(var/method) with a same name then from a sub class if we try to call a parent class member by using super (super.member_name) then again it will get confused whether to call which parent class

1) what is inheritance ?
      flow defn
      explain extends keyword
      explain types
2) what is diamond problem
        reasoon-1
        reason-2
        reason-3

3) difference between this() and super()
4) difference between this and super

name, Eid, sal

deptno   desg

1   MTE

2   ATE

3   FED

4   BED

```
if ( sal 3 = 10500 )
{
    tws_sal = sal
}
else
{
    tws_sal = 10500
}
```