4 pillars of OOP

1) Encapsulation → Packing contents in Entity

```
Class A:
    # methods
    # attribute
```

```
a1 = A()
↑
entity
```

Access specifier:
Control the access of attribs and Methods
- public (default)
- protected
- private

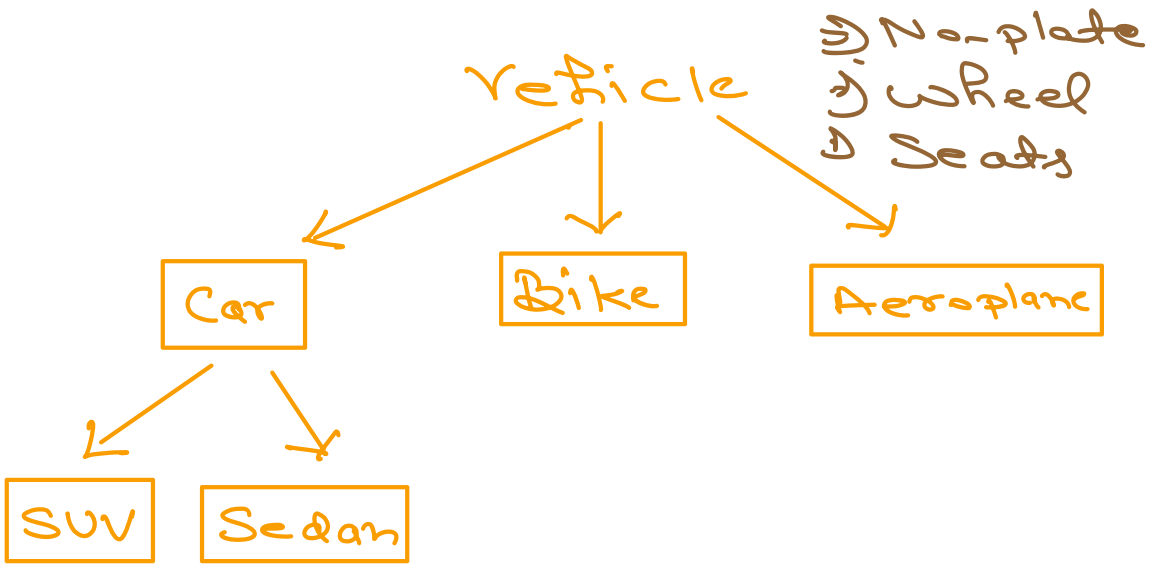
1) Private Vars are Not inherited
There is No such thing as truly private in Python
2) Hiding Data

1) Abstraction:
   1) Hide irrelevant data
   2) Hiding implementation

1) abstract Class

- Polymorphism
  - Same entity different behaviour

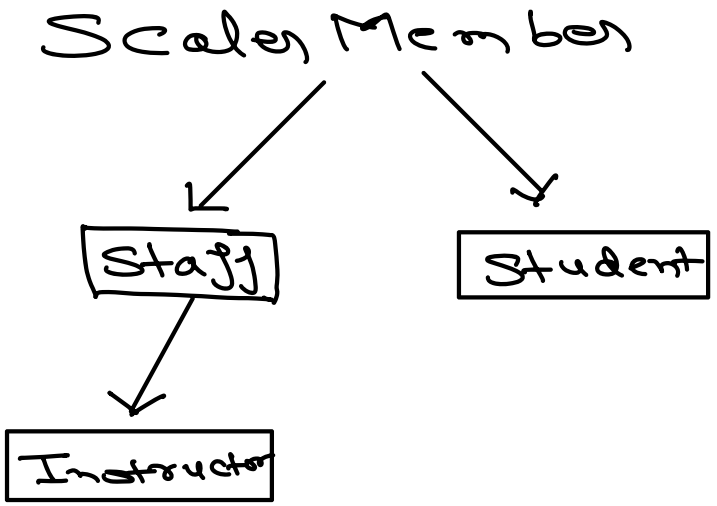
- InHeritance:
  - to avoid redundancy
  - To make code more maintainable



Without InHeritance

```
class SUV:
    def __init__(self,1,2,3)
        self.wheel = 4
        self.no_plate = 2
        self.seats = 3
```

```
class Car :
    def __init__(self,1,2,3)
        self.wheel = 4
        self.no_plate = 2
        self.seats = 3
```

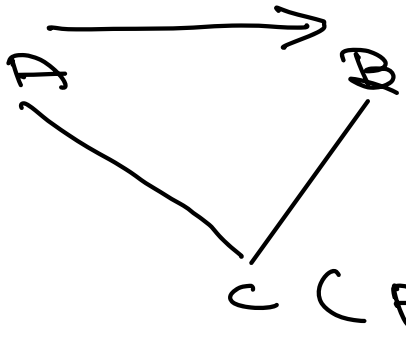


A
↑
B
↑
C

C → B → A

Multilevel Inheritance

Multiple Inheritance



C → A → B