Training Day – 18

Create A Generalized Function which takes any no of arguments and return the multiplication of all arguments using variable length keyword argument function.

# #BLL

# def mul(\*args): #args=(2, 3, 4, 5, 6)

# r=1

# for i in range(len(args)):

# r=r\*args[i]

# return r

# #PL

# r1=mul(2,3,4,5,6)

# r2=mul(1,2)

# print(r1,r2)

"""

Create your own generalized index function. Print all the matching index positions of an element present in a list [2,3,4,5,6,2,3,4,2,3,4,2]. Take the element as input from the user.

"""

# #New Program

# L=[2,3,4,5,6,2,3,4,2,3,4,2]

# ele=2

# for i in range(len(L)): #i=0,1,2,...n-1 # if(L[i]==ele):

# print(i)

Constructor: is a method, which is called automatically everytime we create an object in python. In Python the name of the constructor is fixed ie \_\_init\_\_()

"""

# #New Program # class C1:

# def \_\_init\_\_(self): #Constructor

# print("CETPA")

#

# ob1=C1()

# ob2=C1()

# ob3=C1()

Generally in programming in real world, the variables of all objects of a class are common like all customers will have same variables like id, name, age, mob so we mostly create the variables inside constructor.

# class Customer:

# def \_\_init\_\_(self): #self=1000, self=2000

# self.id=0 #1000.id=0, 2000.id=0

# self.name=0 #1000.name=0

# self.age=0 #1000.age=0

# self.mob=0 #1000.mob=0

# cus1=Customer() #cus1 1000, self 1000

# print(cus1.id,cus1.name,cus1.age,cus1.mob) # cus2=Customer() #cus2 2000 , self 2000

# print(cus2.id,cus2.name,cus2.age,cus2.mob)

Now class or static variables and methods. These variables or method are like normal variables or functions which we have studied outside class.

How To Create Static Variables:

Same syntax like outside class. Directly inside class, assign the value var\_name=value

How To Access Static Variables: using class name class\_name.var\_name

Static variables will be common variables