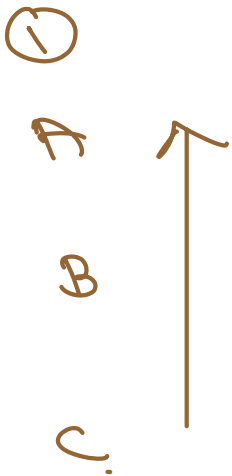
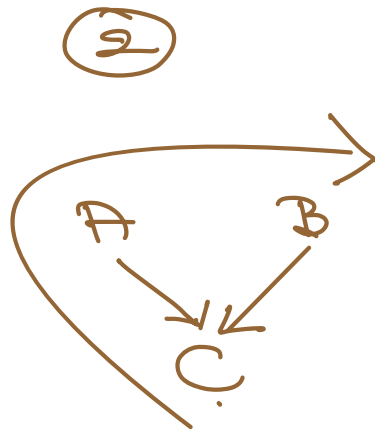


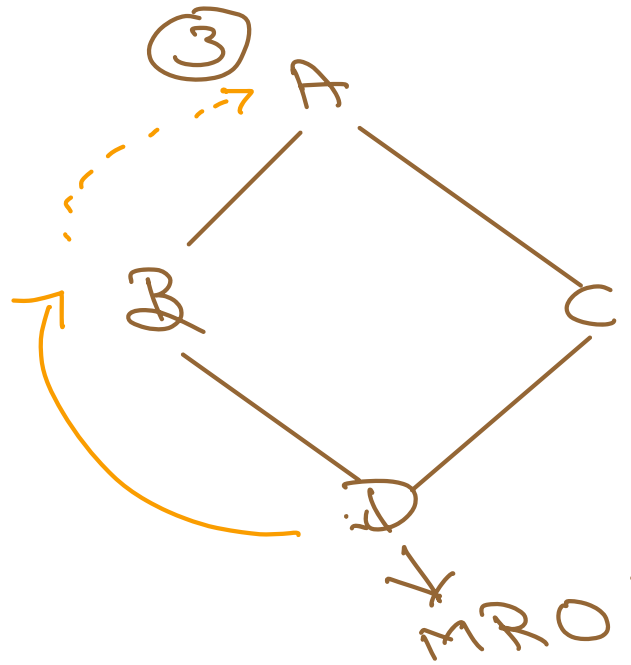
MRO



$C \rightarrow B \rightarrow A$



$C \rightarrow A \rightarrow B$



$D \rightarrow B \rightarrow C \rightarrow A$

* You can't go to a parent class until all its children are already visited

```
class A
    def method(*)
```

==

```
class B
    def func2()
```

No
Self

either

① $self().methodname(*)$

② $A.methodname(self, *)$



Whenever object of Class is created :-

- * new is called first and is responsible for creating object
- * init takes object create by new and initializes variable on that

Relationship in Python

- 1) Inheritance \rightarrow is a
- 2) Composition \rightarrow has a
- 3) Association \rightarrow uses a

B (A)

```
class A
    par
```

* Has a

```
class B
```

```
    def __init__(self, a)
        self.a = a
```

```
    def method(self, c)
```

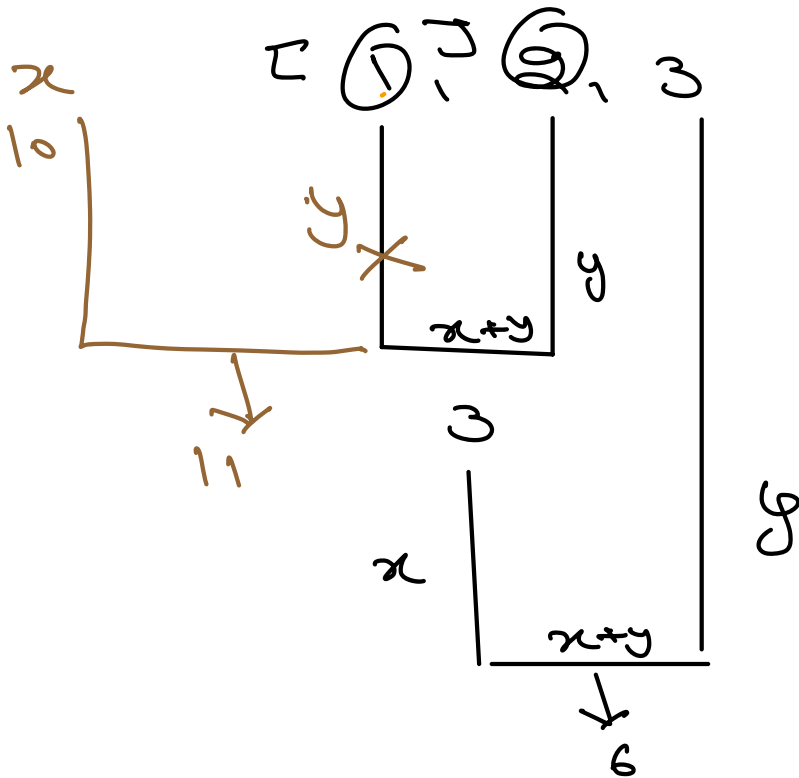
```
        print(c)
```

class C

*

* Uses a

reduce (lambda x,y : x+y, [1,2,3], 10)



reduce (lambda P, x : P + [subset | x in P]
for subset in P]
[set ()]

array > { 1, 2, 3 }

P > [{ 3 }]
x > 1
↓

$P + [\text{sub} \mid \Sigma x, 3] \text{ for sub in } P$

\downarrow
 $\Sigma 3 \mid \Sigma 3$
 \downarrow
 $[\Sigma 3] + [\Sigma 1, 3]$

$P \ni [\Sigma 3, \Sigma 1, 3]$
 $\downarrow \Sigma 23$
 $x \ni 2$

\downarrow
 $[\Sigma 3 \mid \Sigma 23, \Sigma 13 \mid \Sigma 23]$

$P + [\Sigma 23, \Sigma 1, 23]$

\downarrow
 $[\Sigma 3, \Sigma 13, \Sigma 23, \Sigma 1, 23]$

\downarrow
 $x \ni 3$

$P + -$

$\tau \in 3, \tau^1 3, \tau^2 3, \tau^1, \tau^2, \bullet \rightarrow \text{old } p$

$\tau 33, \tau^1, 33, \tau^2, 33, \tau^1, \tau^2, 33 \rightarrow \text{new}$