

Abstract class mein super() ki zaroorat hoti hai ya nahi?"

Pehle yeh samjho: super() kya karta hai?

```
super().__init__()
```

Yeh line parent class ka constructor call karti hai — yaani agar child class kuch extra kaam kar rahi hai, to super() se pehle parent class ka setup bhi ho jata hai.

🔧 Abstract class kya hoti hai?

```
from abc import ABC, abstractmethod

class Animal(ABC):
    def __init__(self):
        print("Animal created!")

    @abstractmethod
    def make_sound(self):
        pass
```

Yeh ek **abstract class** hai. `Animal` class ka `make_sound()` method **abstract** hai, yaani koi child class jab tak isay **implement nahi karegi**, wo class complete nahi hogi.

✓ **Jab `super()` kaam aata hai:**

□ **Example:**

```
from abc import ABC, abstractmethod

class Animal(ABC):
    def __init__(self):
        print("Animal constructor called")

    @abstractmethod
    def make_sound(self):
        pass

class Dog(Animal):
    def __init__(self):
        super().__init__() # 🏠 Yahan parent ka constructor call kiya
        print("Dog constructor called")

    def make_sound(self):
        print("Bark!")
```

```
Animal constructor called
Dog constructor called
```

💡 **Tashreeh:**

- Dog ne jab `super().__init__()` use kiya to `Animal` ka constructor bhi chala.
 - Agar `super()` na lagate, to `Animal` ka constructor skip ho jata.
 - **To agar parent class (abstract ya normal) kuch important setup kar rahi hai, to `super()` zaroori hai.**
-

✗ Jab `super()` ki zarurat nahi hoti:

Agar tumhari **abstract class ka constructor kuch bhi nahi kar raha**, to `super()` lagana **optional** hota hai.

🔗 Example:

```
from abc import ABC, abstractmethod

class Animal(ABC):
    @abstractmethod
    def make_sound(self):
        pass

class Cat(Animal):
    def make_sound(self):
        print("Meow")
```

Yahan `Animal` ka constructor hai hi nahi, is liye `Cat` class mein `super()` ki zaroorat nahi.

❑ Viva Question jese jawab:

Q: Kya abstract class mein `super()` lagana zaroori hota hai?

✓ Answer:

Nahi, hamesha zaroori nahi hota. Agar abstract class ka constructor koi kaam kar raha ho — jaise variables set karna ya logging — to child class mein `super().__init__()` lagana chahiye taake wo kaam bhi ho jaye. Lekin agar abstract class ka constructor empty ho, to `super()` lagana optional hai.