

Ch12-More on Inheritance + functions

a Chroherty decorator

This can be used to convert a function look like an attribute.

@ property - also called getter method.

class School: students = 2000

@. setter decorator

newAdmission = 200

@ property -> decorator def newSession (self): return self. students + self. new Admission

new = School ()

hrint (new · new Session) -> 2200

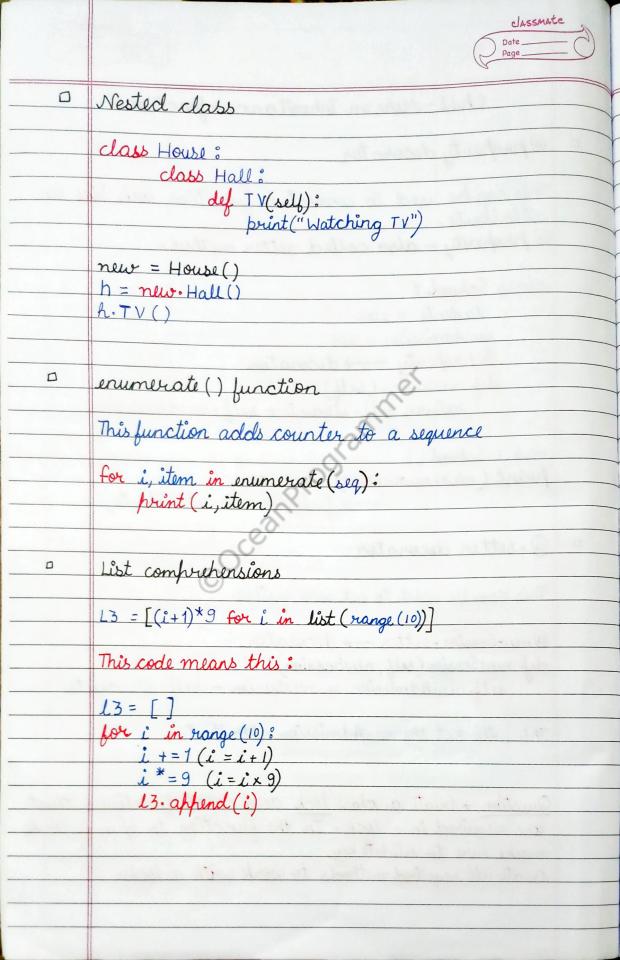
(feels like an attribute)

This can be used to set some method.

@new Session · setter -> decorator def new Session (self, new Session): self. new Admission = new Session - self. students

This will set the newAdmission attribute

Question > Greate a class lock and create functions that are required in a lock. In the function to open a lock, make sure to validate key. create all required methods to work with a lock.





*args and **kwargs def myargs (* args): > convention
print (args) myargs ("hython", "Java", "C") To insert any other sequence syntax is like: myargs (* seq) The * means - ALL ELEMENTS OF THE SEQUENCE def cricketers (** kwargs)
print (kwargs) d1 = § "M5 Dhoni": ["Batsman", "Wicket keeper"], "Yuzvendra Chahal": "Bowler", "Hardik Pandya": "All rounder"} cricketers (**d1) kwargs can be used in dictionaries Lambda functions
This is a shorter way to define functions lambda parameters: operation double = lambda a: a*2 print (double (3))

