

```

class Time:
    def __init__(self, hours, minutes):
        self.hours = hours
        self.minutes = minutes

    def displayTime(self):
        print(self.hours, ' hour and ', self.minutes, ' minutes')

    def displayMinute(self):
        total_minutes = self.minutes + self.hours * 60
        print(total_minutes)

    @staticmethod
    def utility(minutes):
        x = minutes // 60
        y = minutes % 60
        return (x, y)

    def addTime(self, other):
        newObjectRef = Time(0, 0)
        x, y = Time.utility(self.minutes + other.minutes)
        newObjectRef.hours = self.hours + other.hours + x
        newObjectRef.minutes = y // 10
        return newObjectRef

```

Diagram illustrating object creation and addition:

- Two objects, obj1 and obj2, are created from the Time class.
- obj1 is initialized with hours=2 and minutes=50, represented by a box labeled "2 50".
- obj2 is initialized with hours=1 and minutes=20, represented by a box labeled "1 20".
- The code shows obj1.addTime(obj2) to calculate the sum of the two times.
- The calculation involves adding the minutes (50 + 20 = 70) and then dividing by 60 to find the total hours (1) and the remaining minutes (10), which is then converted back to a Time object (1 10).

Handwritten annotations:

- obj1 = Time(2, 50)
- obj2 = Time(1, 20)
- obj1.addTime(obj2)

4,10

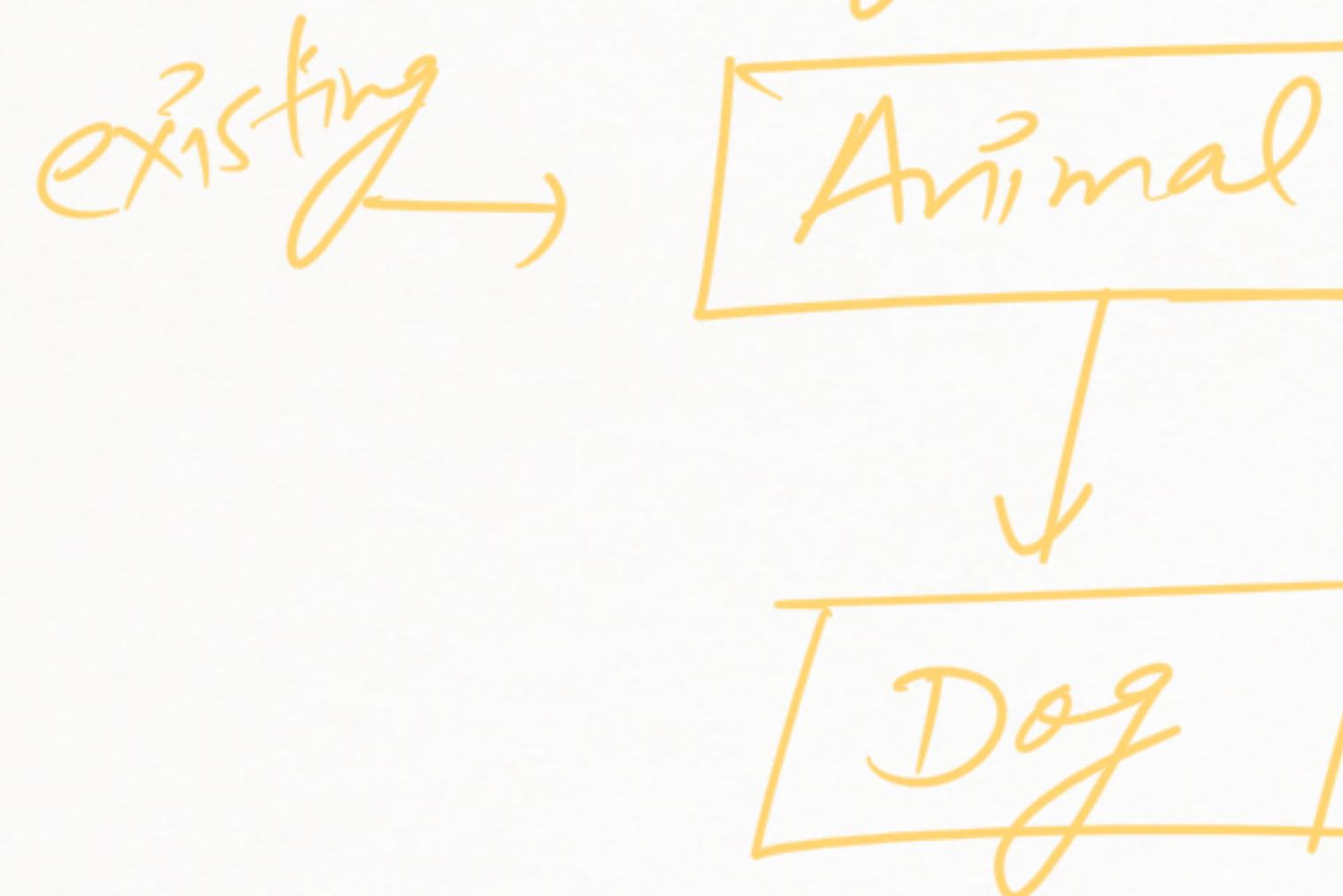
newObj

4 hour

10 min

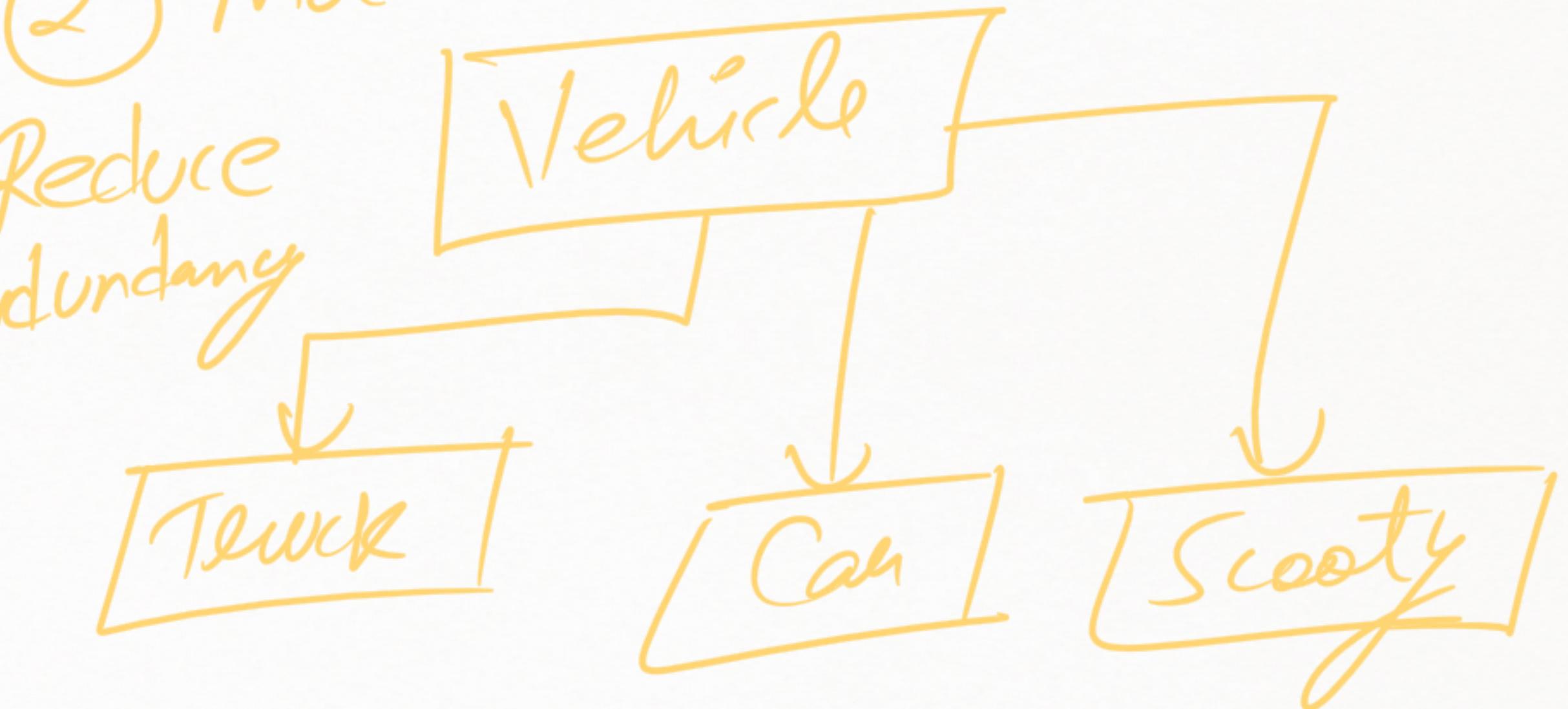
object
method

Inheritance
→ Acquiring properties & methods from
Parent/existing classes.

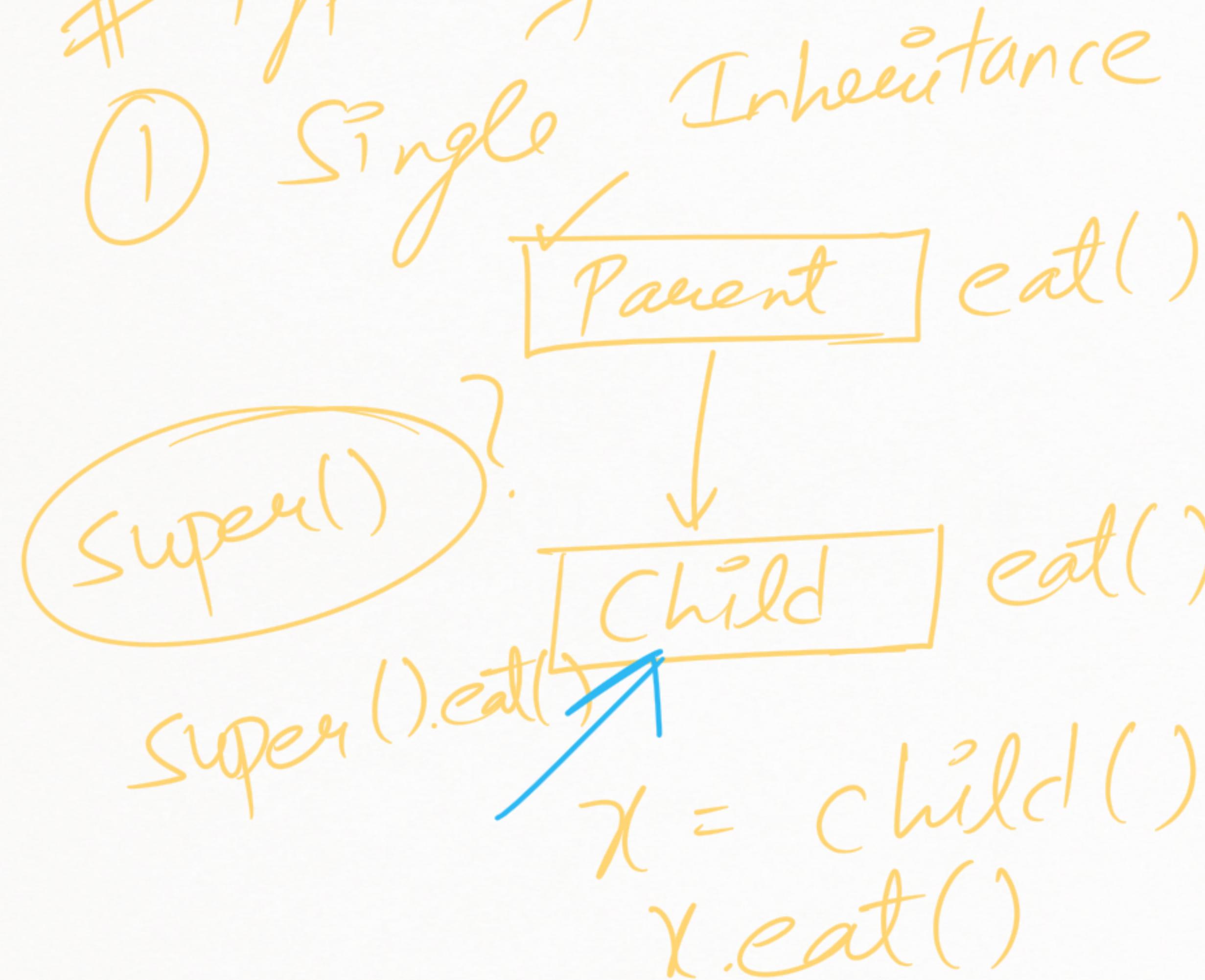


Benefits:

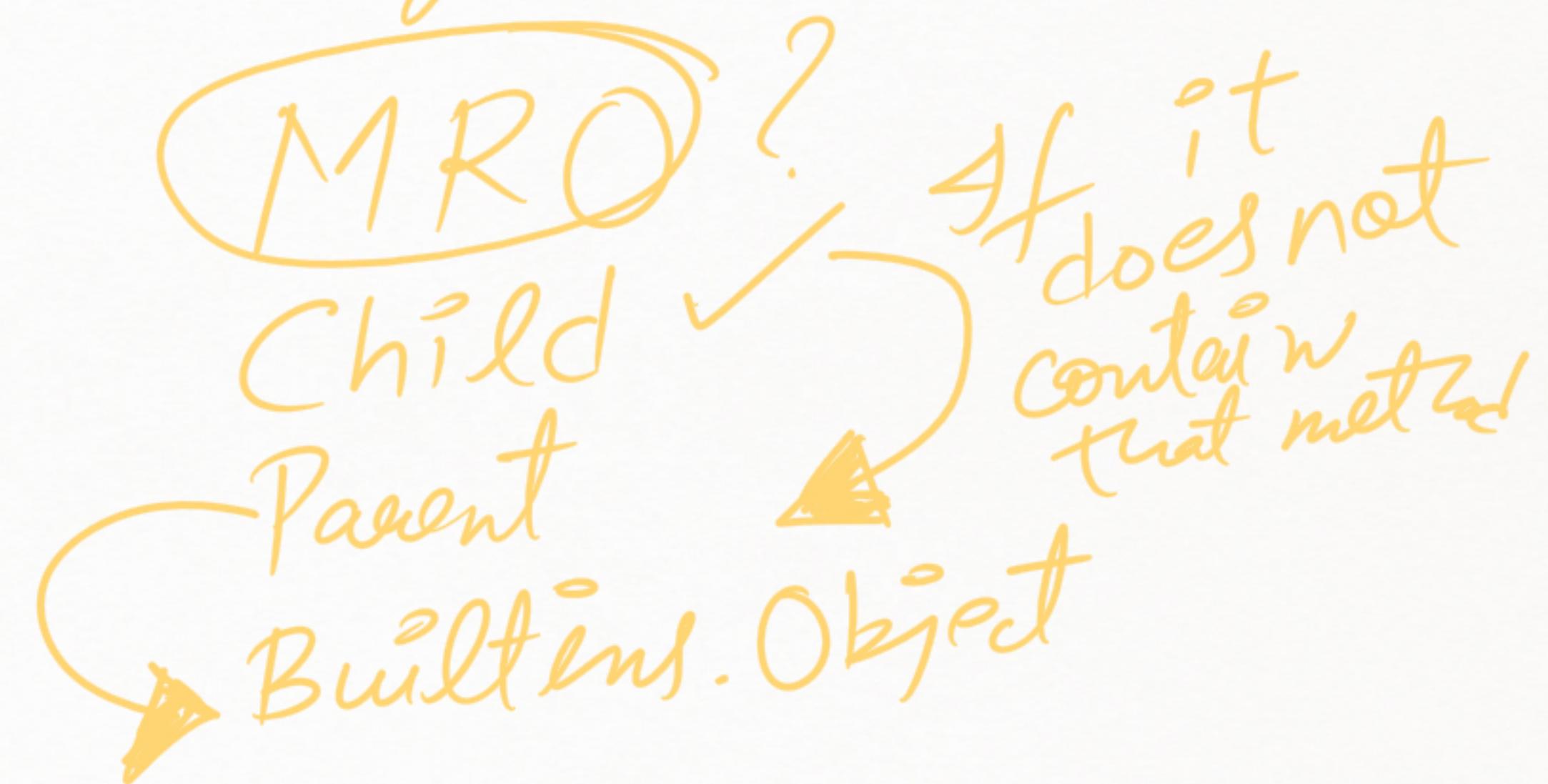
- ① Code reusability
- ② Modular
- ③ Reduce redundancy



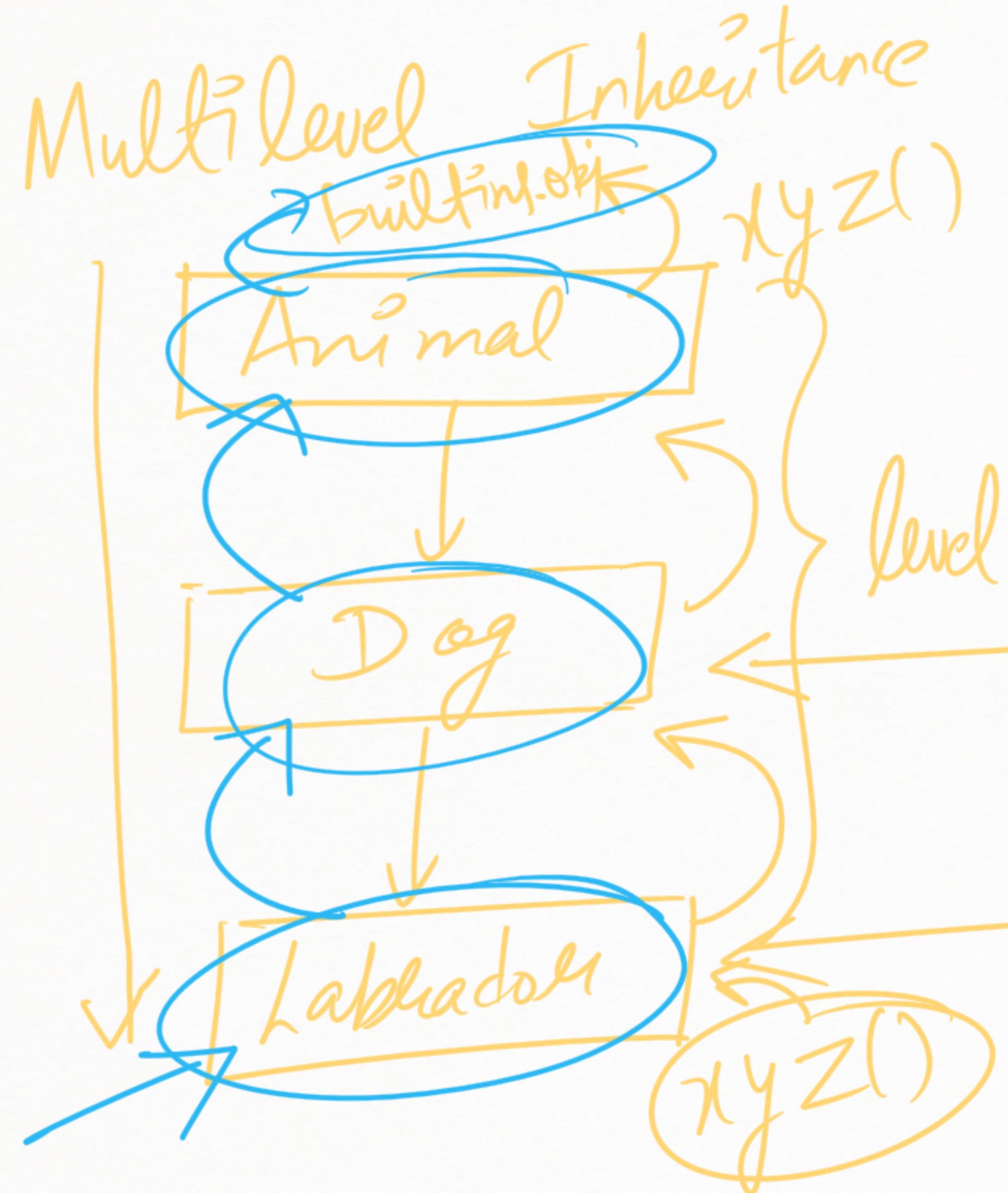
Types of Inheritance



Method Overriding
in Single Inheritance ✓



Q



MRO

Lab ✓

Dog ✓

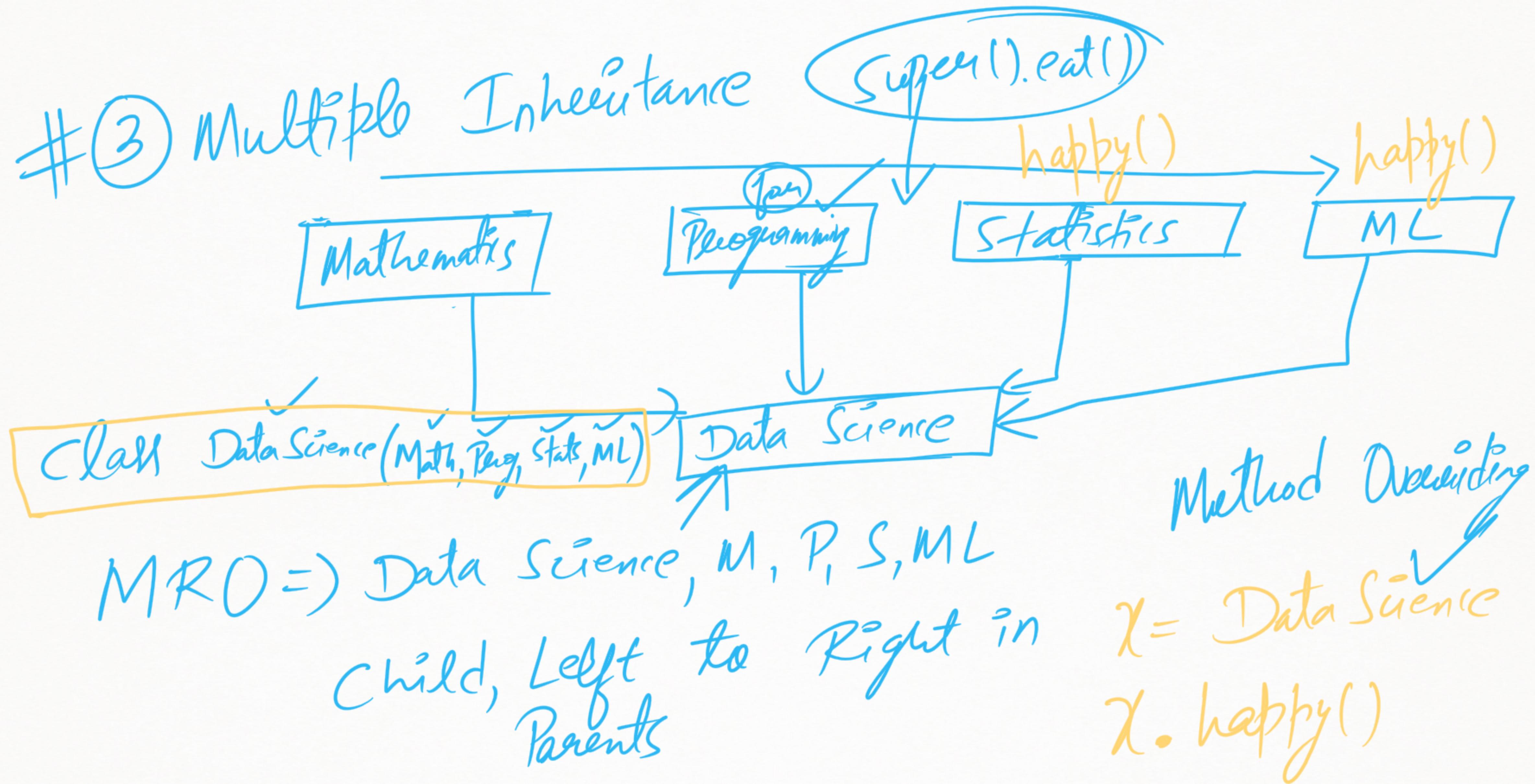
Animal ✓

builtins object. ✓

super().eat()

super().eat()

#③ Multiple Inheritance

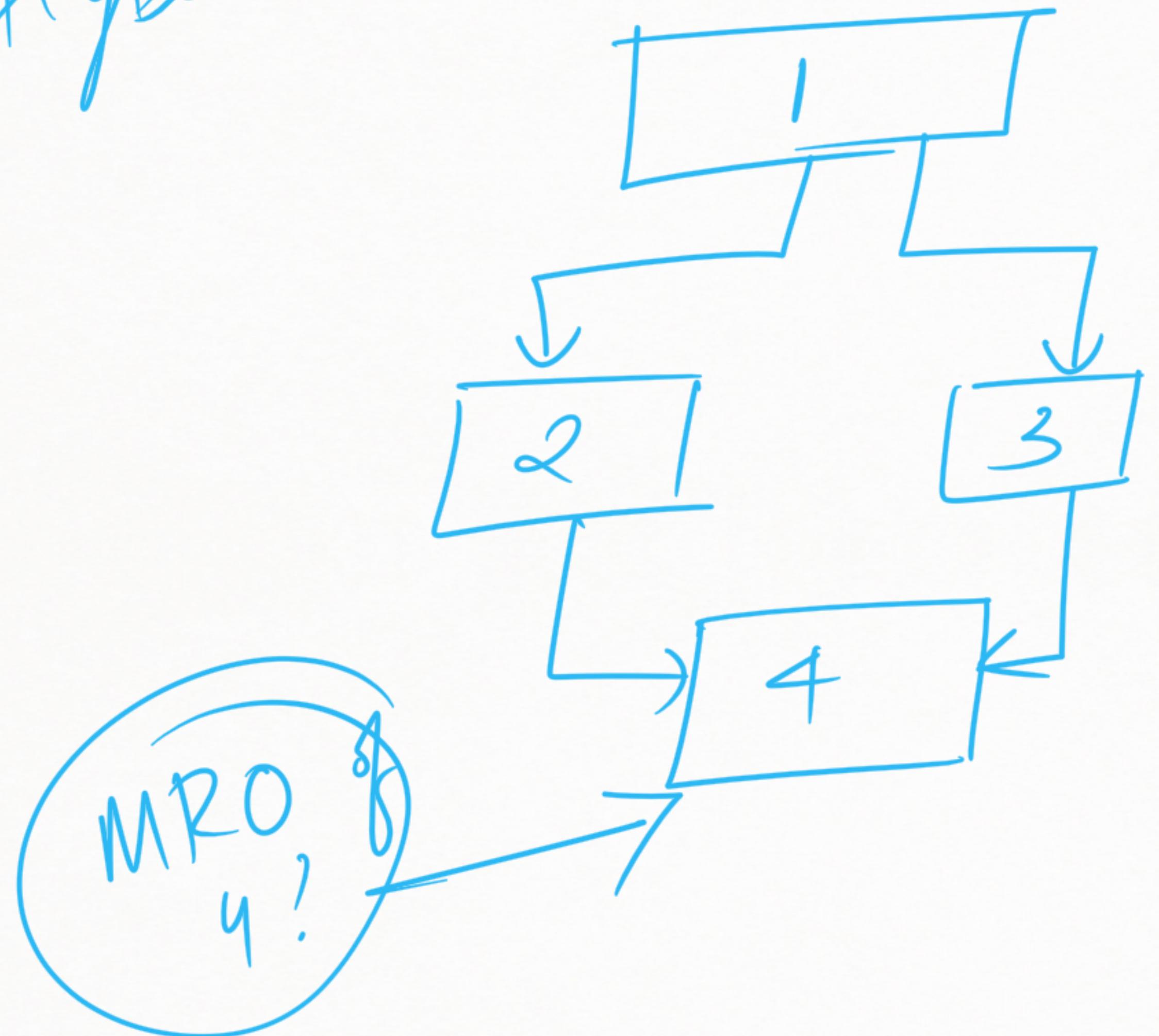


④ Hierarchical Inheritance

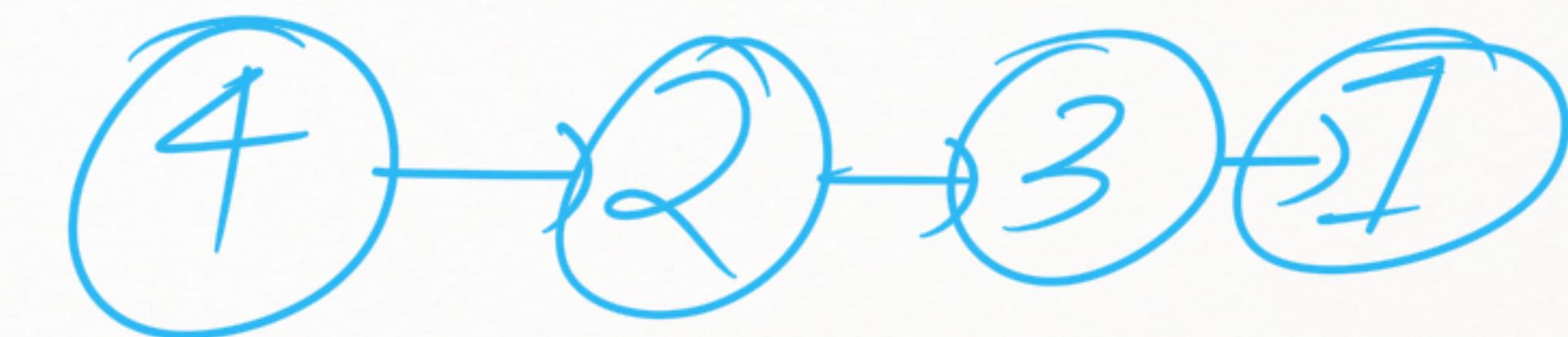


5

Hybrid Inheritance

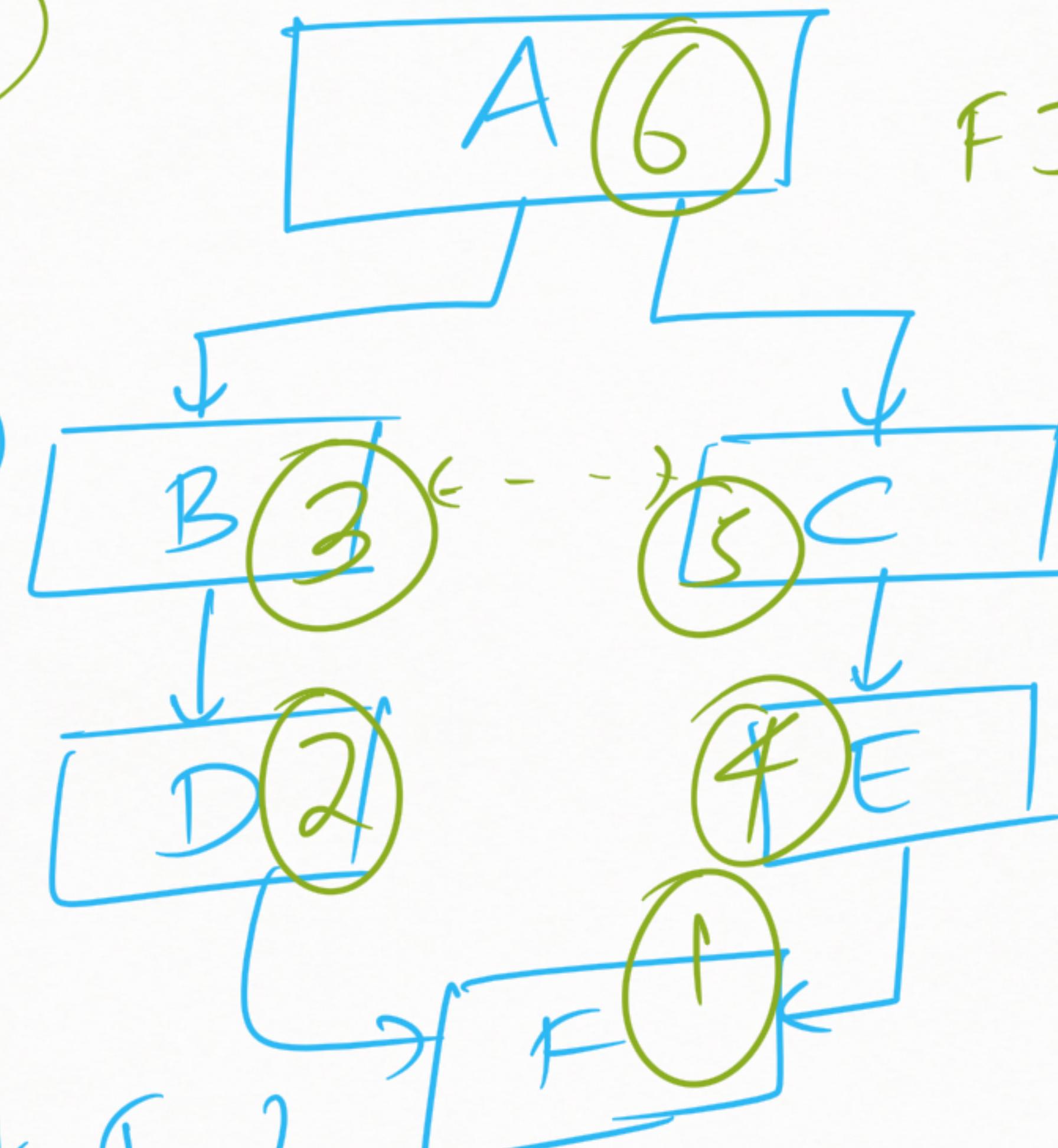


Bottom to top, left
to right



Depth First

$E(C)$
 $C(A)$
 $F(D, E)$
 $D(B)$
 $B(A)$



F D B E C A ✓

MRO of F?