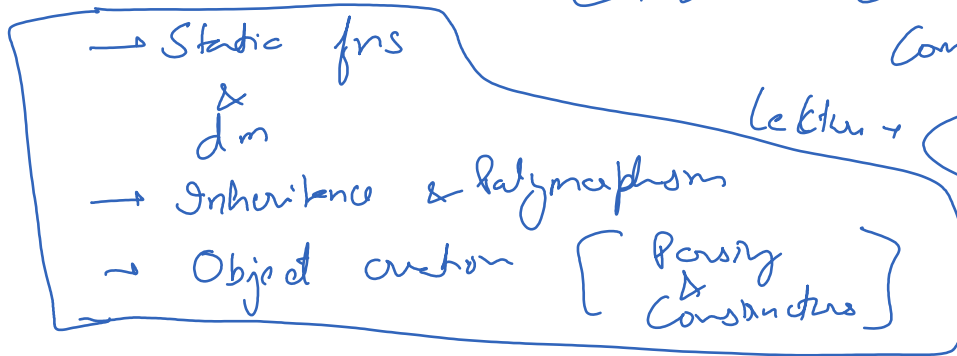
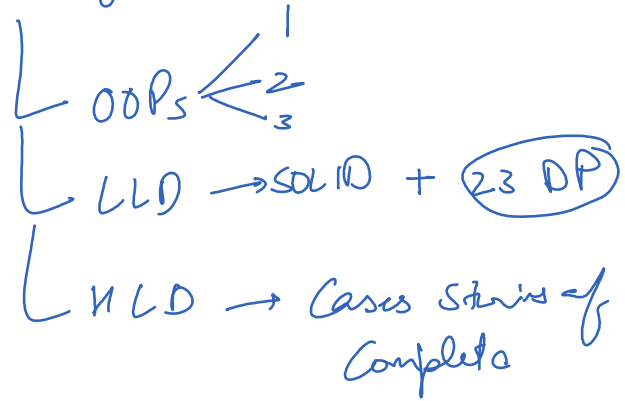


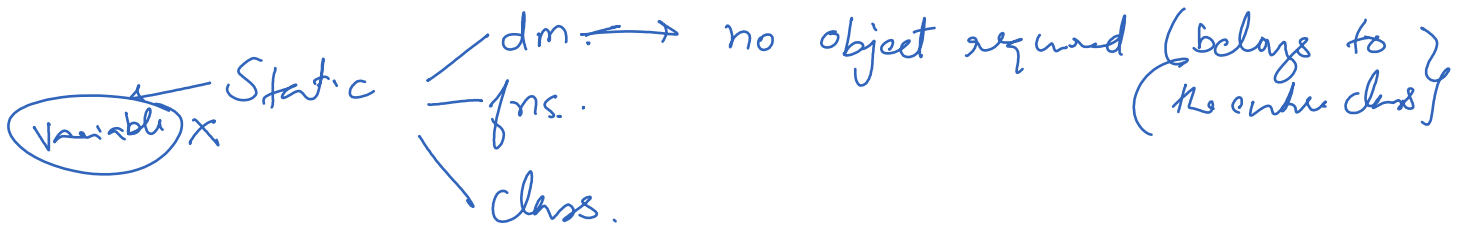
System Design

We begin at
2:25



Lecture → 4:00 - 4:15

Doubts → 4:15 +



```
static class BankAccount {
    int anum;
    int money;
    static int roi;

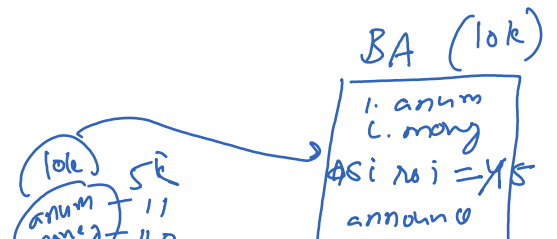
    void announce(){
        System.out.println(this
    }
}
```

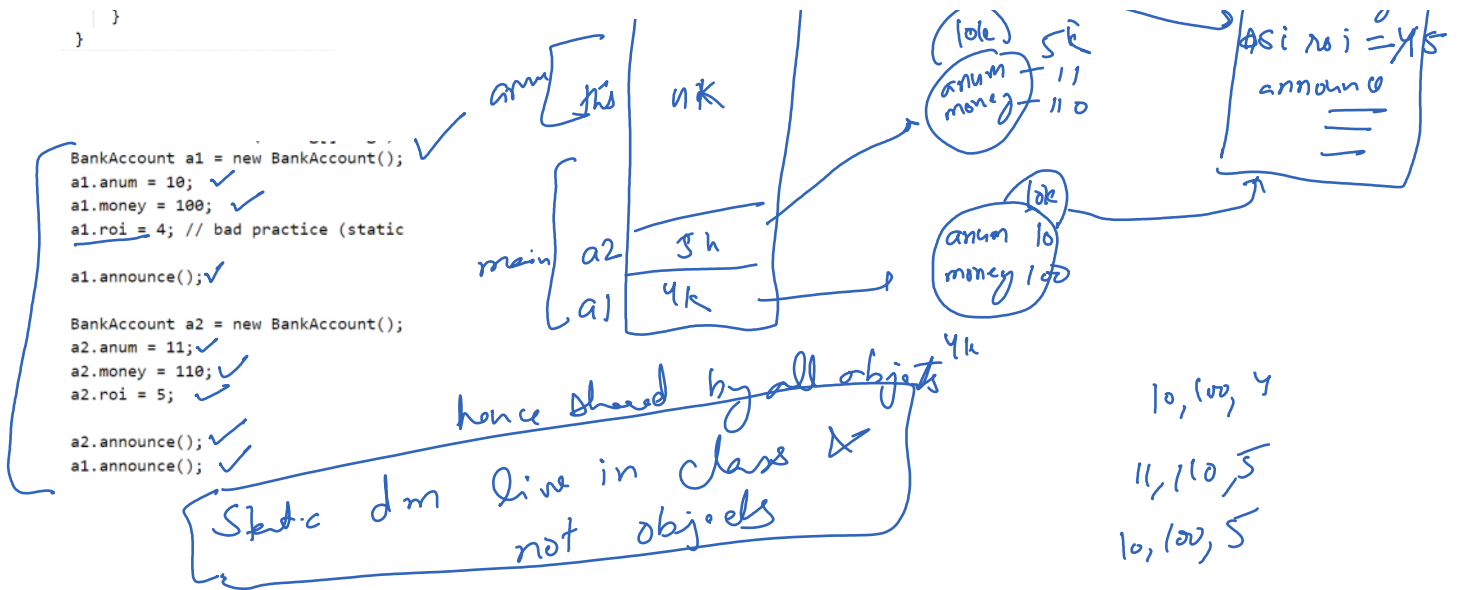
```
static class BankAccount {
    int anum;
    int money;
    static int roi;

    void announce(){
        System.out.println(this.anum + ",
    }

    static void decreaseROI(){
        roi--;
        // anum and money can't be used
    }
}
```

anum nk nk





Why warning? $a1.roi = 4$ ✓

Because static dm belong to class & not objects

BankAccount.roi = 4 ✓ (bad practice)

Static fns. → Such fns. which do not require object for their invocation { are invoked directly via class name }

Uses → Utility static fns.

2.49 - 2.52

USE → using static fns.

2:49-2:52

this keyword cannot be used in static fns. Why?
 non-static data members can't be used in static fns. Why?
 non-static fns can't be used in static fns. Why?

Inheritance & Polymorphism

P and C - all cases, except abstract fns and class

3:07-3:12

Donb/s

Notes

```
static class P {
    int d = 1;
    int d1 = 10;

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

    void fun1(){
        System.out.println("P fun1");
    }
}

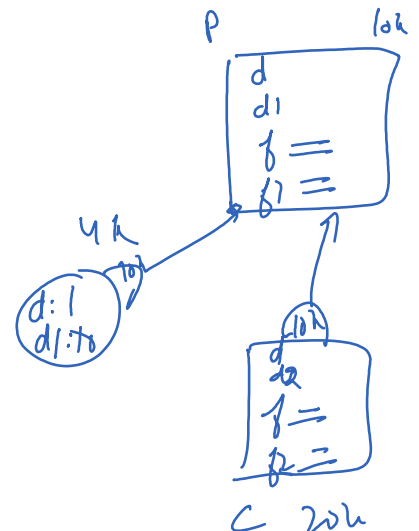
static class C extends P {
    int d = 2;
    int d2 = 20;

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

    void fun2(){
        System.out.println("C fun2");
    }
}
```

```
P obj = new P();
System.out.println(obj.d);
System.out.println(obj.d1);
obj.fun();
obj.fun1();
```

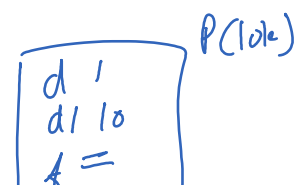
obj

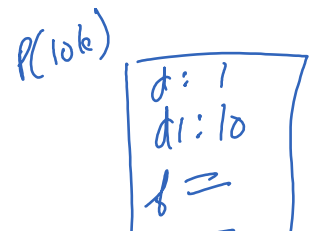
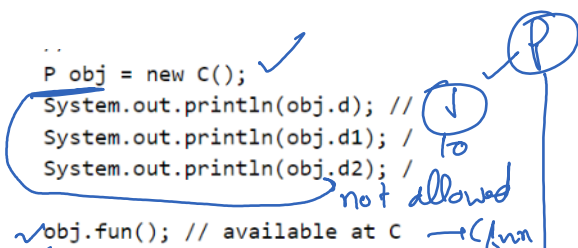
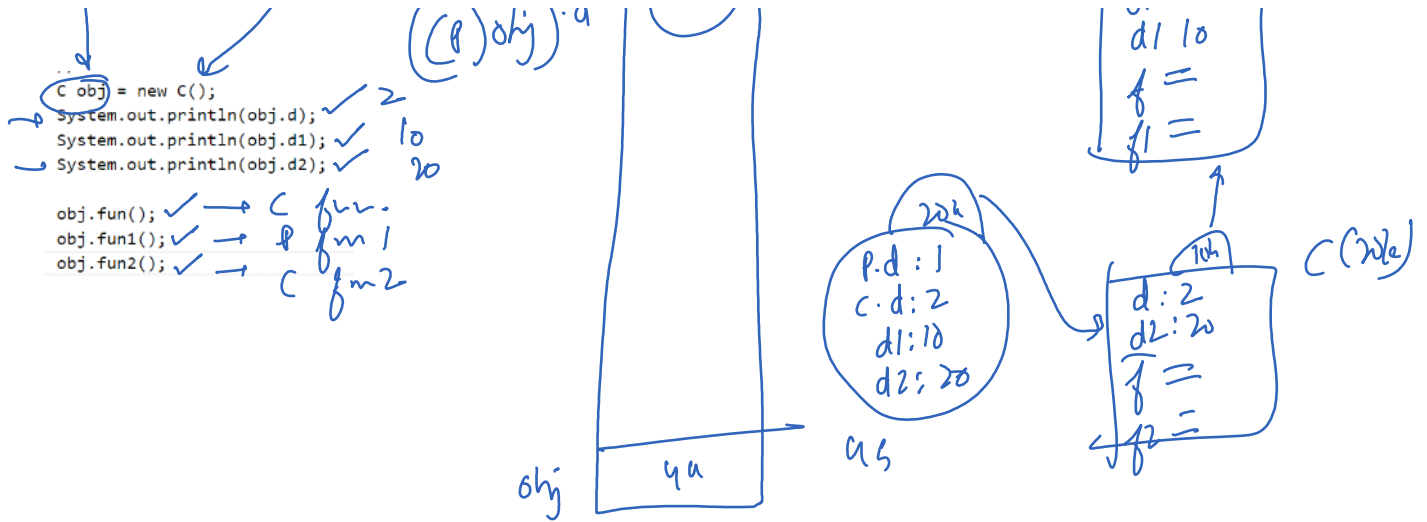


1. Compiler watches LHS i.e. ref
2. Runtime RNS i.e. instance

1. Compile 2. RT 3. Conflict → Same level → Ref

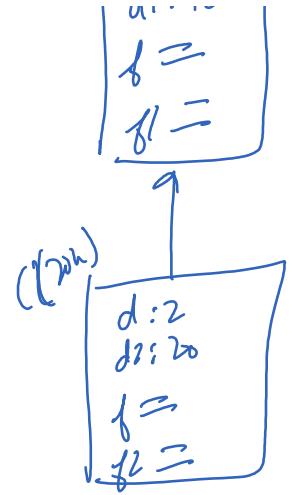
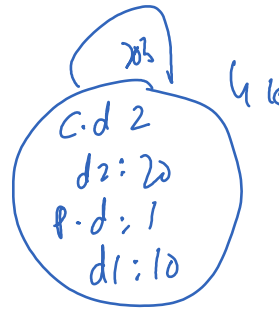
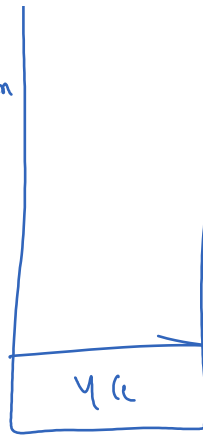
$(C) obj.d \rightarrow P$





```
System.out.println(obj.d2); // not allowed
obj.fun(); // available at C → C.fun
obj.fun1(); // available only → P.fun
obj.fun2(); // not allowed, b
```

↓
Compile error



```
static class P {
    int d = 1;
    int d1 = 10;

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

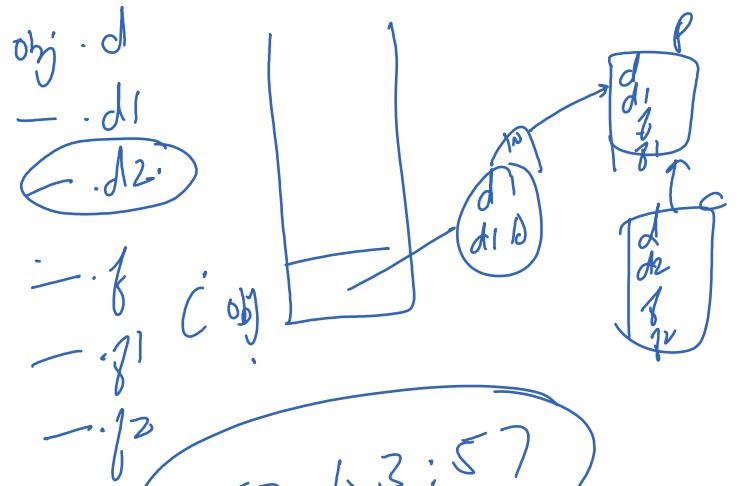
    void fun1(){
        System.out.println("P fun1");
    }
}

static class C extends P {
    int d = 2;
    int d2 = 20;

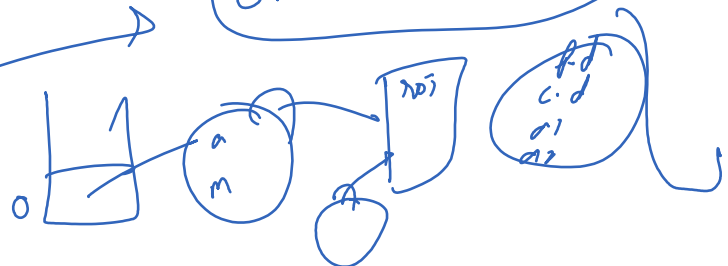
    void fun(){
        System.out.println("C fun");
    }

    void fun2(){
        System.out.println("C fun2");
    }
}

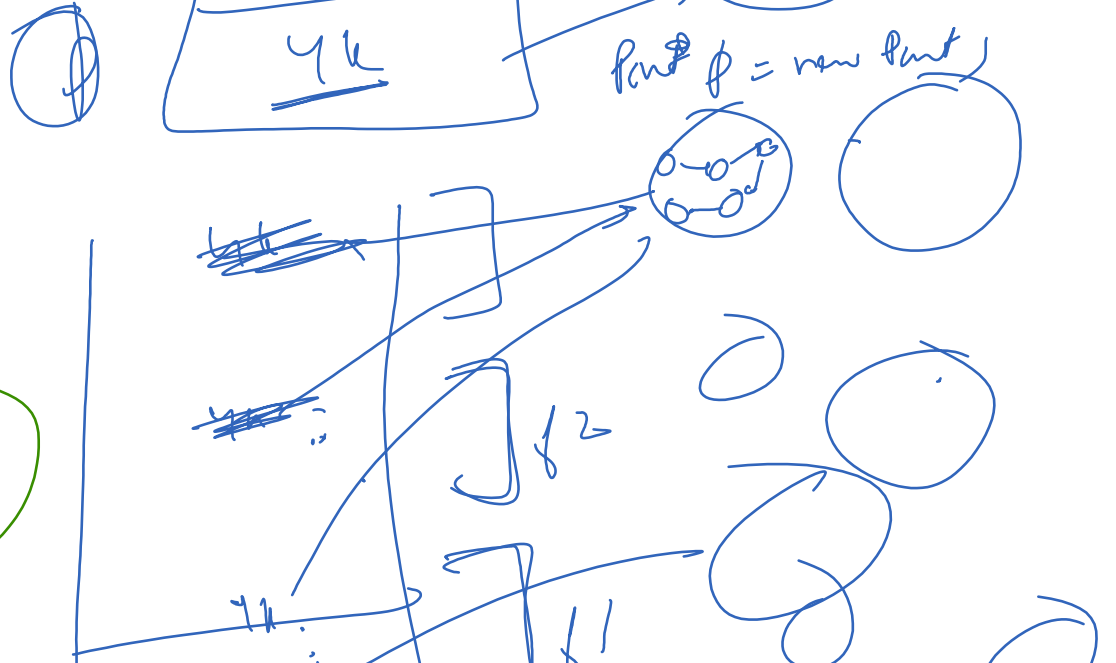
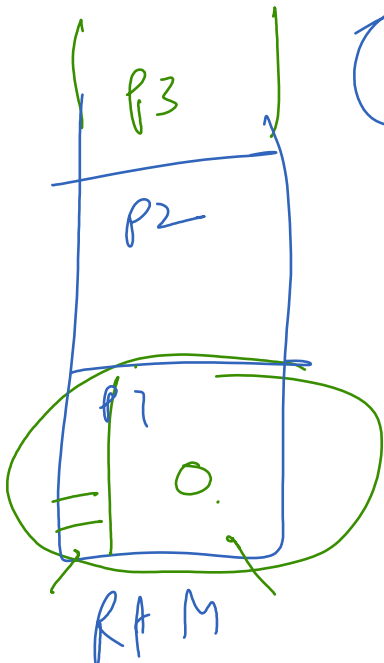
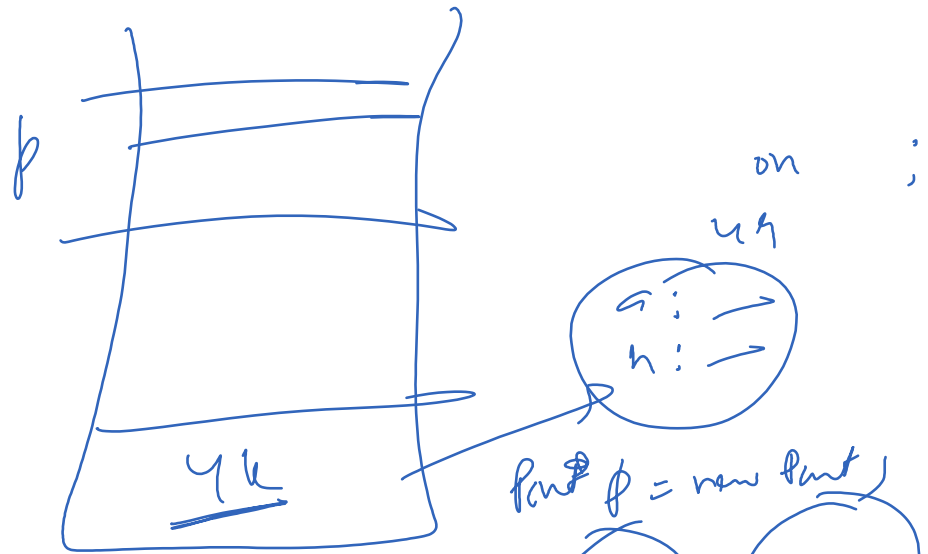
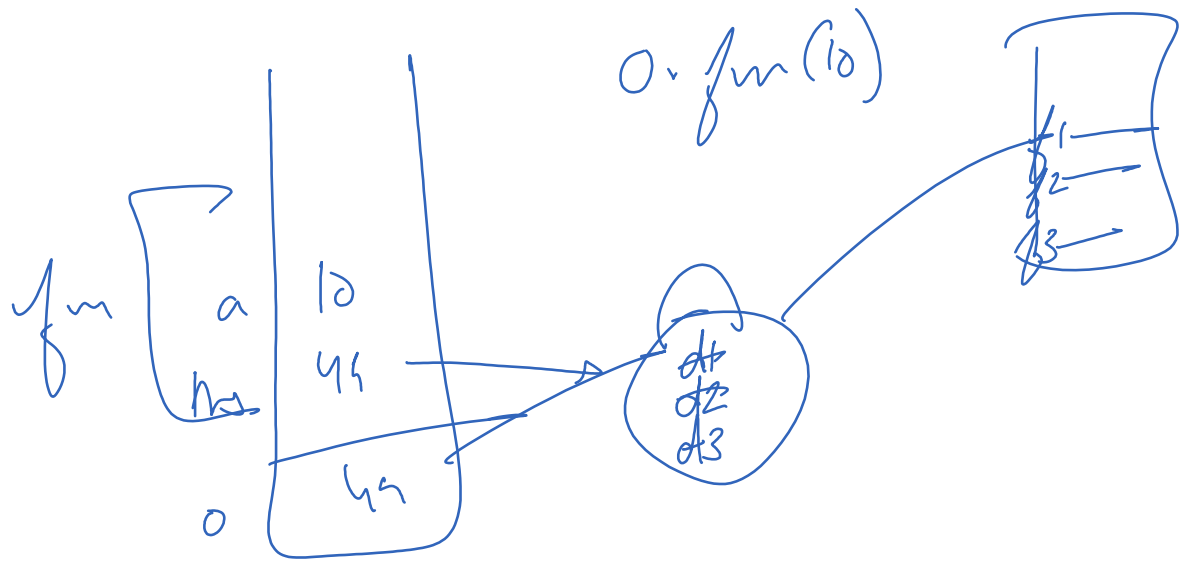
public static void main(String[] args) throws Exception {
    // case 2 => C and p
    C obj = new P();
    // why not allowed
}
```



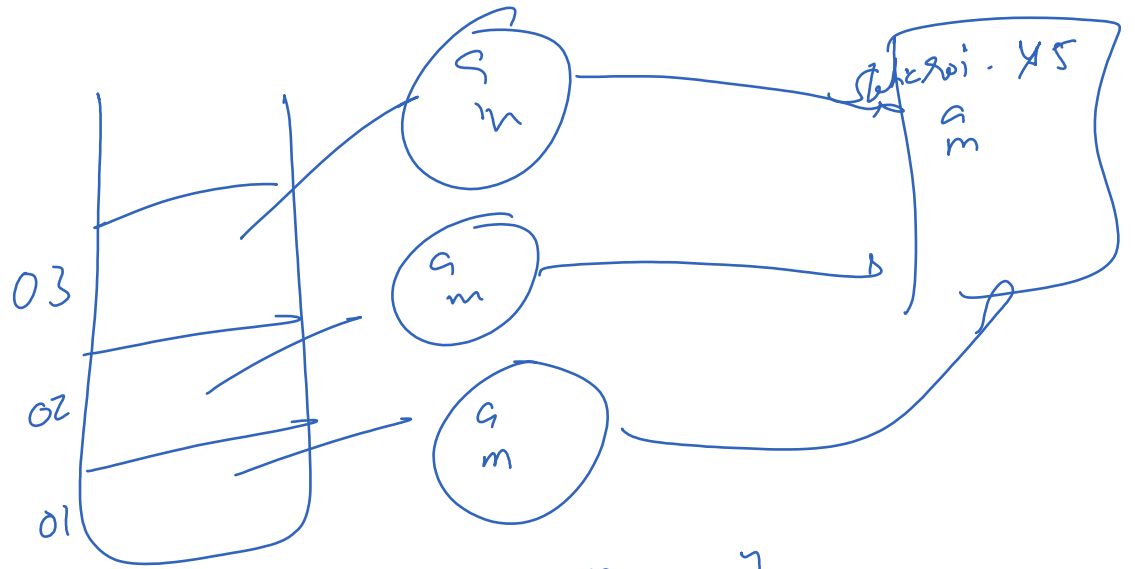
3:52 to 3:57



obj



RAM



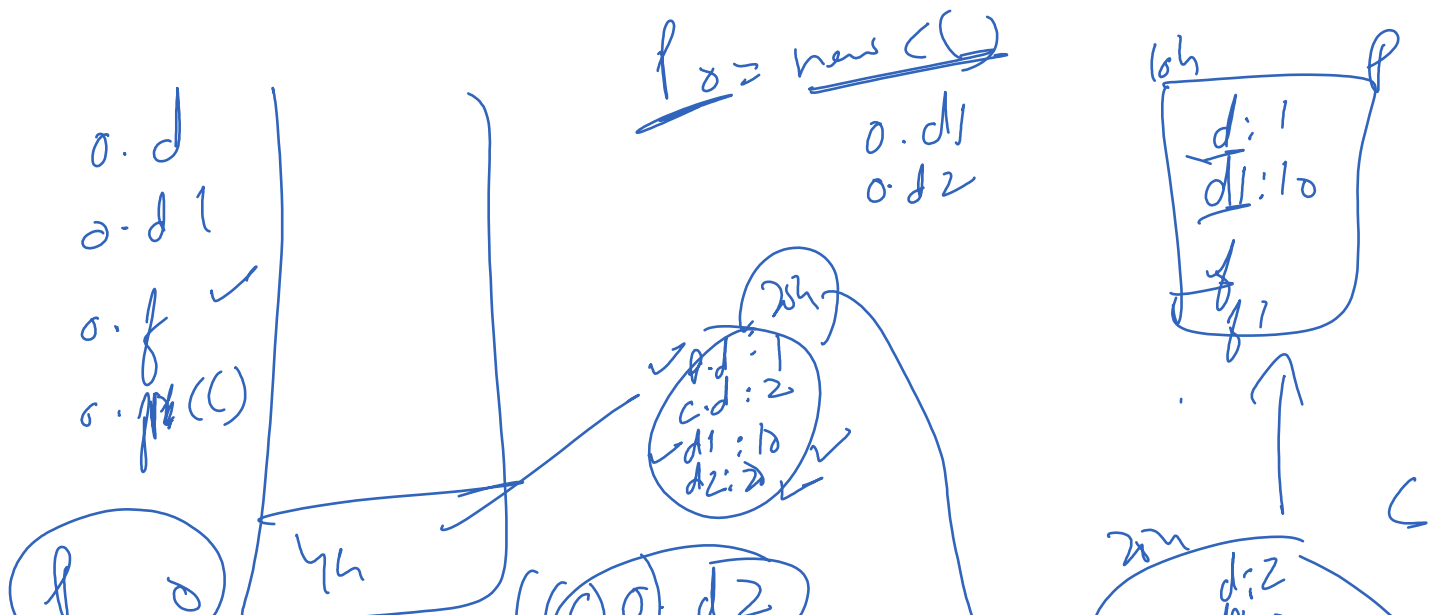
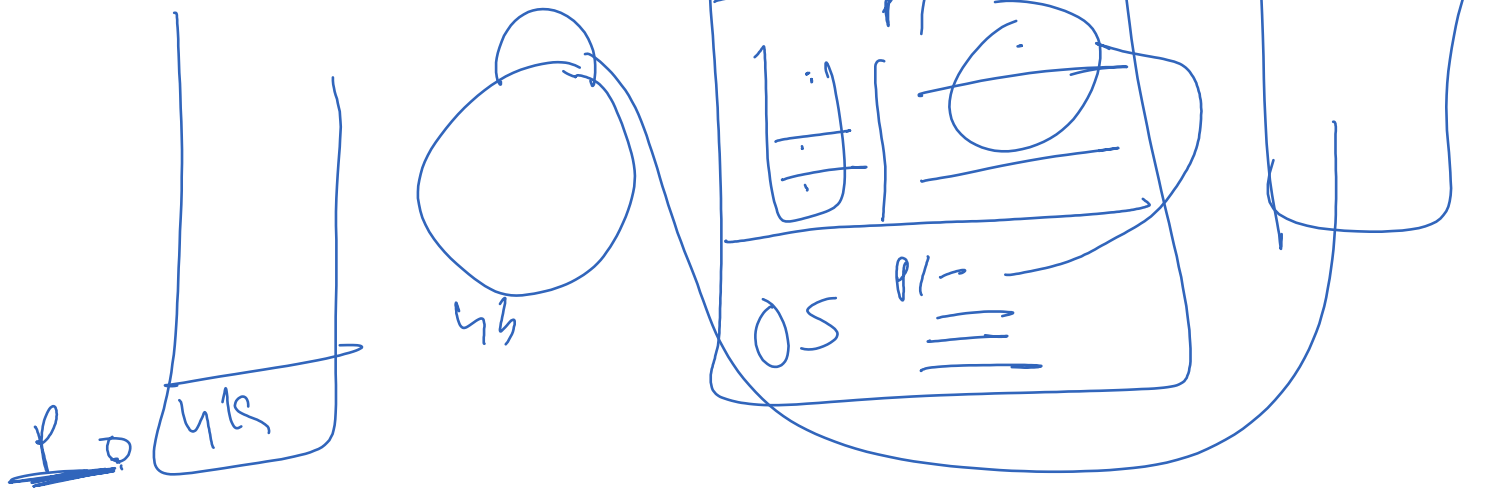
01. no ≤ 7
02. no ≤ 8





4:52 - 4:57

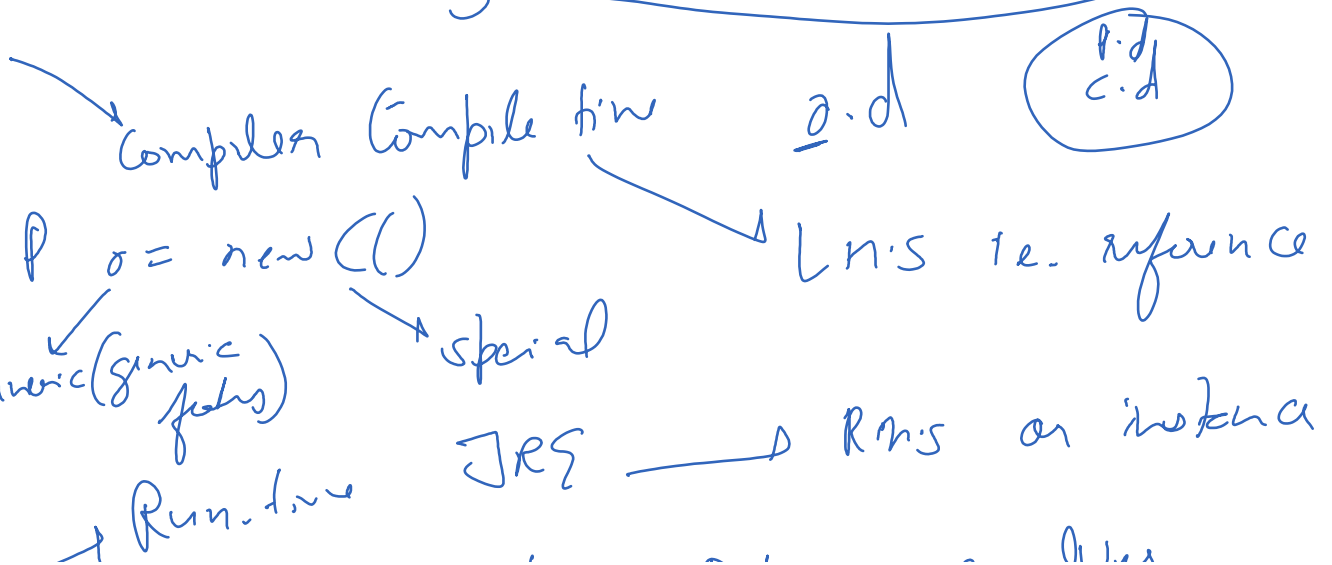
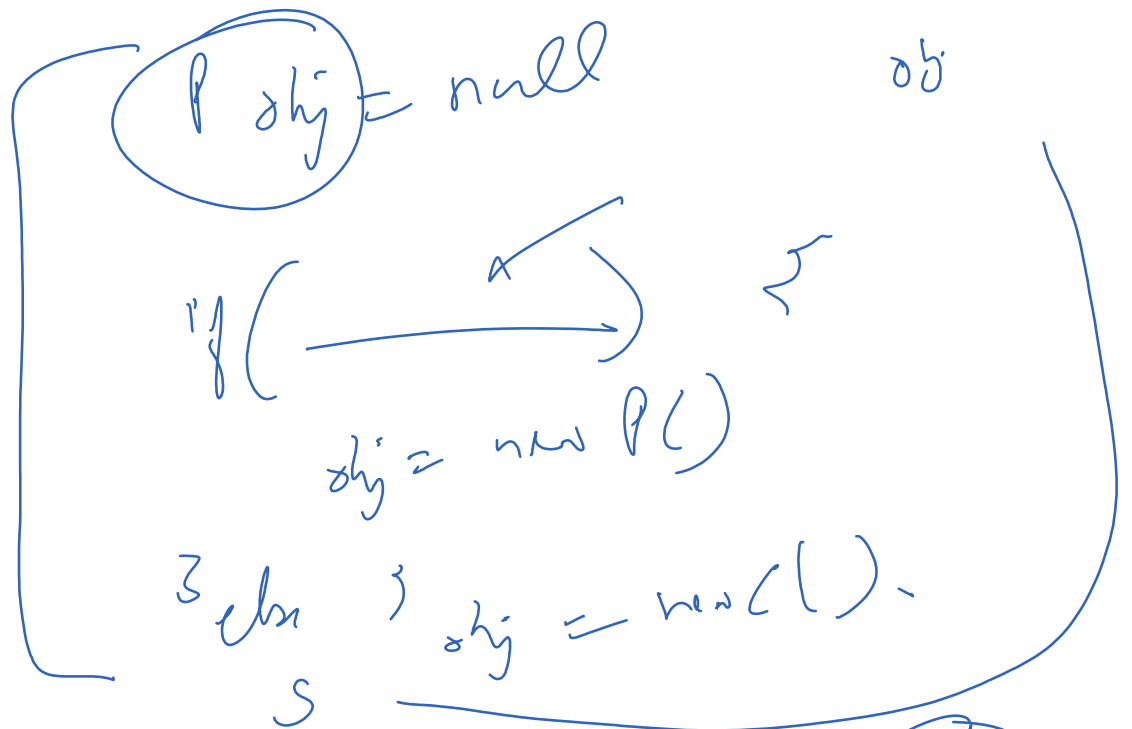
P0 = new C



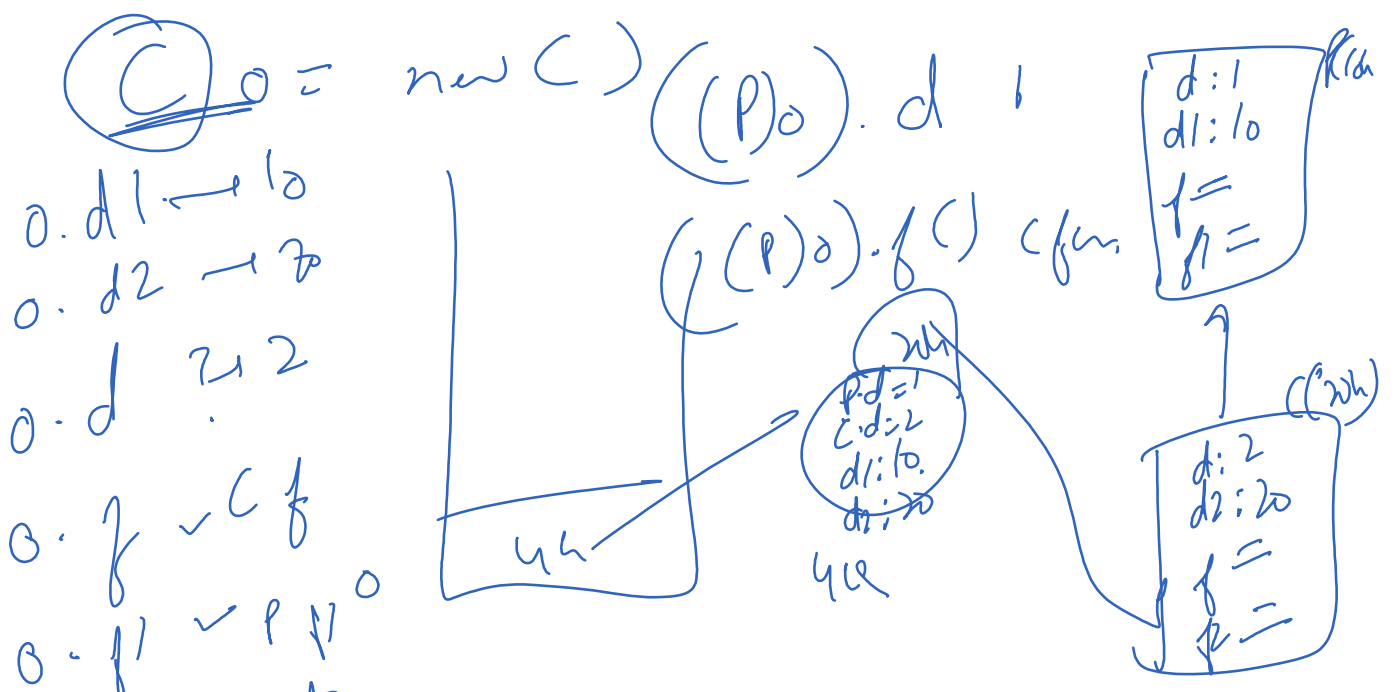
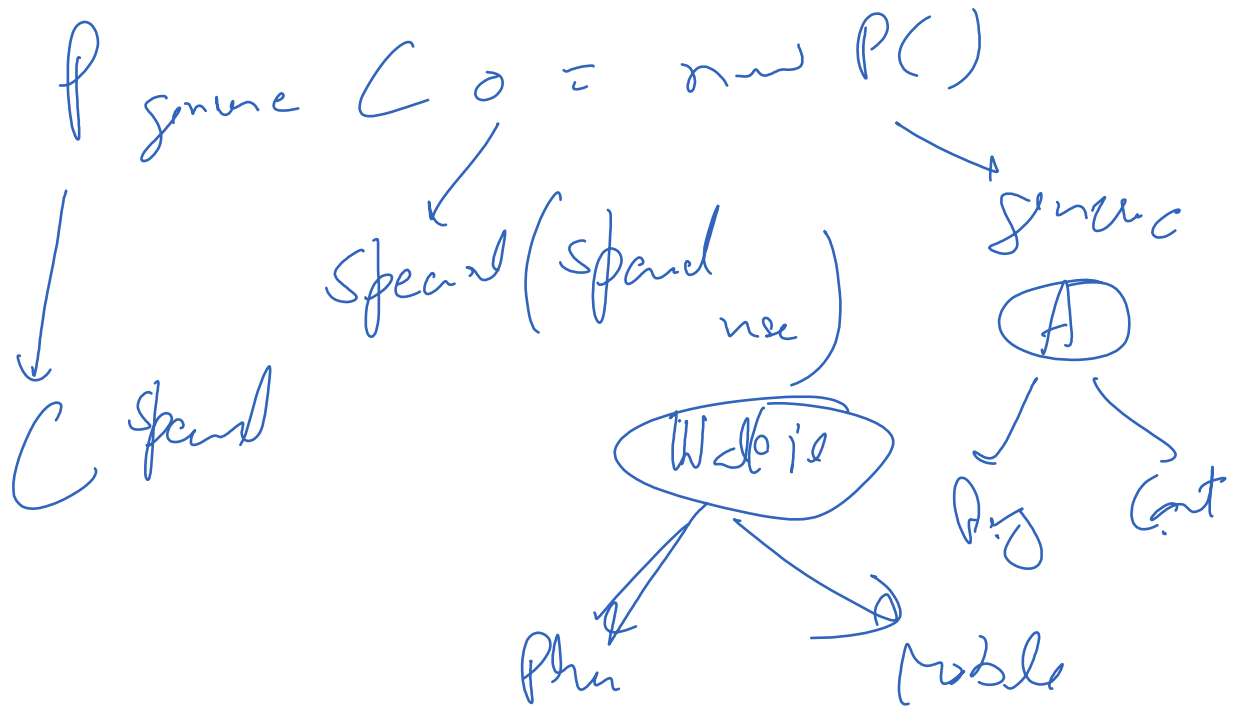
P 0 [4h]

(C) 0 d2
(1) 0 - P(L)

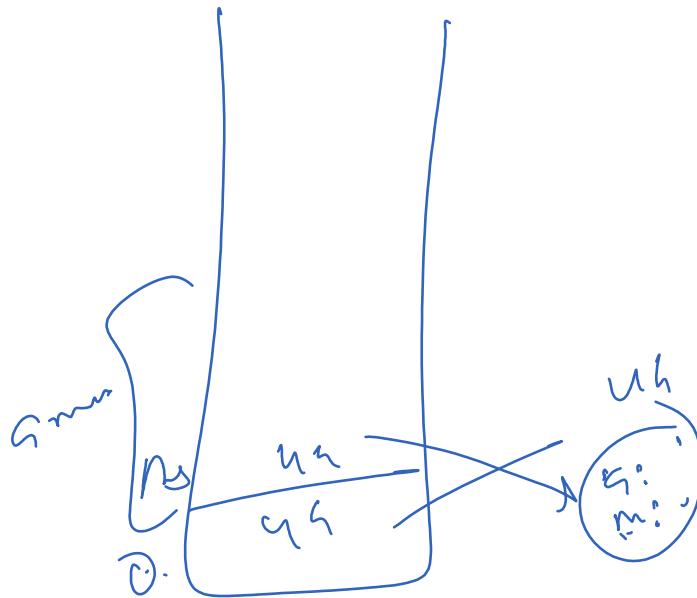
2m
d:2
d:2
d:2



→ conflict → Reference resources



a. $\beta \cdot \checkmark c \beta$



i cum
i m
S: wi-
S dwi()P
wi
S
amua()P
am
m