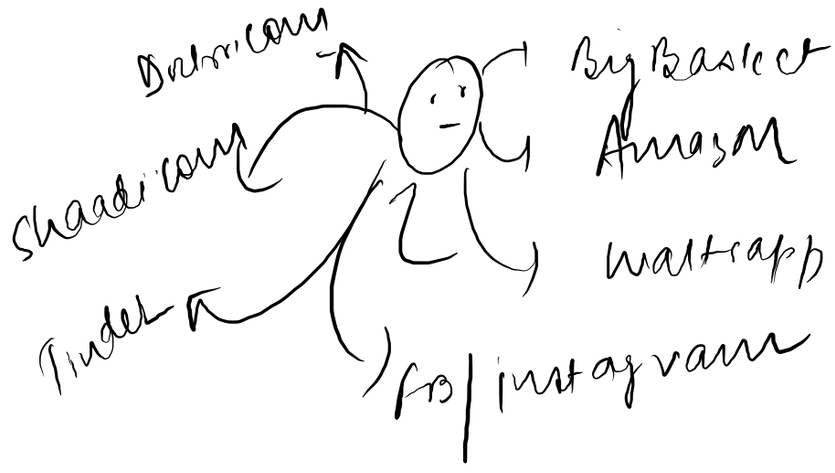
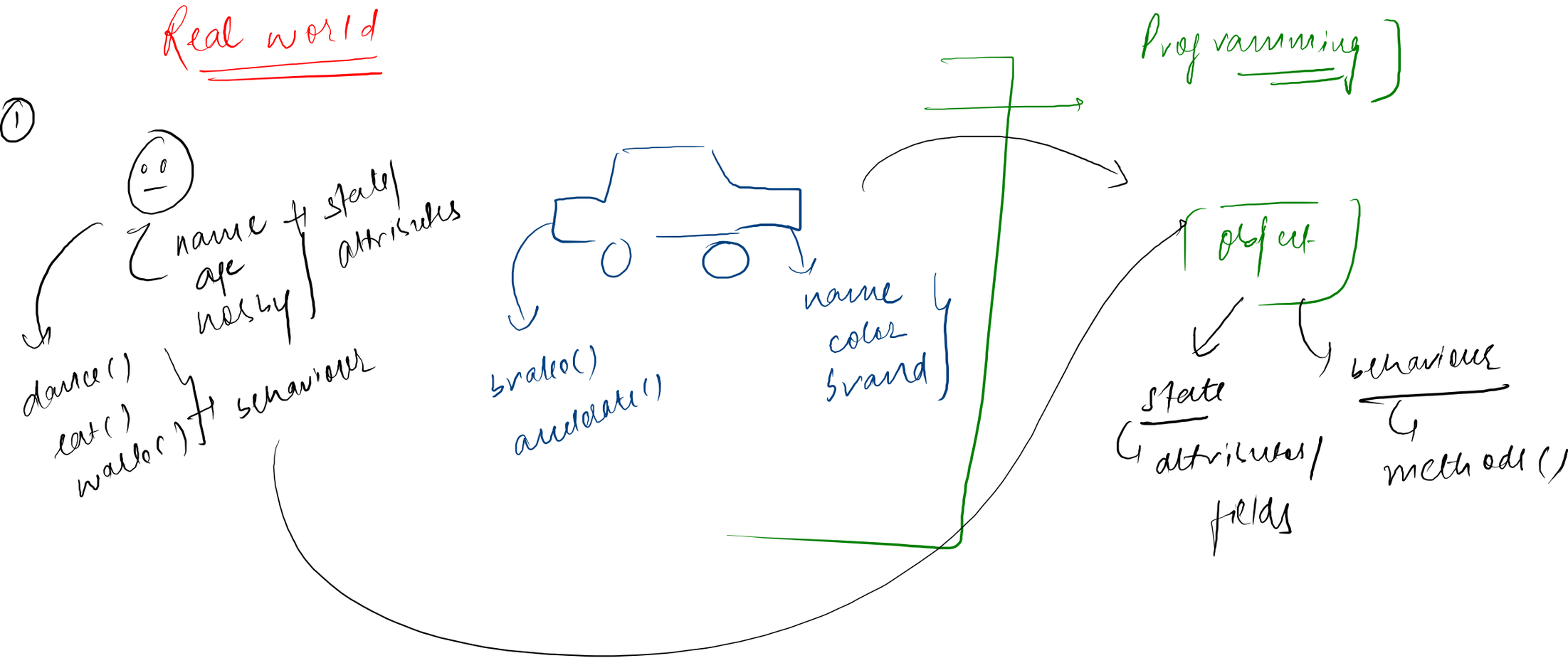


90% of the companies

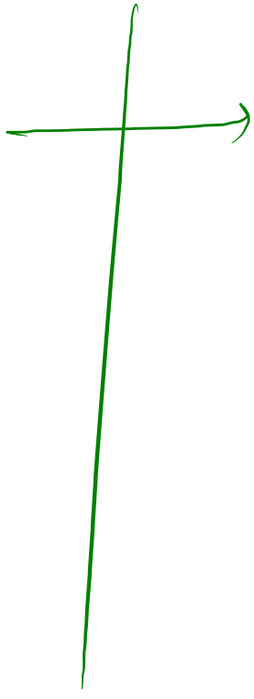
Products

Real world


Real world
Programming
1
name
age
nesty
state/
attributes
dance()
eat()
walk()
behaviours
brake()
accelerate()
name
color
brand
object
state
attributes/
fields
behaviours
methods()

Real world

↳ Relationships



Programming

↳ OOPs concept

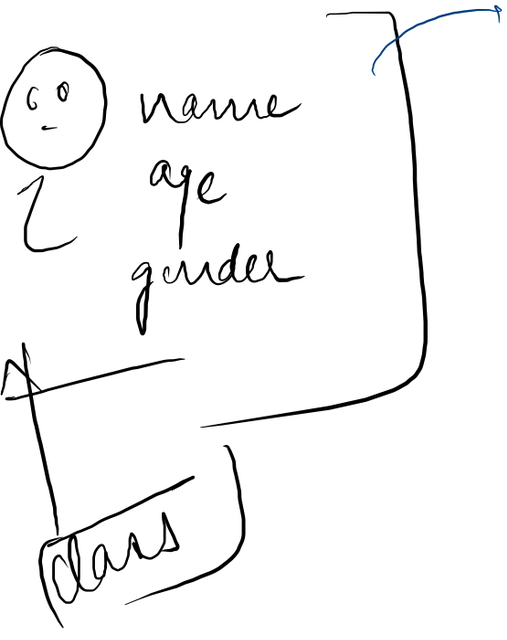
Class and Object

Class → type

defined as a blueprint

↓

specific instance of that blueprint

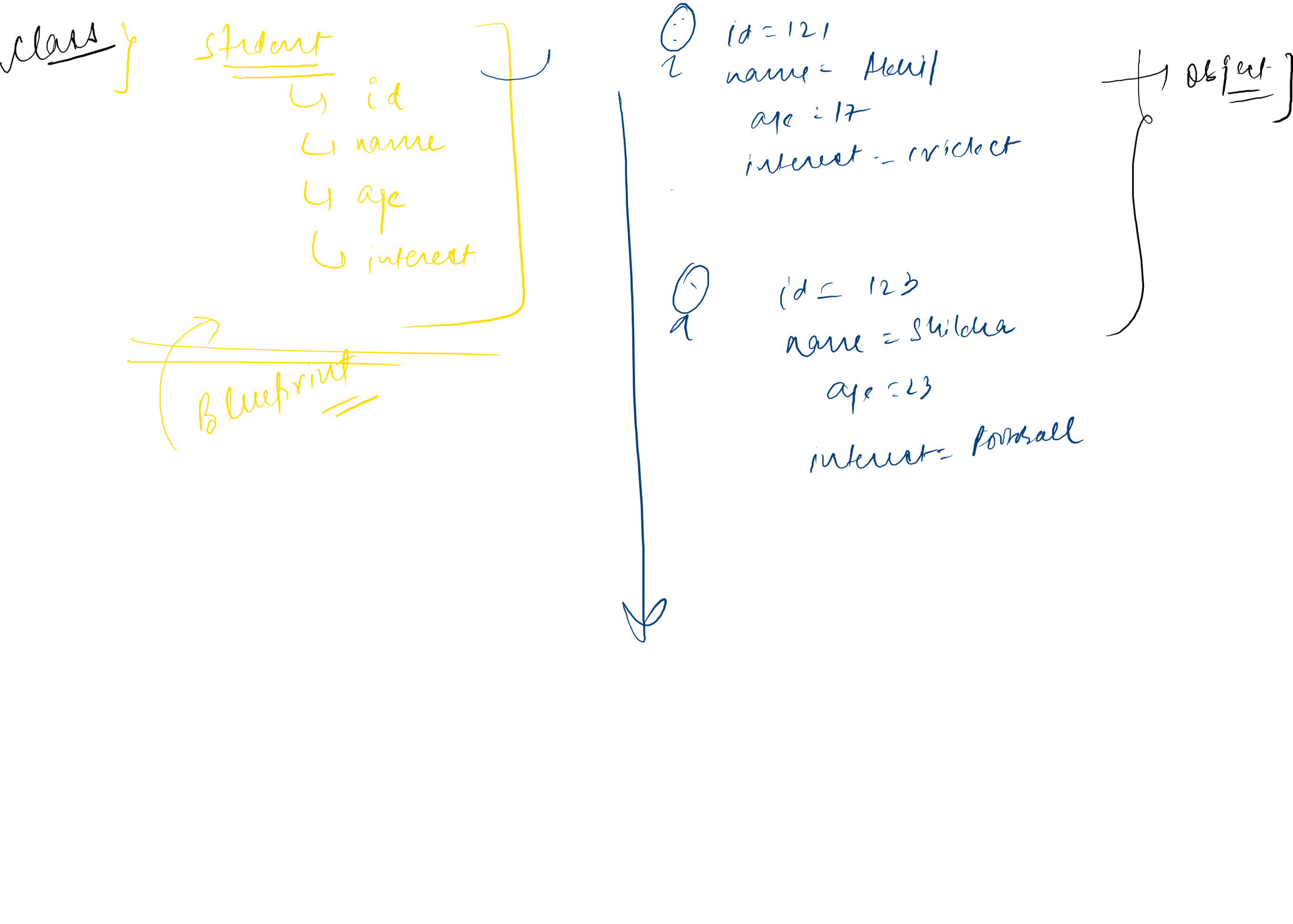


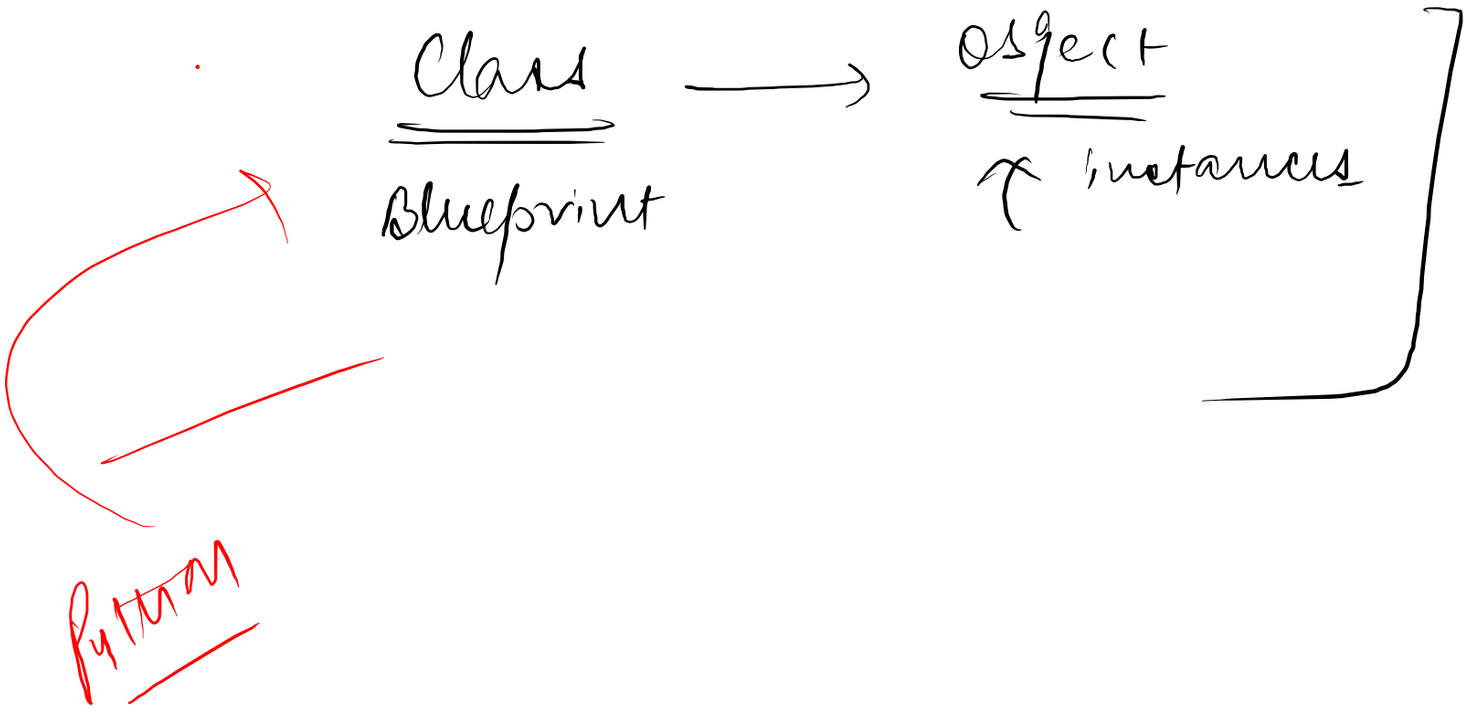
name = Vrishna
age = 91
gender = M

object → instances of the class

name = Lavanii
age = 95
gender = F

object


class
student
id
name
age
interest
Blueprint
id = 121
name = Akhil
age = 17
interest = cricket
id = 123
name = Shikha
age = 23
interest = football
object


Class
Blueprint
Object
instances
Python

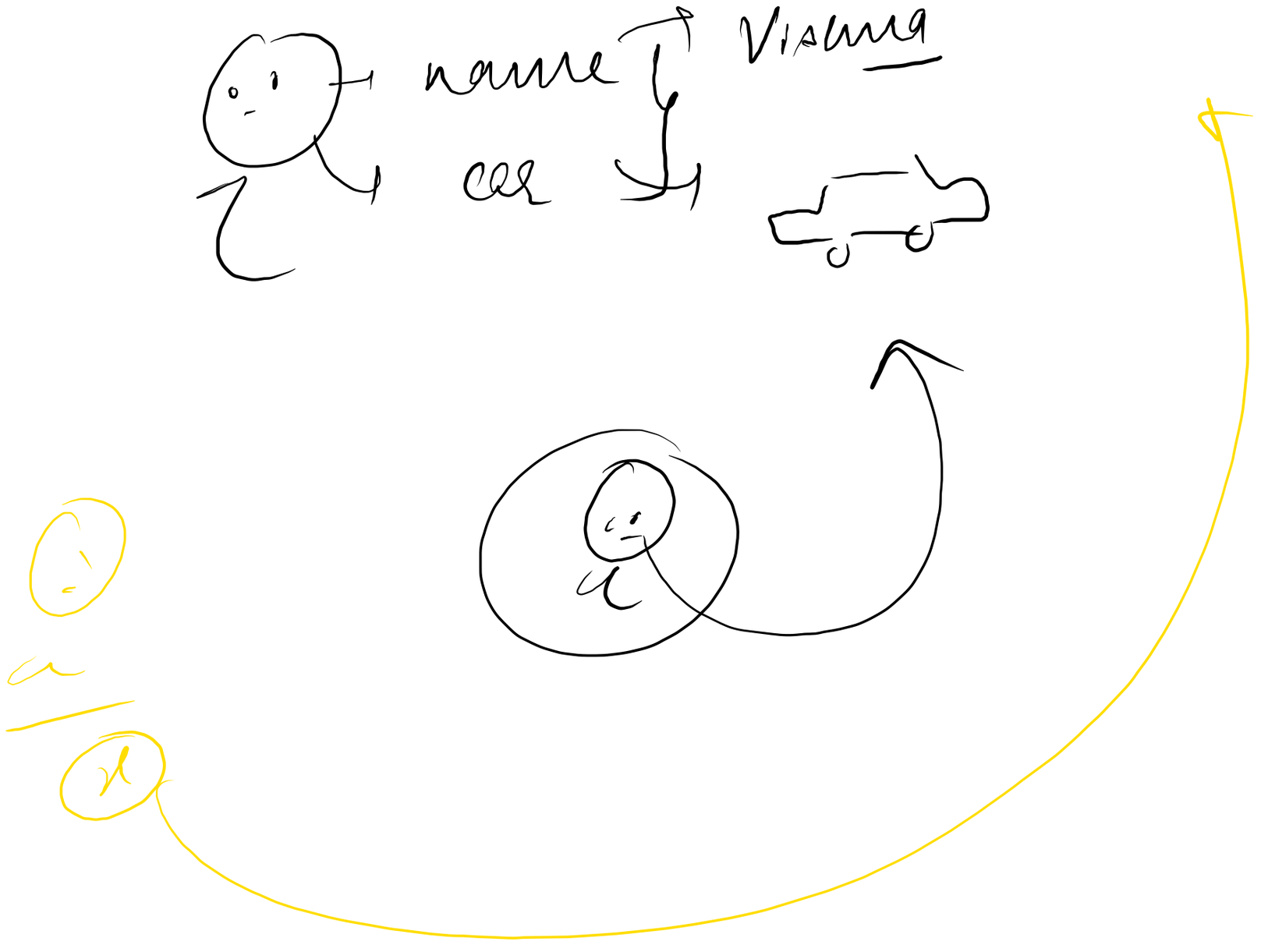
Encapsulation

↳ capsule

object → data] state
Behaviour ↑ methods

bind the data/state of the object with methods

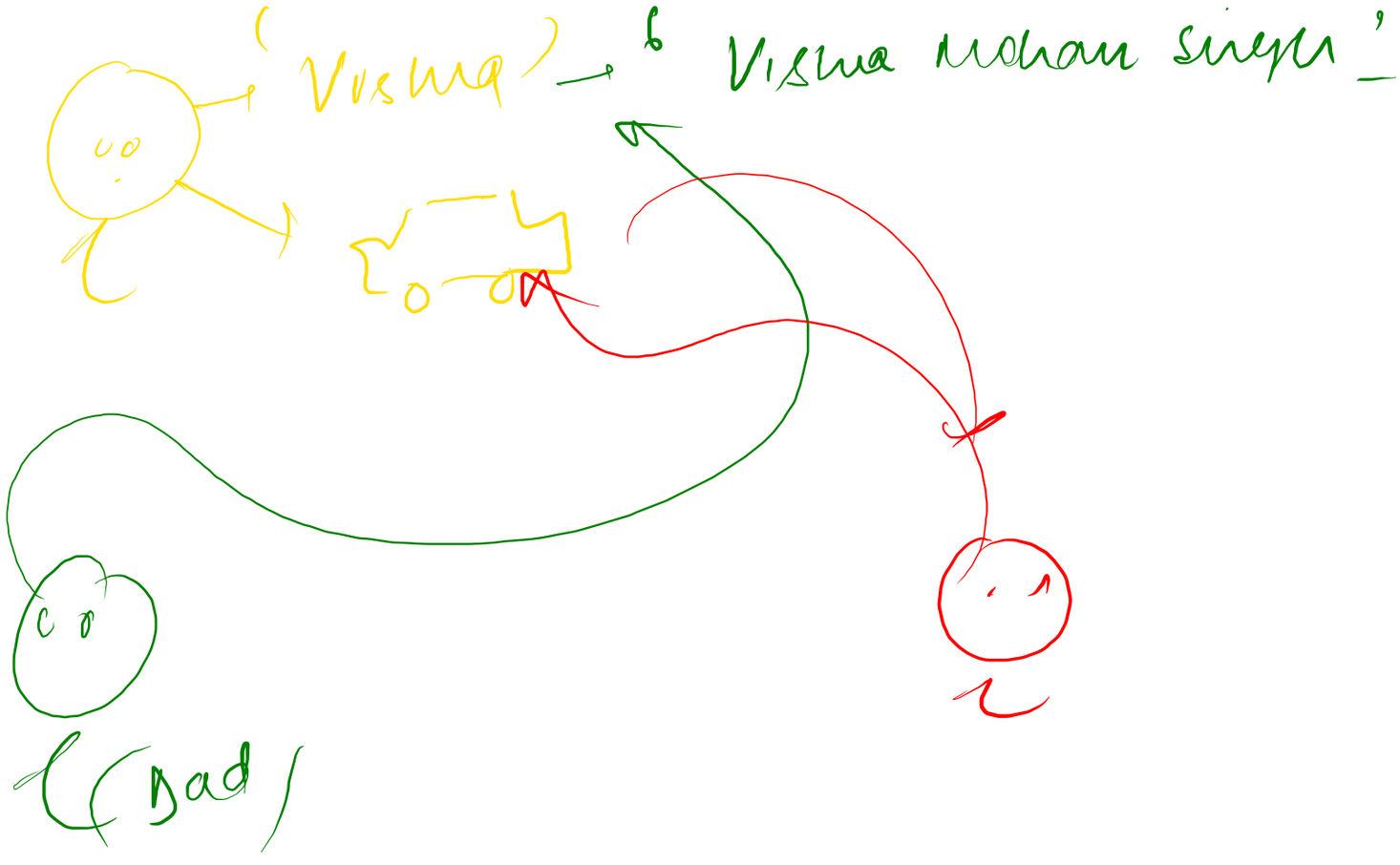
↳ hide or make the state/data private]


name
Viaggia
car

person → Object → name = 'Vishu'
age = 19
→ private

← No control

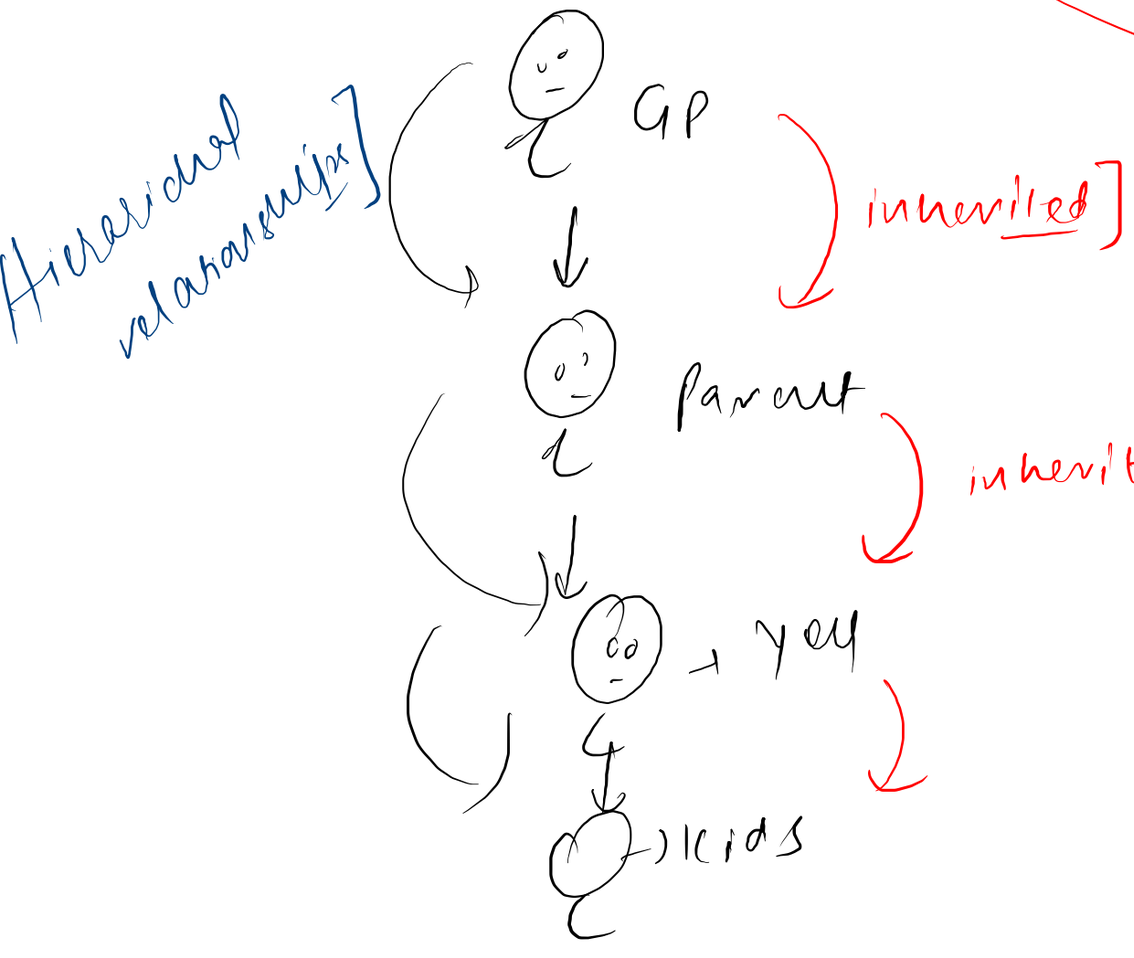
① All the fields / properties they should be private


'Vishnu'
Vishnu mohan singh!
Dad

Relationship

Parent Child relationship

Hierarchical relationship

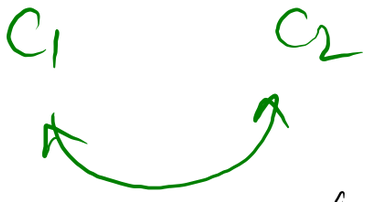


inheritance

1. Reusability

2. Hierarchical Parent child relation

What ~~~ → when



is A

C1 is A C2 → true / C1 can be the child of C2

→ false ✗ inheritance can't be applied

Cat isA Animal → true ↳ inheritance

Cat isA Machine → false ↳ X

Advantages :-

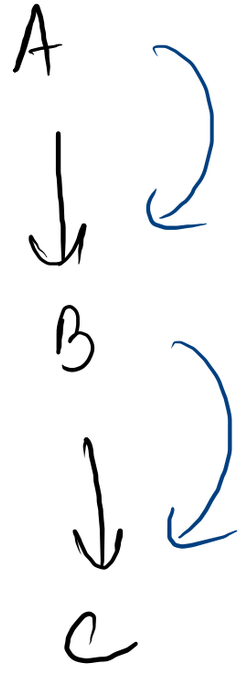
① Reuse the code

② Relationships] parent - child

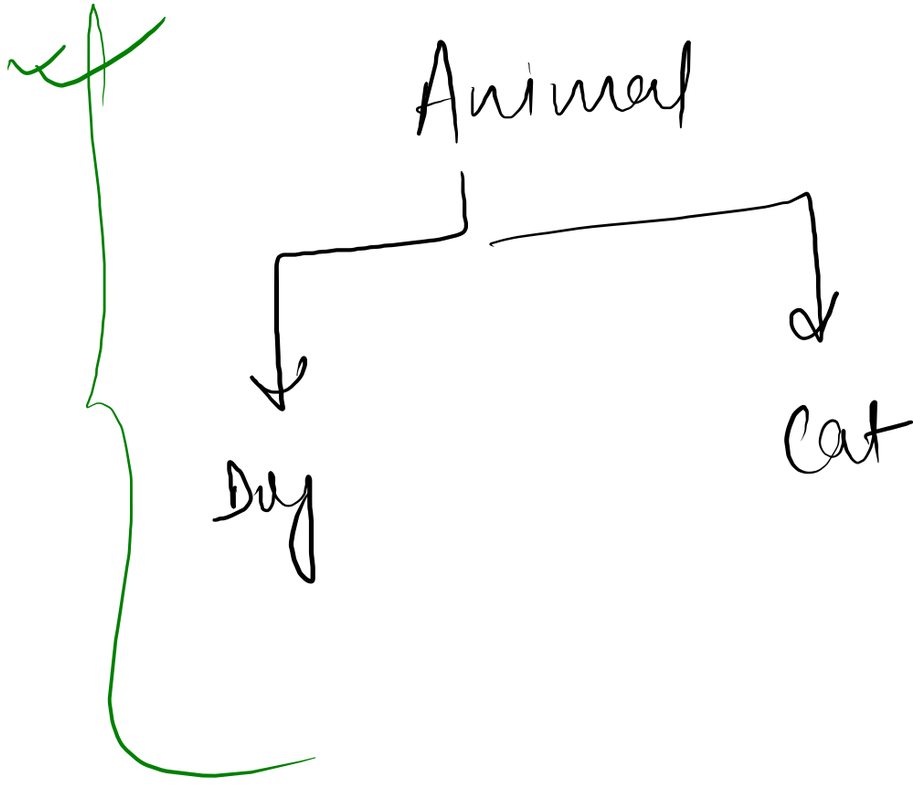
Types of inheritance

1. Single level inheritance

2. Multi level inheritance



(3) Hierarchical inheritance



④ Multi inheritance

(5) Hybrid inheritance