

Object Oriented Programming

L = [1, 2, 3]

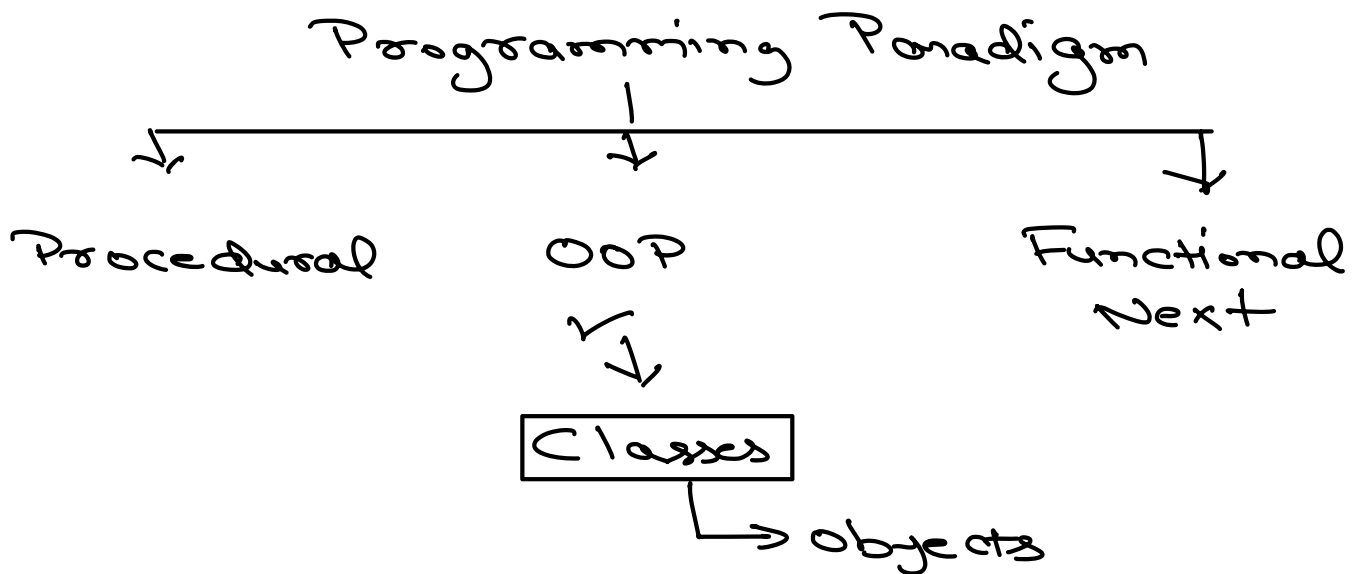
L.count(2)

↑
2

Class List():

count
--iter

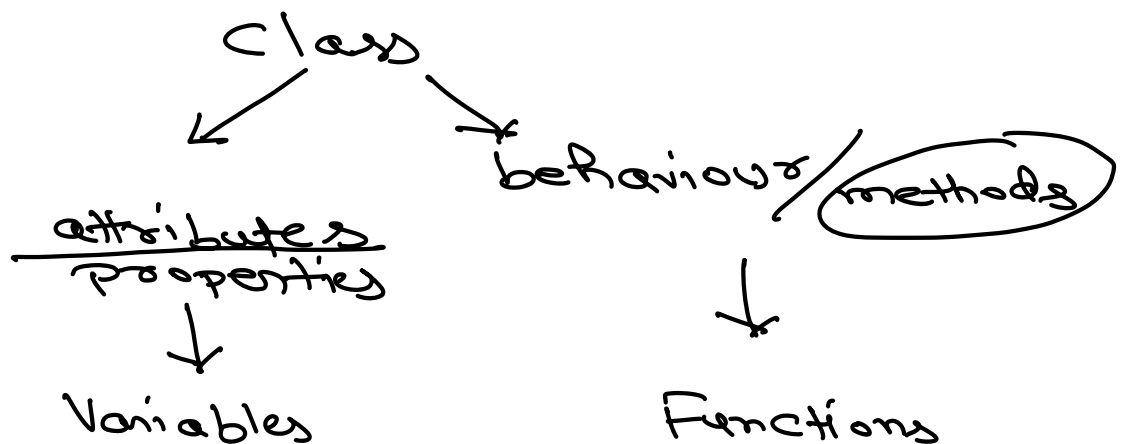
l = list() or []



Class

→ blueprint

→ Collection of similar attribute and behaviours



```
class Student:
```

```
    name
```

```
    id
```

```
    batch
```

```
    def watch_lecture():
```

```
        _____  
        _____  
        _____
```

```
s1 = Student('Neethu', 101, 'Aug23')
```

S2 = Student ('Aryan', 102, 'Mar23')

↓
instance of student class
↓

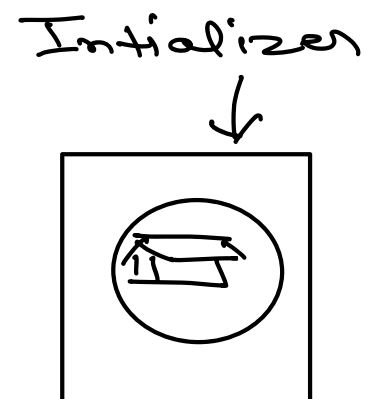
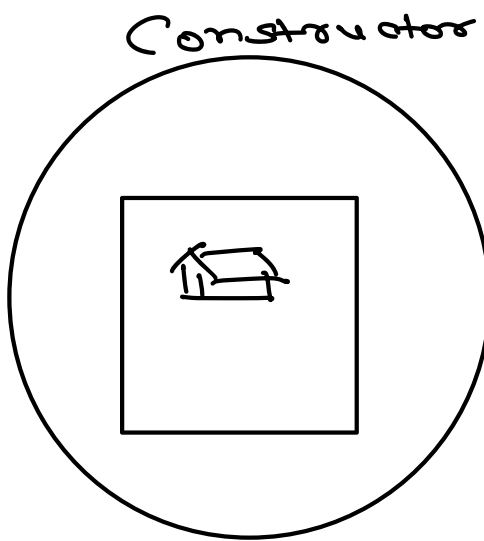
* Instance Variables are specific to an instance

Constructor

- It assigns memory
- Initializes instance vars

Initializer:

- Just initializes vars



-- Method -- \leftarrow inherited from object
 \leftarrow dunder Methods

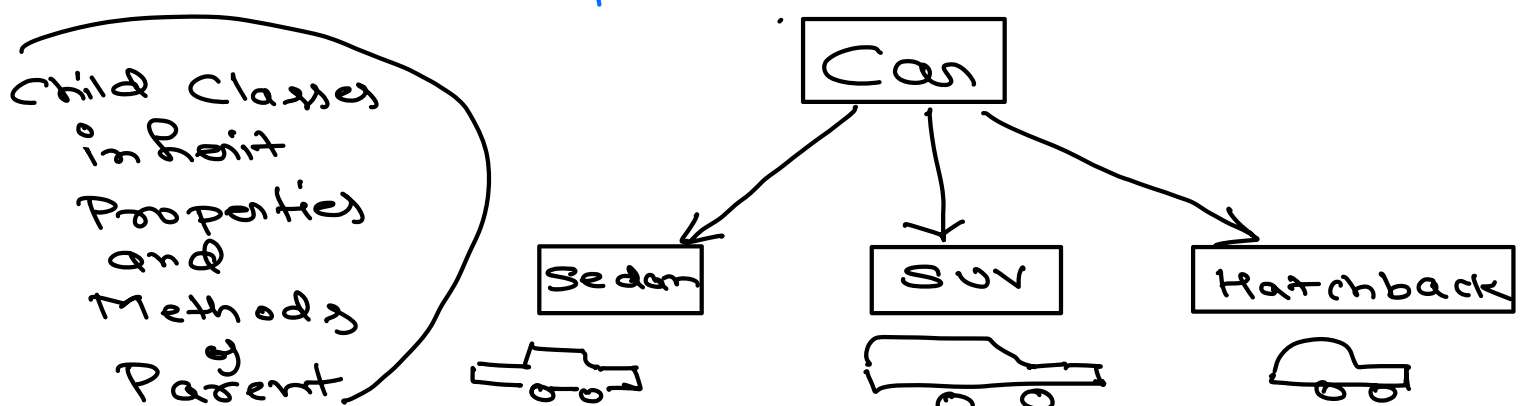
* `--init--` : It's special dunder method that gets invoked automatically
 \Rightarrow Performs tasks after object is created

4 pillars of OOP
 \Rightarrow Polymorphism:

\Rightarrow Same Entity different behaviour

\Rightarrow +

\Rightarrow Inheritance



⇒ Hiding Information

⇒ Abstraction :

⇒ Implementation details

⇒ General Template to create classes

↳ Abstract Methods

⇒ Encapsulation :

⇒ Packing data and methods into single

Entity

⇒ Provides mechanism to prevent data modifications

⇒ Access Specifier

Hides Data

