OOps Object oriented Design

→ way of programming

Q What is OOP?
↓
programming Techniqu
↓
Class, object

4 pillars → Inheritence
→ Polymorphism
→ Encapsulation
→ Abstraction

procedural programming
↓
functions.

⇒ **Class :-** (Blueprint)

(design) / type → State / property / field / data member
template → behaviour / function / data function

Ex Human → 2 Hand
→ 2 Leg
→ 2 Eyes
} State / properties.

Human ↓ (functi / behaviours)
walk, eat, speak

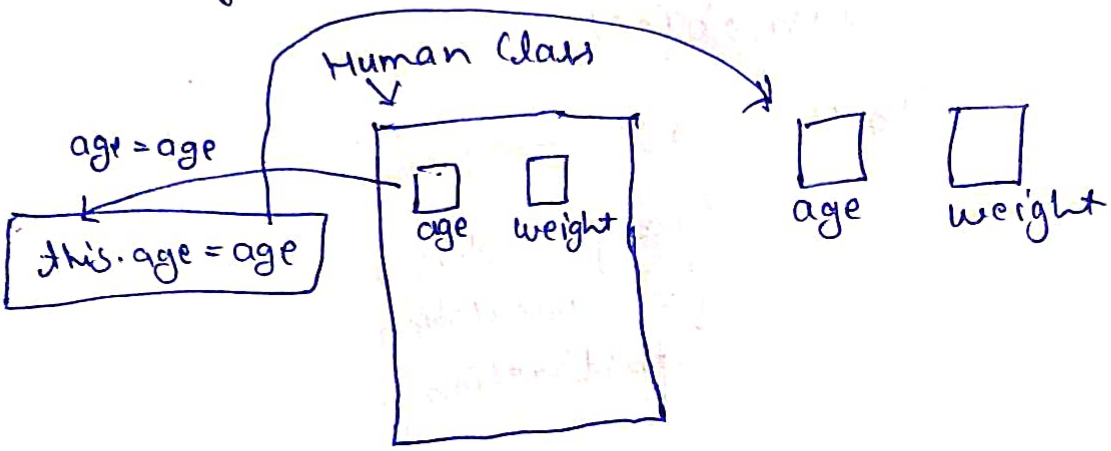
Human → class
State → hand, leg, eye
behaviour → walk(), eat(), speak()

Class is a blueprint (not take space)
object is the Implementation (take space)

constructor

- no-arg constructor
- parameterized constructor

→ Object creation (initialization)



Rules:-

⇒ No Return type.

⇒ Name → Same as Class Name.

⇒ abstract / static / final / sync. X

⇒ what if we made Constructor private.

⇒ No, object of the class can be created using "new" outside of the class.

⇒ Singleton Pattern ⇒ Only one object of the class is ever created

⇒ Utility Class → Prevent ~~ata~~ creating object of a class that only contains static methods.

⇒ factory method Pattern → static methods to control object creation logic.

⇒ Constructor Overloading :-

* Having multiple Constructors with diffrent parameters.

Ex

```
class Human {

    int age = 24;
    int weight = 74;

        public Human (int age , int weight)
        {
                this.age = age;
                this. weight = weight;
          }

        public Human (int age)
         {      this.age  = age ;
          }

    }
```

⇒ Does Constructor Return any Value ?

There is no "return value" statements in the constructor, but the constructor returns the current class instance we can write return inside the constructor.