Chapter 11 - Supermethod

Theory:

Super method calls methods from parent classes.

Code Example:

```
# super().__init__() is a method used in Python classes when working with inheritance.
# It allows a child class to call the constructor (__init__ method) of its parent class.
# ■ Why use super().__init__()?
# When a class inherits from another class,
# it doesn't automatically run the parent’s __init__() method
# — you have to call it explicitly using super().__init__().
#syntax
# class Parent:
# def __init__(self):
# print("Parent constructor")
# class Child(Parent):
# def __init__(self):
# super().__init__() # Calls Parent's __init__
# print("Child constructor")
class Animal:
def __init__(self, name):
print(f"Animal ka naam: {name}")
class Dog(Animal):
def __init__(self, name, breed):
super().__init__(name) # ■ Animal class ka kaam
print(f"Dog ka breed: {breed}")
class Price(Dog):
def __init__(self, name, breed, price):
super().__init__(name, breed)
print(f"The price of this dog is : {price}")
d = Price("Tommy", "Labrador",50000)
class Employee:
def __init__(self):
print(f"constructor of Employe")
a = 1
class Programmer(Employee):
def __init__(self):
print(f"constructor of Programer")
b = 2
class Manager(Programmer):
def __init__(self):
super().__init__()
print(f"constructor of Manager")
c = 3
o = Employee()
print(o.a)
o2 = Programmer()
print(o2.a , o2.b)
o3 = Manager()
print(o3.a , o3.b , o3.c)
```