***Home Work 4***

**مکتب 52 – هادی کمیلی**

پاسخ سوال 1 – این خطا به این دلیل ایجاد شده که self به عنوان یک پارامتر در متد کلاس حذف شده است.

پاسخ سوال 3 – تفاوت \_\_call\_\_ و \_\_init\_\_ :

\_\_init\_\_: این متود در زمان ساخت یک instance از class استفاده می شود و به کلاس اجازه میده تا بهattributes  مقداردهی اولیه بکند.

\_\_call\_\_: این متود اجازه می دهد تا instance مثل function با آوردن نامش فراخوانی شود.

در نتیجه \_\_init\_\_ تنها در زمان ایجاد(مقداردهی اولیه) instance استفاده می شود، اما \_\_call\_\_ تنها در زمان فراخوانی instance استفاده می شود(در هر بار فراخوانی \_\_call\_\_ استفاده می شود، انگار instance مانند تابع فراخوانی می شود)

پاسخ سوال 4 - \_\_del\_\_ ، \_\_getattr\_\_، \_\_setattr\_\_، \_\_new\_\_ :

\_\_del\_\_: این متود به عنوان یک متود destructor در پایتون محسوب می شود (به نوعی نقطه مقابل \_\_new\_\_ ) که وقتی همه ارجاعات به شی حذف می شوند فراخوانی می شود.

def \_\_getattr\_\_(self, name):

\_\_getattr\_\_: زمانی فراخوانی می شود که هنگام جستجوی یک صفت / ویژگی ، آن صفت / ویژگی را در مکان های معمول پیدا نکرده است (صفت / ویژگی فراخوانی شده یک instance attr نیست یا در class پیدا نشده). این متود باید مقدار صفت / ویژگی (مقدار محاسبه شده) را برگرداند یا یک استثنا (exception AttributeError) را raise کند.

def \_\_getattr\_\_(self, name):

\_\_setattr\_\_: این متود هنگام انتساب یک صفت / ویژگی فراخوانی می شود.

def \_\_setattr\_\_(self, name, value):

\_\_new\_\_ : این متود هنگام ایجاد یک شی فراخوانی می شود. در پایتون \_\_new\_\_ متود constructor است و \_\_init\_\_ متود مقداردهی اولیه است. از \_\_new\_\_ هنگامی استفاده می شود که نیاز به ایجاد یک نمونه جدید داریم در حالی که از \_\_init\_\_ درصورت نیاز به مقداردهی اولیه یک نمونه جدید استفاده می شود.

def \_\_new\_\_(cls):

return object.\_\_new\_\_(cls)