Day 41

super keyword in Python:

What is super()?

super() is used to call methods or access attributes from a parent (base) class inside a child (derived) class.

It helps with:

- Code reuse
- Method overriding
- Multiple inheritance
- Avoiding direct parent class names

Why do we need super()?

Without super():

- You must manually call the parent class
- Breaks easily if class names change
- Causes issues in multiple inheritance

With super():

- Python automatically finds the next class in the inheritance chain

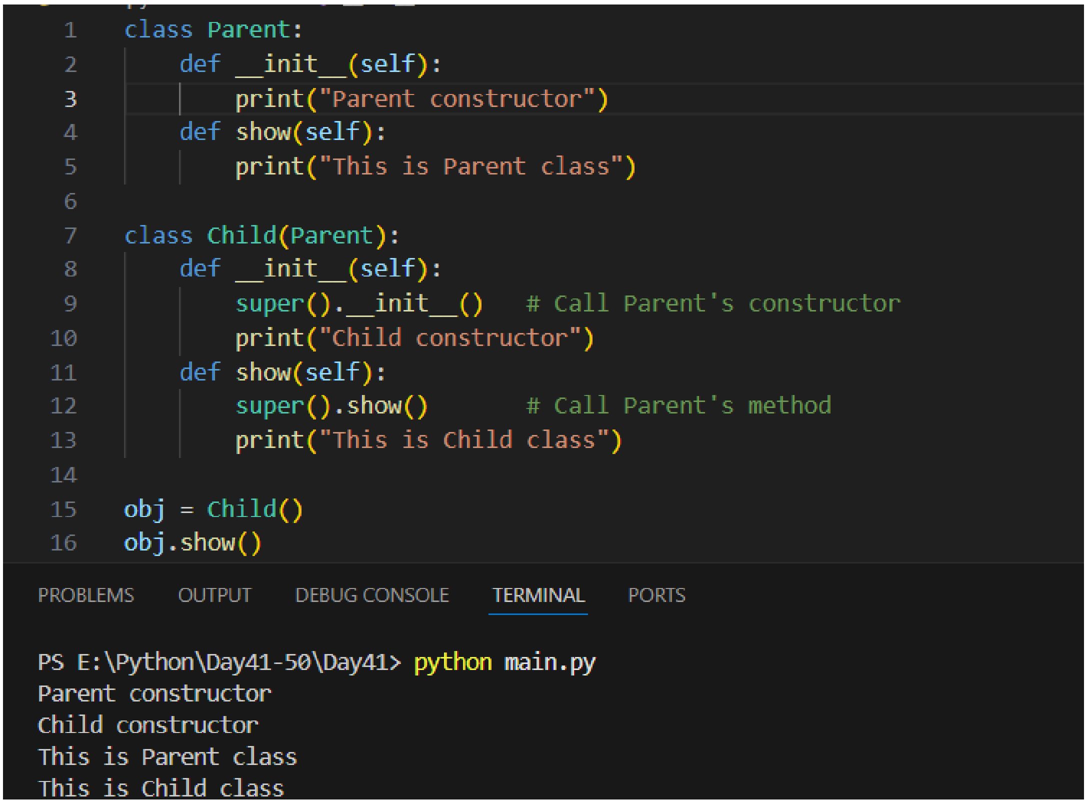
Basic Syntax:

super().method_name()

or inside __init__:

super().__init__(arguments)

Example:



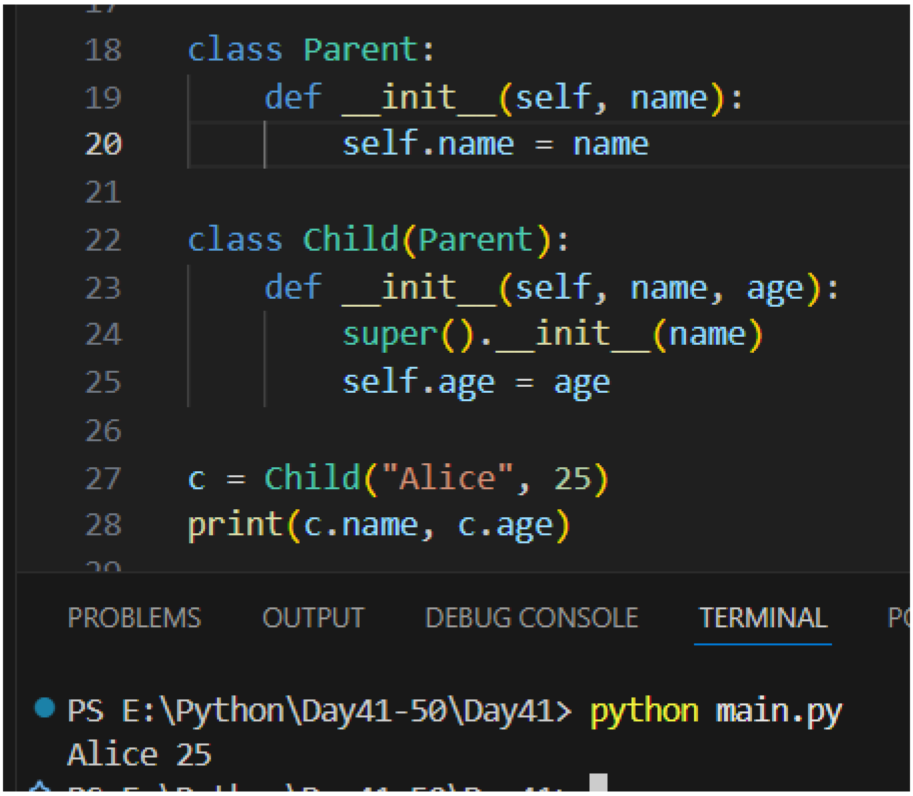
```python
class Parent:
    def __init__(self):
        print("Parent constructor")
    def show(self):
        print("This is Parent class")

class Child(Parent):
    def __init__(self):
        super().__init__()    # Call Parent's constructor
        print("Child constructor")
    def show(self):
        super().show()        # Call Parent's method
        print("This is Child class")

obj = Child()
obj.show()
```

```
PS E:\Python\Day41-50\Day41> python main.py
Parent constructor
Child constructor
This is Parent class
This is Child class
```

Example: super() with parameters



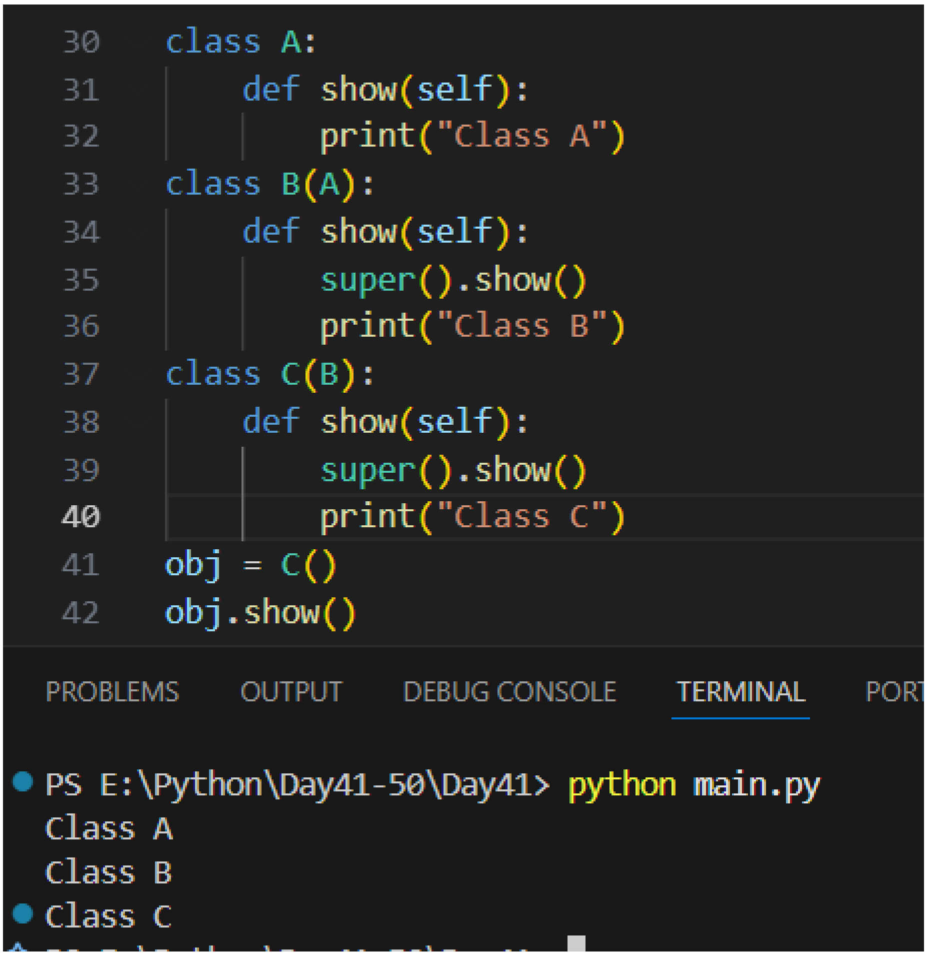
```python
class Parent:
    def __init__(self, name):
        self.name = name

class Child(Parent):
    def __init__(self, name, age):
        super().__init__(name)
        self.age = age

c = Child("Alice", 25)
print(c.name, c.age)
```

```
PS E:\Python\Day41-50\Day41> python main.py
Alice 25
```

Example: super() in multiple inheritance



```python
class A:
    def show(self):
        print("Class A")
class B(A):
    def show(self):
        super().show()
        print("Class B")
class C(B):
    def show(self):
        super().show()
        print("Class C")
obj = C()
obj.show()
```

```
PS E:\Python\Day41-50\Day41> python main.py
Class A
Class B
Class C
```