Agenda

- Dunder Methods
- Inheritance
- Access Specifiers
- Multiple Inheritance
  - MRO : Method Resolution Order

class A:

===

obj = A()

```
class A:
    print("Am I a class?")

A()                          L1

obj1 = A()                   L2   (this will not be GC)

if A():                      L3
    print(A())               L4
```

Object's Truth Value is always True

S

Dunder Method

1. `__init__(self, .....)` → ~~Constructor~~ Initializer
   - initialize instance Variables

2. `__add__(self, object)`
   - a → self
   - b → object

```
class A:
  def __init__(self, a):
    self.a = a
  def __add__(self, b):
    if isinstance(b, int):      L 1
      return A(self.a + b)      L 2
    return A(self.a + b.a)      L 3
  def __str__(self):
    return f"{self.a}"


one = A(1)
two = A(2)
print(one + two + 100)
```

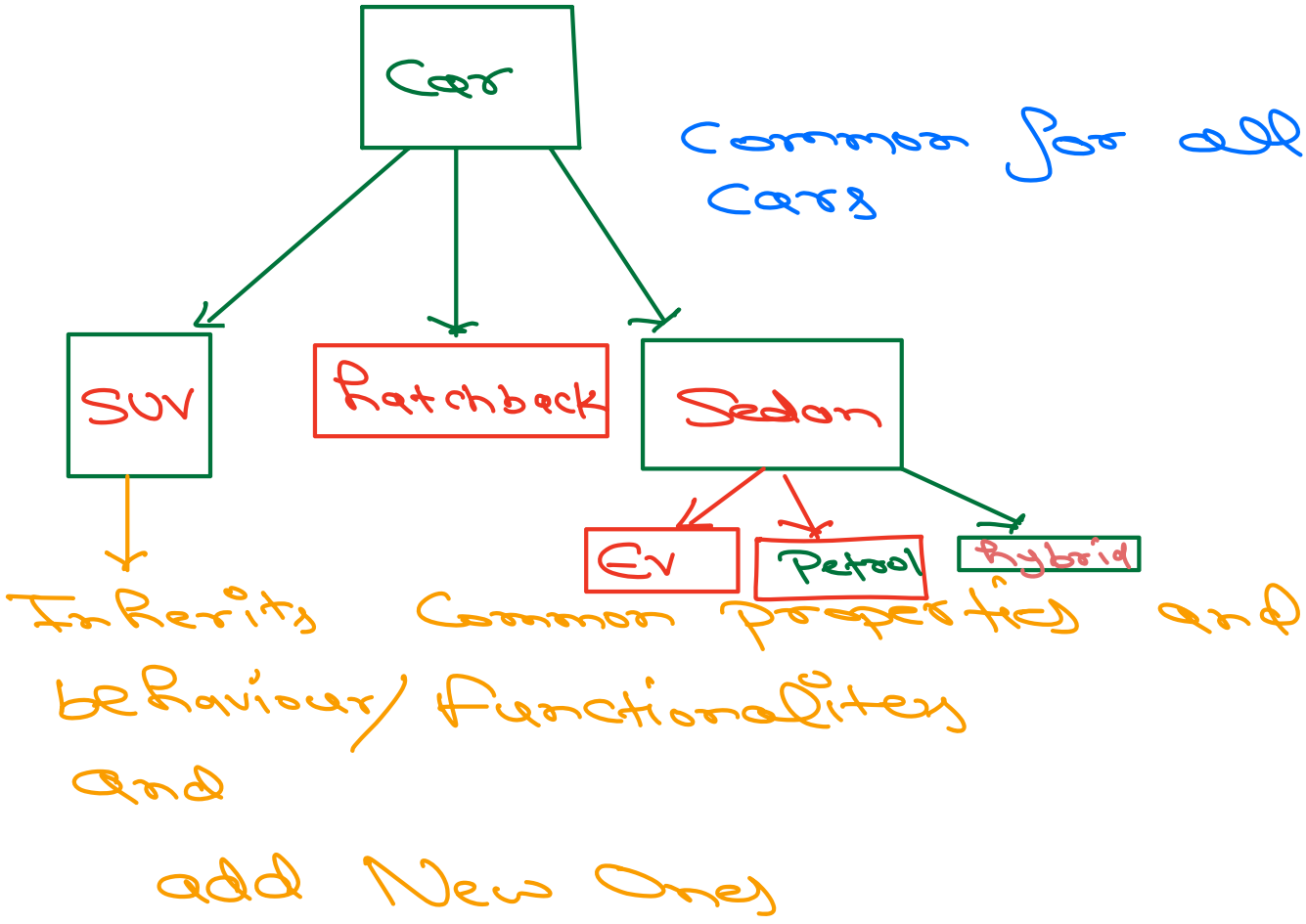
a = one, b = two

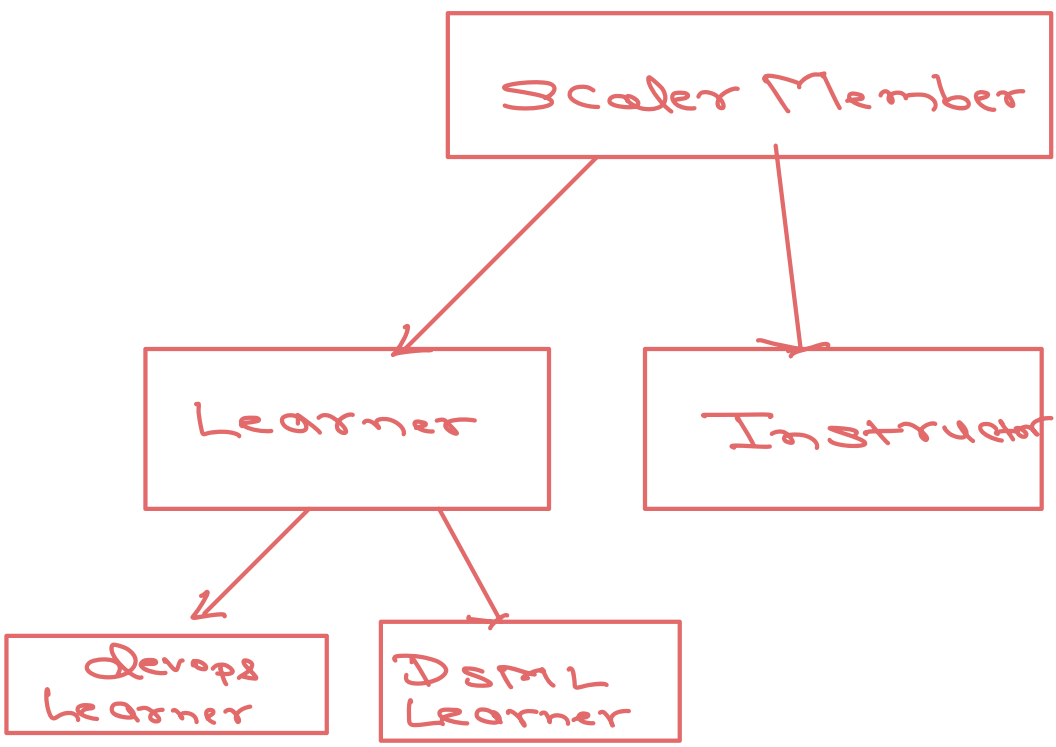
A(3) + 100

A (3+100)

print (A(103)) ⇒ 103

Inheritance





* super().method( )

→ Executes immediate parents method

Ex: super().__init__(....)

↓

Calls parents initializer to initialize common instance Variables

```
class A:
    def do_something(self):
        print("I do something!", end=" ")

class B(A):
    def do_something(self):
        super().do_something()
        print("I also do something!", end=" ")

b = B()
b.do_something()
```

F2

F1

Access Specifiers

* Public ⇒ (default) Can be accessed from anywhere including Outside the class

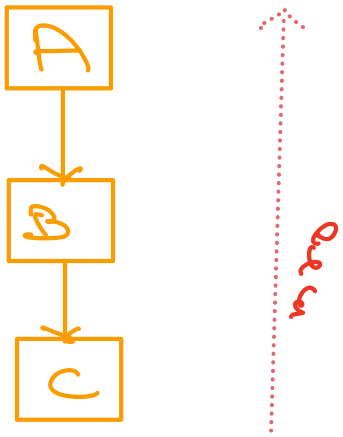
* Private ⇒ Can only be accessed inside the class
  - Variables
  - Methods

⊙ To make any variable as private you can prefix it with __Var_name

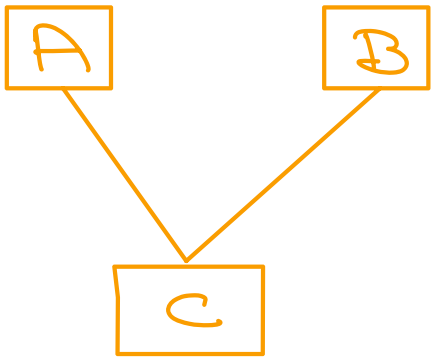
* Name mangling :

__Varname ⟶ _classname__Varname

Multiple Inheritance



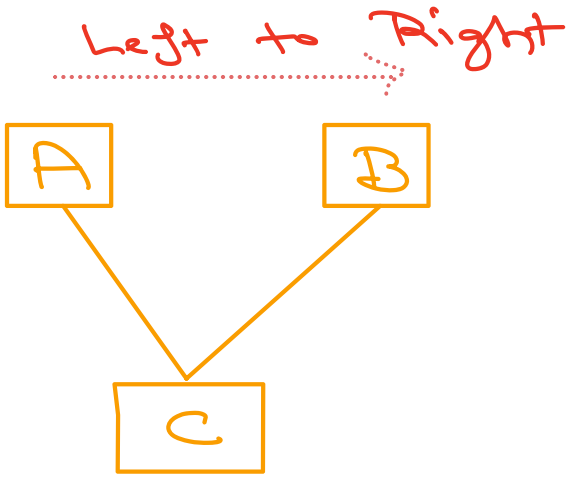
Multi-level Inheritance



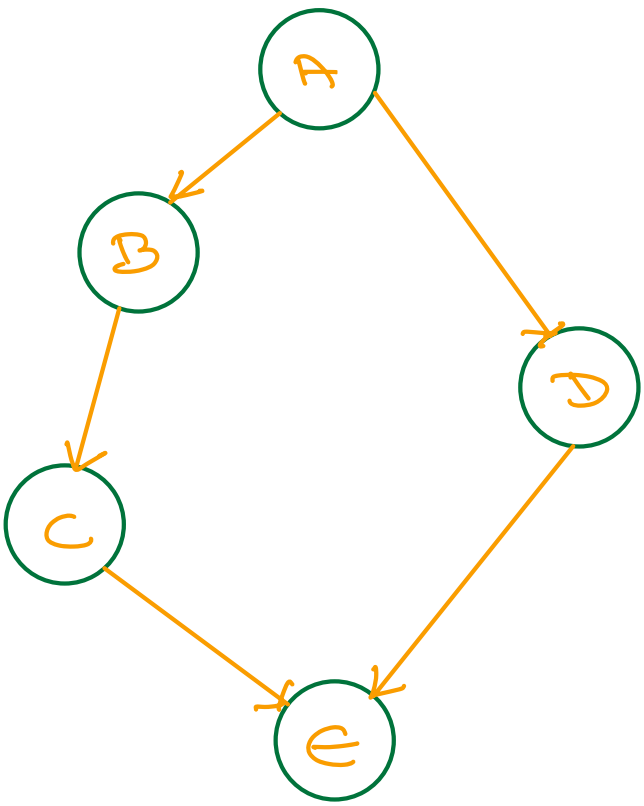
Multiple Inheritance

Flow will be from bottom to Top

MRO : Method Resolution Order



Multiple Inheritance



Diamond Inheritance

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

E1 • X

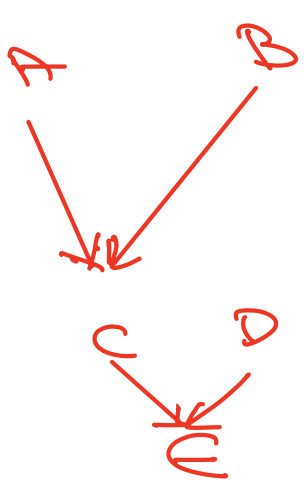
mro

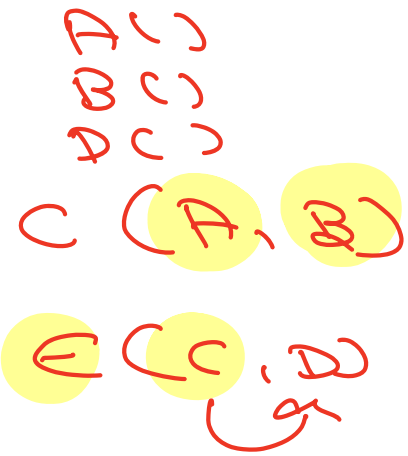
E → C → B → D → A

① * Left to Right

② We go to a parent Only when all its childs have been accessed

E → C —





E → C → A → B → D