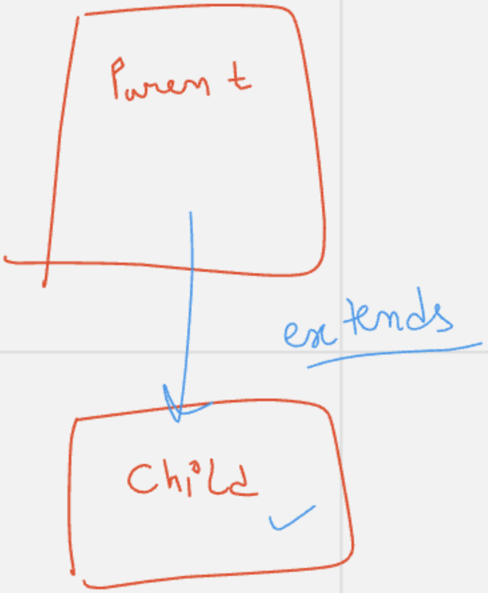
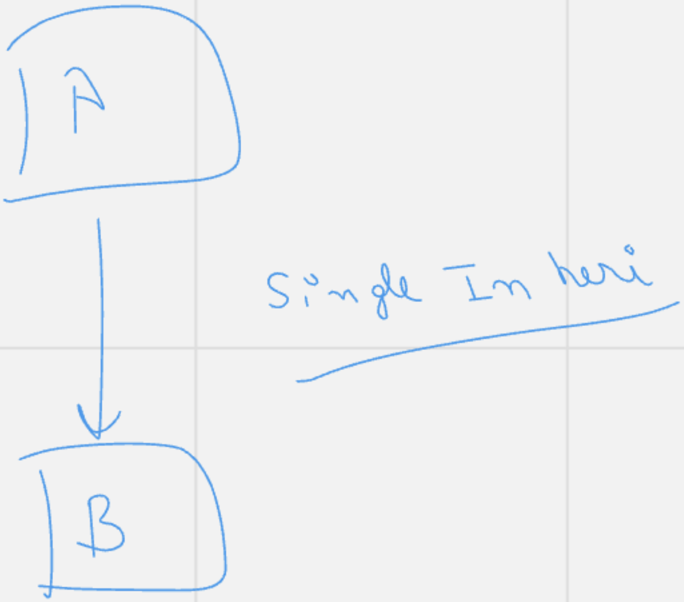
Inheritance

Run time

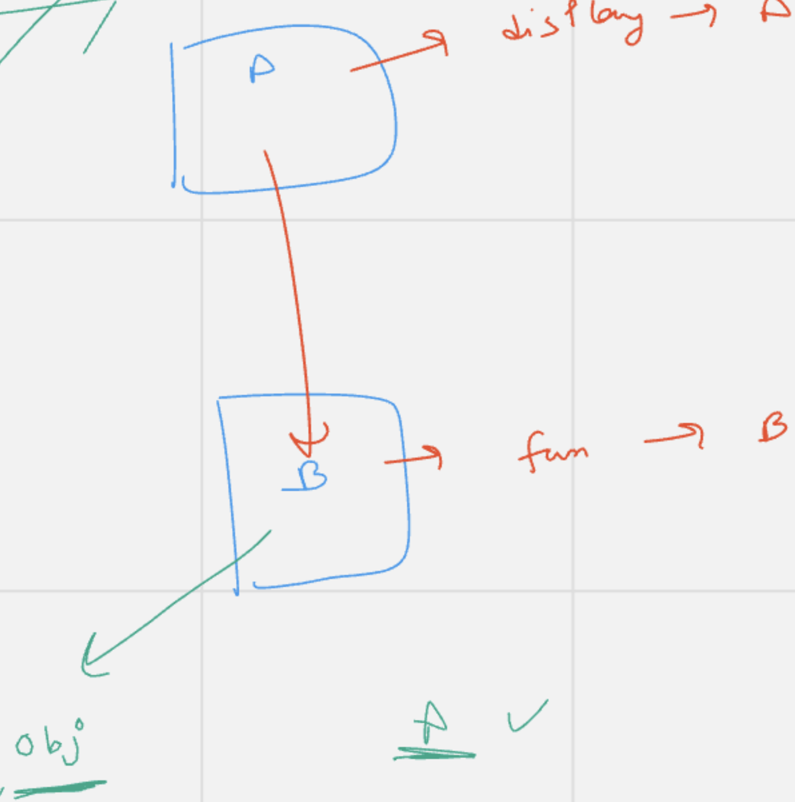
[Interface]

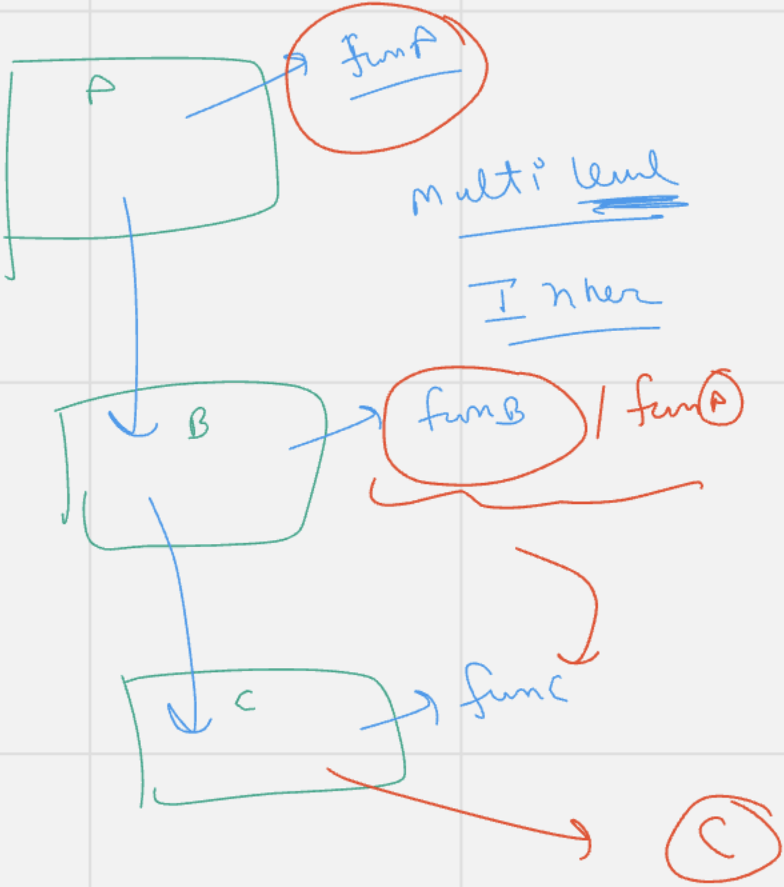


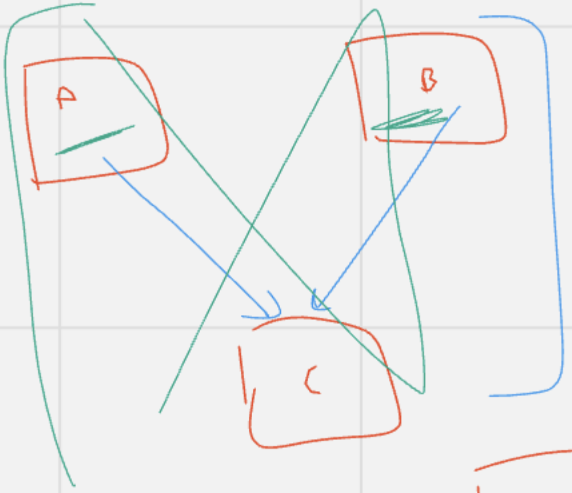


Public, default, Protected

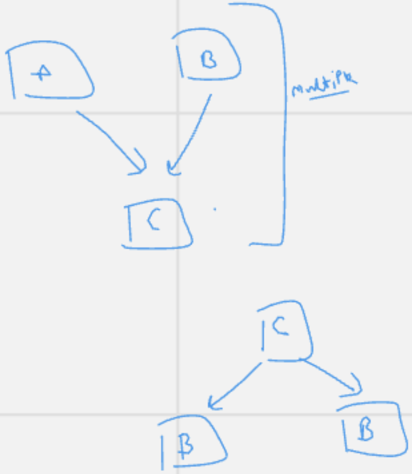
```
class A {

  void display() {
    System.out.println("A");
  }
}

class B extends A {

  void fun() {
    System.out.println("B");
  }
}

  B obj = new B();
      obj.display()
```

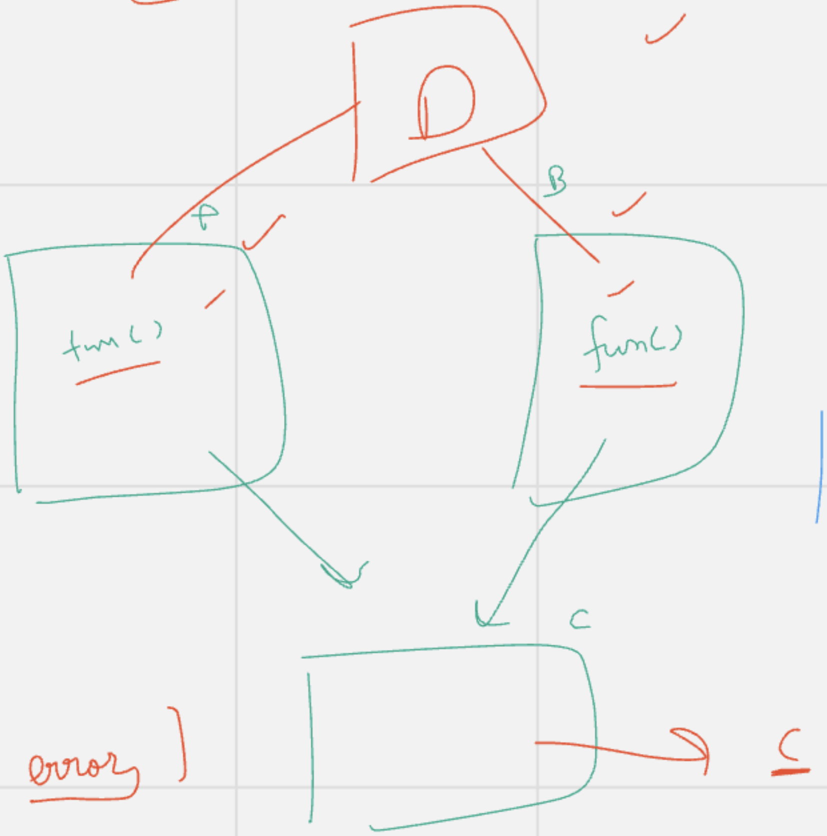


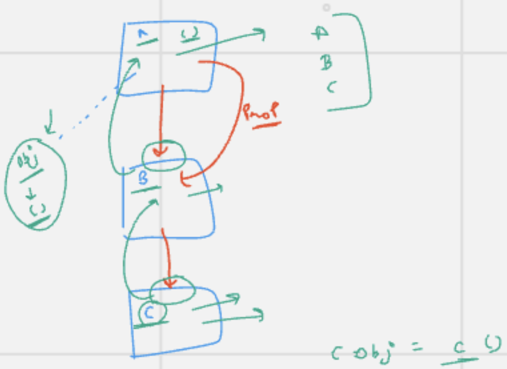

A
funA
multi level
Inher
B
funB / funA
C
funC
C



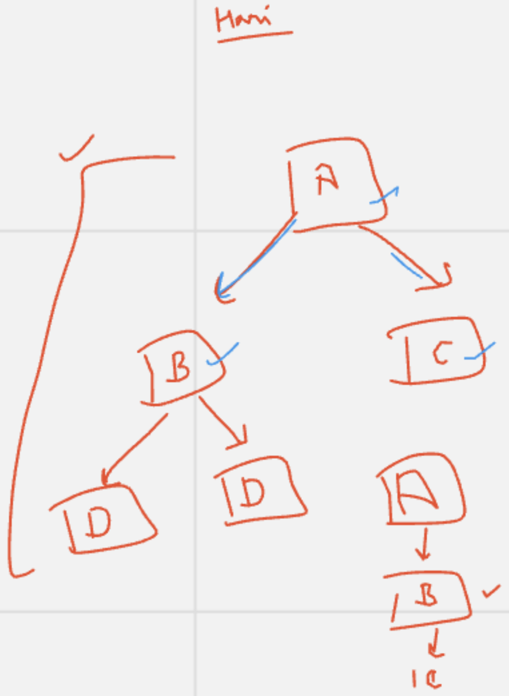
Multiple
Inheritance

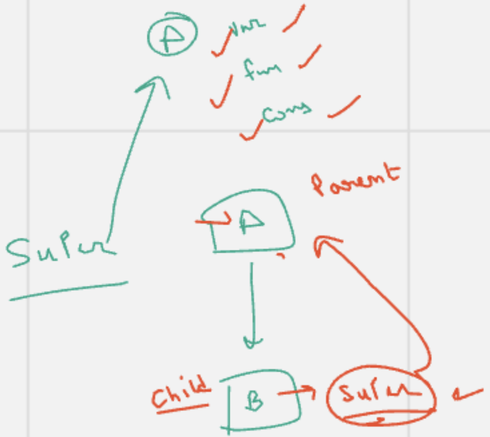


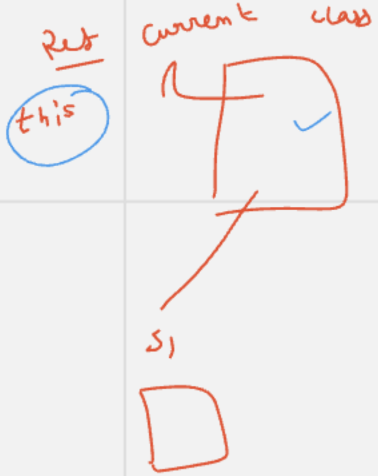


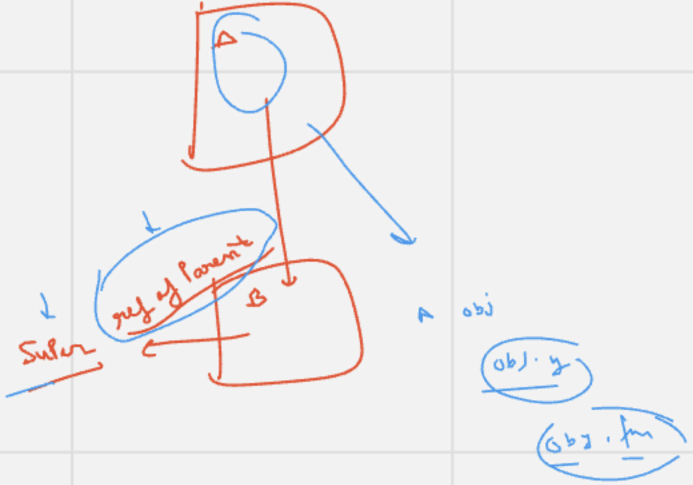


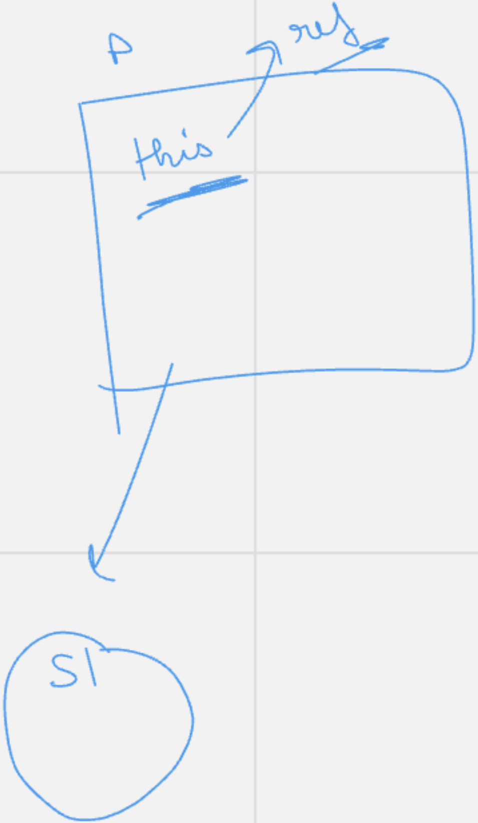
C. fun()

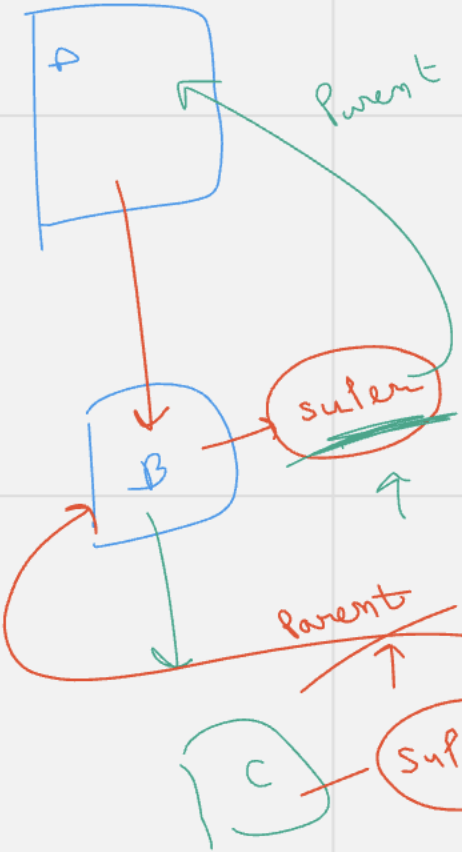

Hari
A
B
C
D
D
A
B
IC

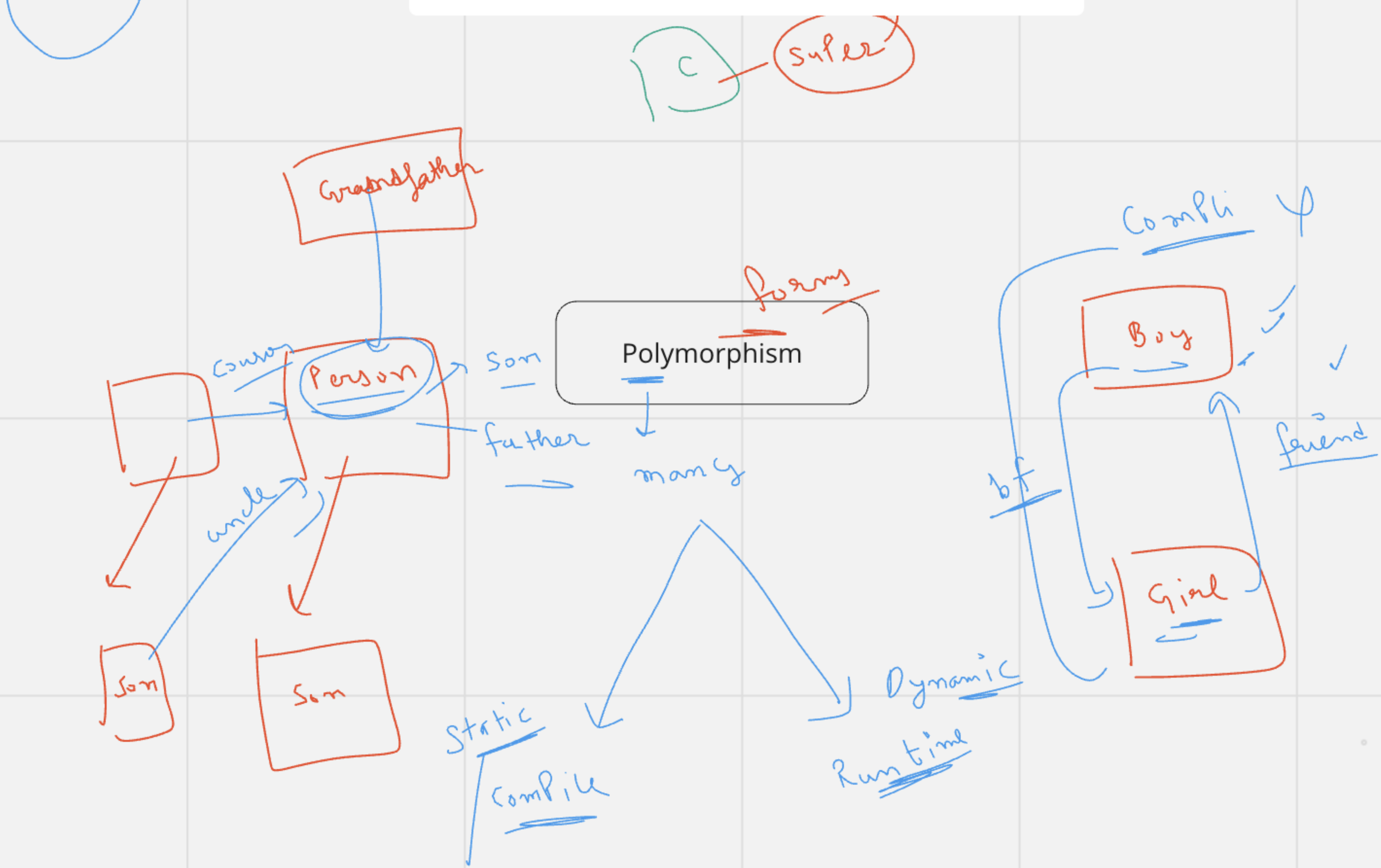

A
var
fun
cons
Super
A
Parent
Child
B
Super

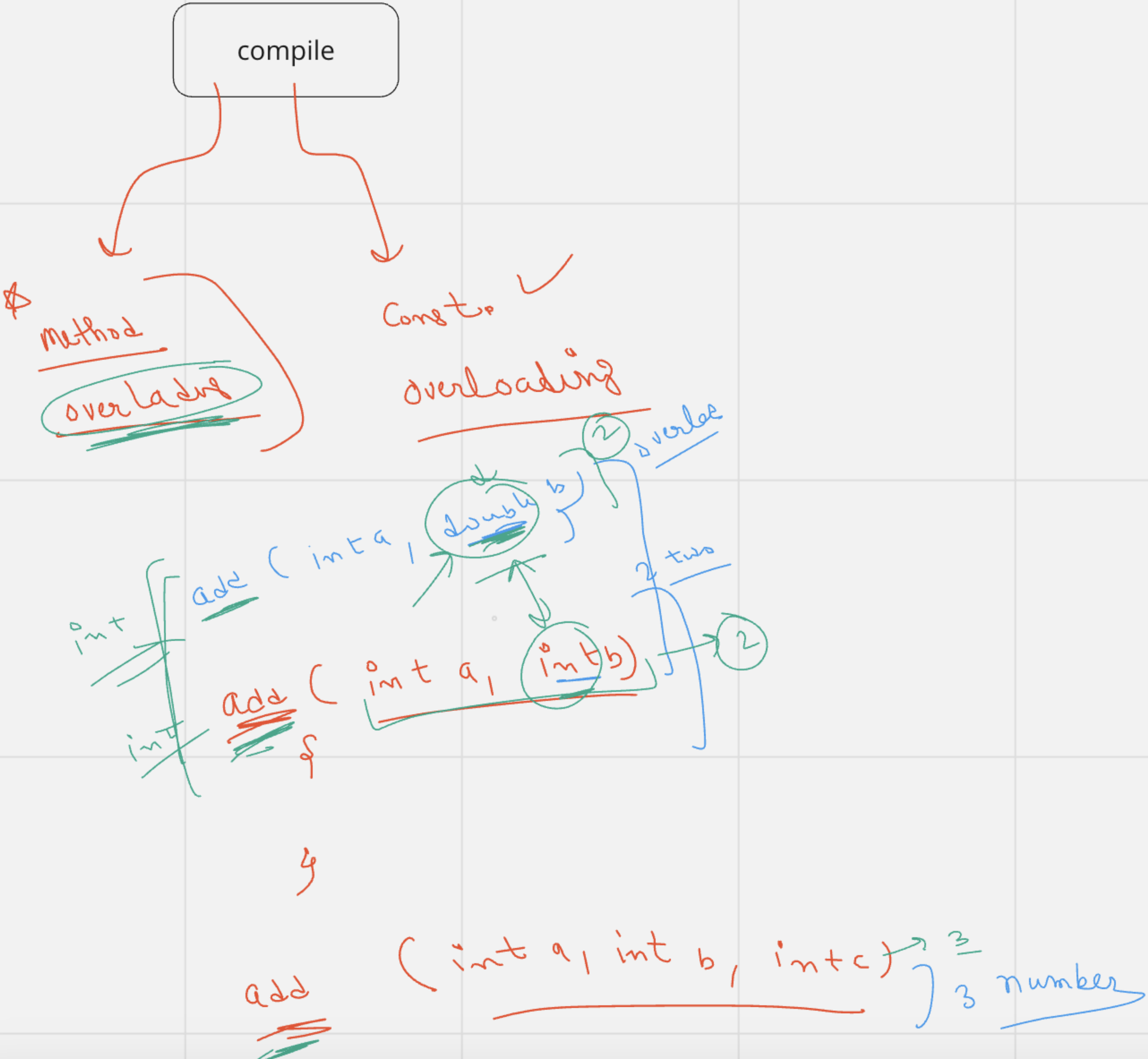

Ref
Current
class
this
S1


A
ref of Parent
B
Super
A
obj
obj . y
obj . fun


A
ref
this
S1


A
Parent
super
B
Parent
C
super


Polymorphism



compile

method overlading

Constr ✓

overloading

add (int a, double b) } 2 order

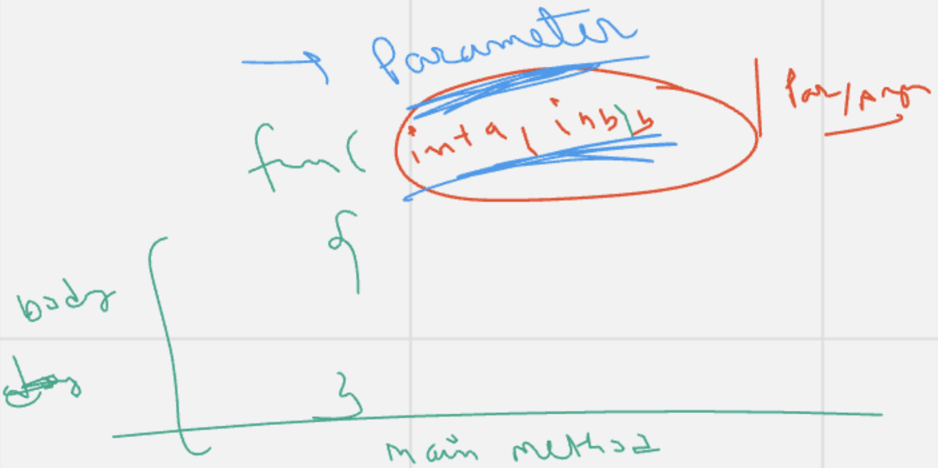
int add (int a, int b) } 2 two → 2

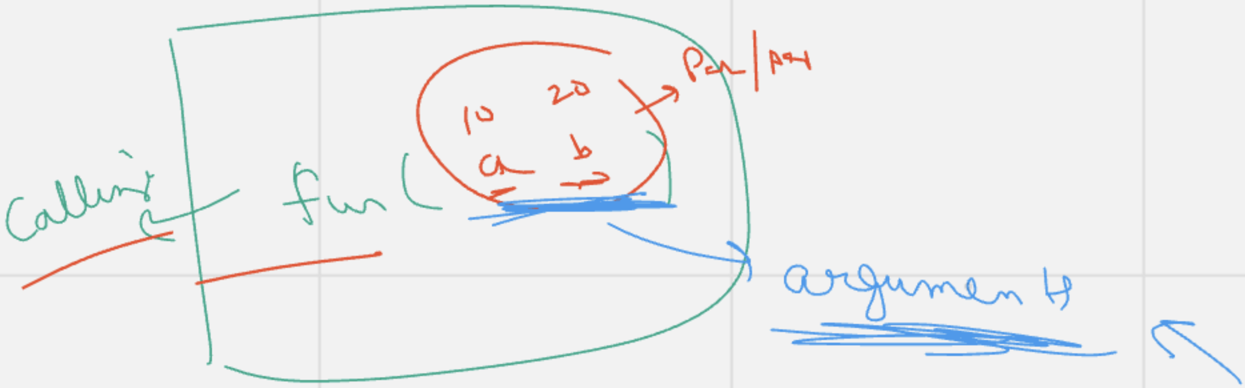
int

{

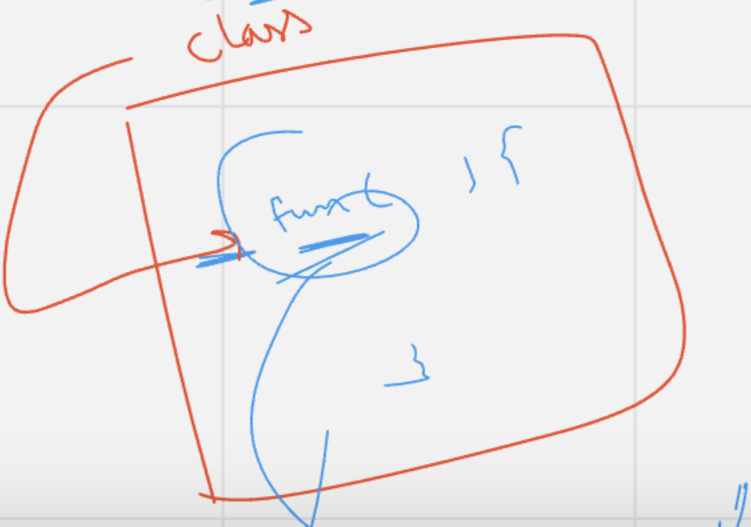
}

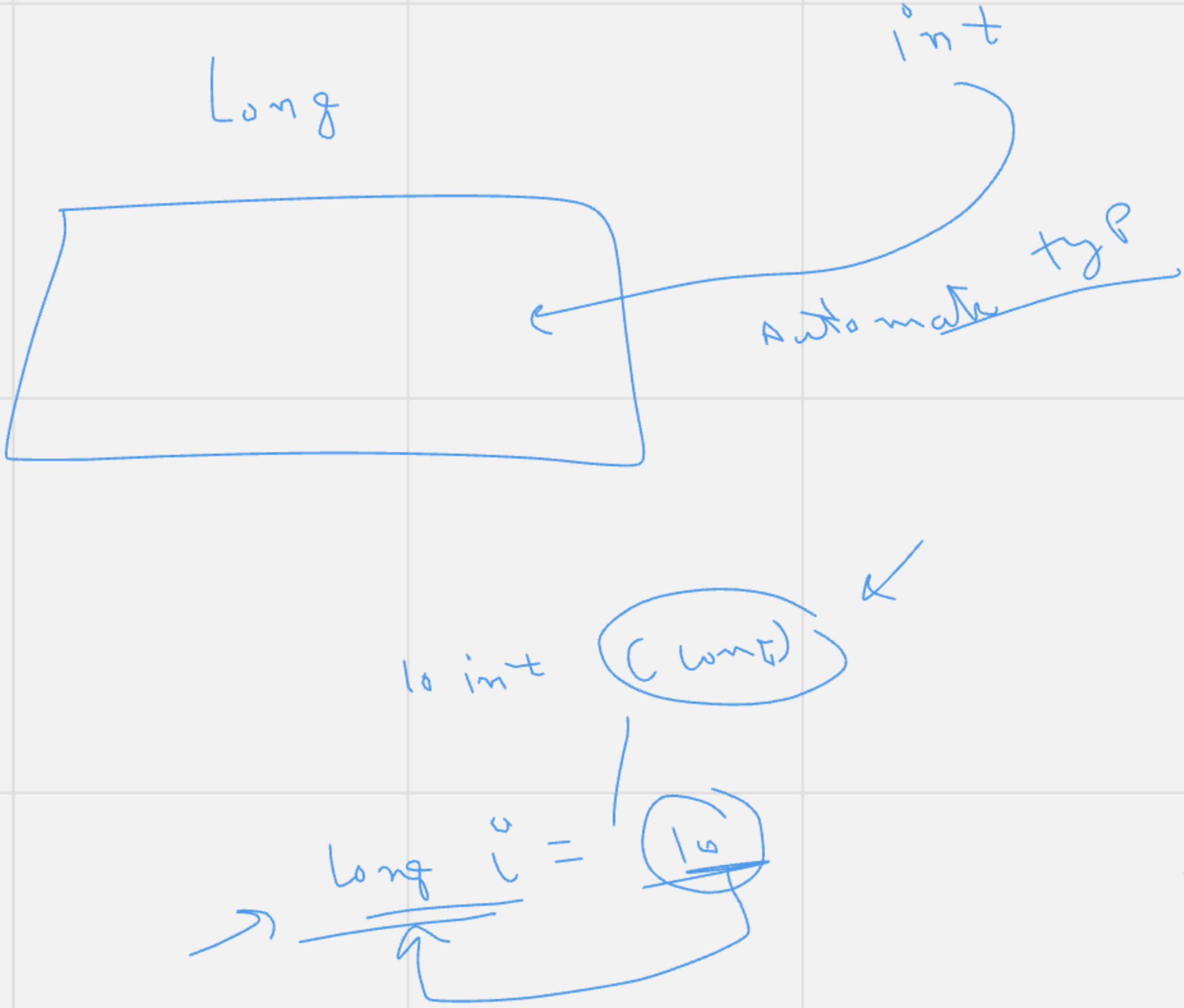
add (int a, int b, int c) → 3 } 3 number





Static functions



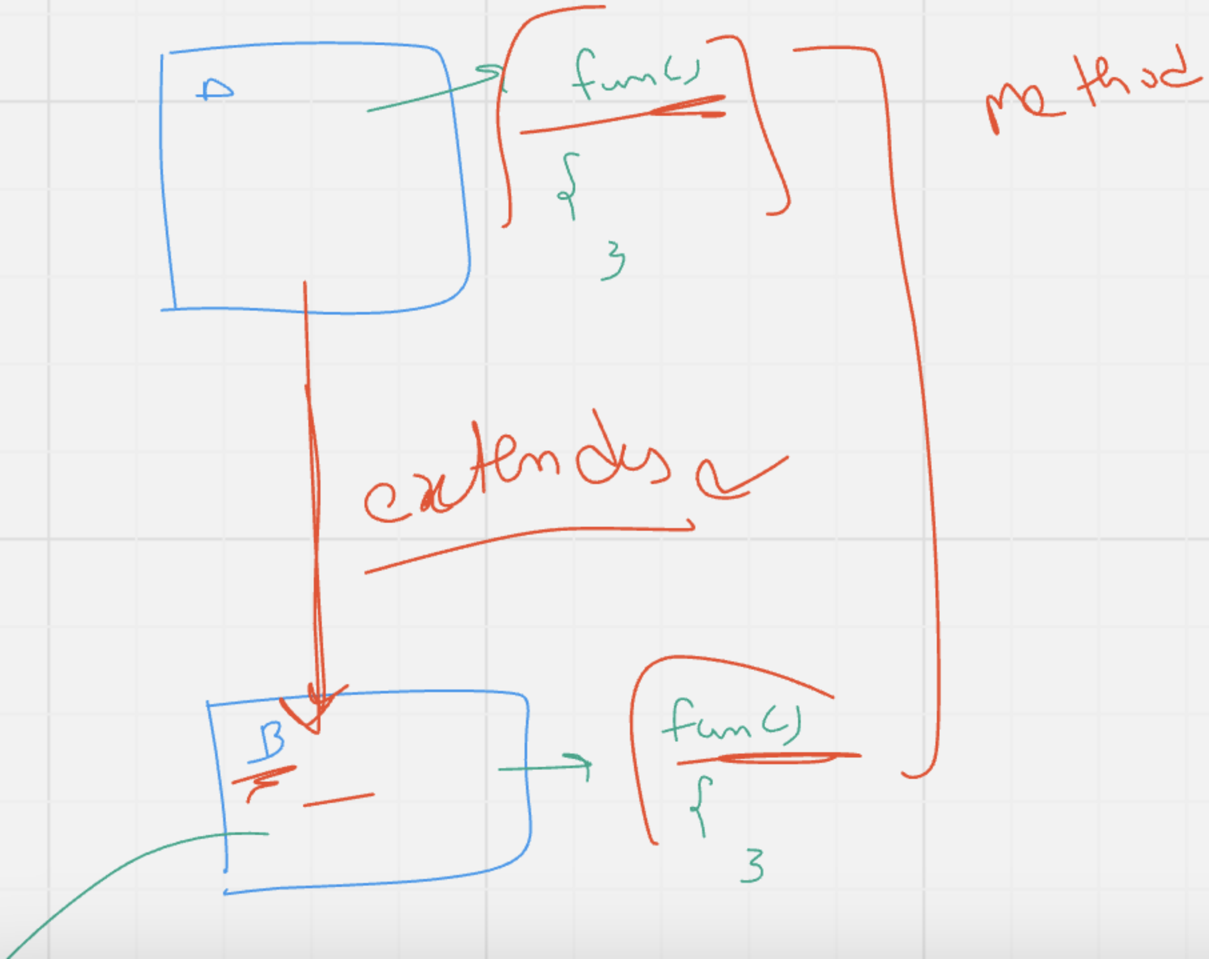


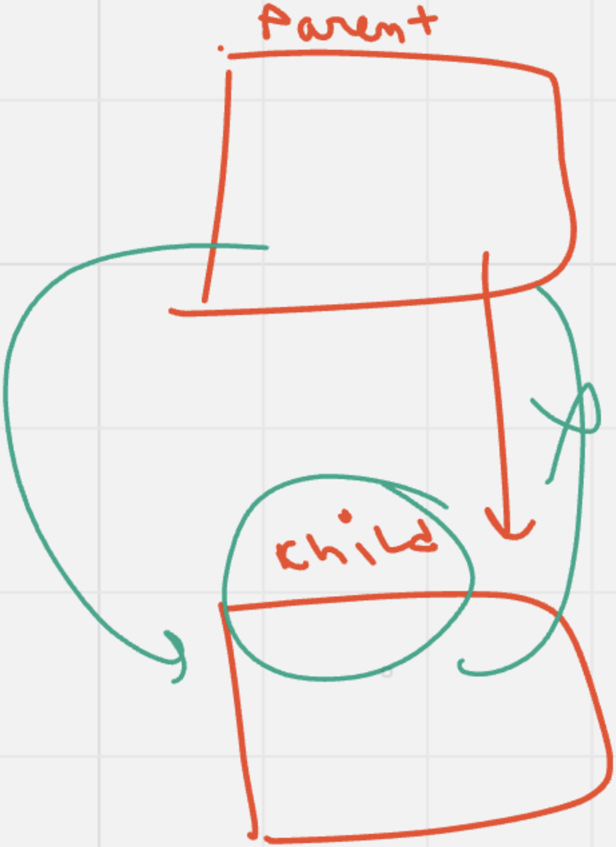
Long
int
automatic typ
10 int
(Long)
long i = 10

Runtime Polymorphism

↓

method over-riding





✓ Parent r = new child

✗ Child c = new Parent ✗

c → Ref of child

```
class Parent {

    void fun() {
        System.out.println("P");
    }

}
class Child extends Parent {

    void fun() {
        System.out.println("C");
    }

}
```

```
Parent c = new Child();
c.fun();
```

Parent c ;

runs X

c = new child()

c.fun() ;

C ←

At compile Time ✓

At runtime

At runtime

```
class Parent {
    int x = 2;
}

class Child extends Parent {

    int x = 4;
    int y = 3;
}

public class MyClass {

    public static void main(String args[]) {

        Parent c = new Child();
        System.out.println(c.x + c.y);
    }
}
```

c

Parent c ;

c = new Child();

Sop(c.x + c.y);

Compile error

y is not Present

At compile time