

في الدرس اللي فات اتكلمنا على ال attributes اللي بياخدوها ال object او بمعنى اخر ال instance من ال class ودي اللي بيتم تعريفها والوصول ليها من خلال ال constructor، في الدرس دا هنتكلم عن ال attributes اللي بتخص ال class ودي مش بياخدوها ال object يعني مينفعش اعرف متغير x (attribute) جوا class اسمه على سبيل المثال Member واعمل من ال class دا object اسمه person1 واجي اقول person1.x ، لا مينفعش لأنني مش معرف ال x دا انه يكون تبع ال object انا معرفه انه يكون تبع ال class فعلشان استخدمه او اوصله هقول Member.x .

بستخدم ال attributes بتاع ال class علشان انفذ عليها tasks تخص ال Class وبتكون عامه على كل object يعني مش كل object هيتعمل من ال class دا هيكون ليه قيمه مختلفه.

على سبيل الحاجات اللي ممكن انفذها على المتغيرات دي، انا عارف ان كل اما اعرف object من ال class فانكشن ال __init__ بتننفذ. فانا عرفت متغير في ال class اسمه user_num واديتله قيمه ب 0 وعملت كاونتر جوا ال constructor انه يزود المتغير واحد كل اما بتننفذ. فبالتالي كل لما object يتعرف من ال class هتشتغل فانكشن ال constructor (ال __init__) وهتزود الكاونتر بواحد، ولو جيت عملت **print(Member.user_num)** هيطبعلي عدد ال objects اللي تم انشاءهم من ال class دا.

كمان مثال عملت attribute عبارته عن list فيها بعض الكلمات اللي مش مسموح ان الشخص يستخدمها كأسم. وفي فانكشن full_name() اللي بتطبعلي الأسم بالكامل عملت شرط ان لو الاسم بتاع الشخص موجود في ال list اللي معرف فيها الاسماء دي يطلع له error ويقله Name Not Allowed.

```
# -----
# -- Object Oriented Programming => Class Attributes --
# -----
# Class Attributes: Attributes Defined Outside The Constructor
# -----

class Member:

    not_allowed_names = ["Hell", "Shit", "Baloot"]

    users_num = 0

    def __init__(self, first_name, middle_name, last_name, gender):

        self.fname = first_name

        self.mname = middle_name

        self.lname = last_name

        self.gender = gender

    Member.users_num += 1 # Member.users_num = Member.users_num + 1
```

```

def full_name(self):

    if self.fname in Member.not_allowed_names:

        raise ValueError("Name Not Allowed")

    else:

        return f"{self.fname} {self.mname} {self.lname}"

def name_with_title(self):

    if self.gender == "Male":
        return f"Hello Mr {self.fname}"
    elif self.gender == "Female":
        return f"Hello Miss {self.fname}"
    else:
        return f"Hello {self.fname}"

def get_all_info(self):

    return f"{self.name_with_title()}, Your Full Name Is: {self.full_name()}"

print(f"There's: {Member.users_num} Object/s Instantiated From This Class.")

member_one = Member("Muhammad", "Ahmed", "Naeem", "Male")
member_two = Member("Ahmed", "Ali", "Mahmoud", "Male")
member_three = Member("Mona", "Ali", "Mahmoud", "Female")
member_four = Member("Shit", "Hell", "Metal", "DD")

print(f"There's: {Member.users_num} Object/s Instantiated From This Class.")

print(member_two.full_name())
print(member_four.full_name())

```

CODE

```

There's: 0 Object/s Instantiated From This Class.
There's: 4 Object/s Instantiated From This Class.
Ahmed Ali Mahmoud
Traceback (most recent call last):
  File "c:\Users\Muhammad\Documents\Python Course\first.py", line 64, in <module>
    print(member_four.full_name())
  File "c:\Users\Muhammad\Documents\Python Course\first.py", line 29, in full_name
    raise ValueError("Name Not Allowed")
ValueError: Name Not Allowed

```

OUTPUT