OOPS OBJECT ORIENTED PROGRAMMING

**OOPs → Object Oriented Programming aik programming paradigm hai jo classes aur objects par based hoti hai.**  
Is mein hum real-world objects ko **class ke zariye** represent karte hain.

jisme unke **attributes → (properties)** aur **methods → (actions)** define kiye jaate hain.

**Class:**

Class = *Blue Print mean map*

****

**Class** aik collection hoti hai objects ki.  
Classes aik **blueprint** ki tarah hoti hain jinki madad se hum objects bana sakte hain.

Ek class kuch **attributes** (variables) aur **methods** (functions) define karti hai jo us class ke objects (instances) ke paas hotay hain.

**Python Class ke kuch important points:**

* Class banane ke liye Python mein **class** keyword use hota hai.
* **Attributes** wo variables hote hain jo class ke andar hotay hain.
* Attributes hamesha **public** hote hain aur un tak pohanchne ke liye **dot (.) operator** use hota hai.  
  For Example: Myclass.Myattribute

### ****Class banane ka tareeqa:****

Yahan class keyword ka matlab hai ke hum ek class create kar rahe hain.  
Uske baad class ka naam likha jata hai

**Example:**

**Class** Dog:  
 pass

Yahan Dog class ka naam hai, jo ab ek **blueprint** ban gaya hai future objects ke liye.

**\_\_init\_\_ :** \_\_init\_\_ Python ka constructor method hota hai jo object banate hi automatic call hota hai aur uske attributes ko initialize karta hai.

**Self :** Self wo reference hota hai jo class ke object ko dikhata hai. Is se hum object ke attributes aur methods ko access kar sakte hain.

Class Dog:  
 def \_\_init\_\_(self, color, sound):  
 self.color = color # Initialize color attribute  
 self.sound = sound # Initialize sound attribute

Maine jo class banayi hai, usmein \_\_init\_\_ se class initialize hoti hai aur self se attributes ko access kiya jata hai