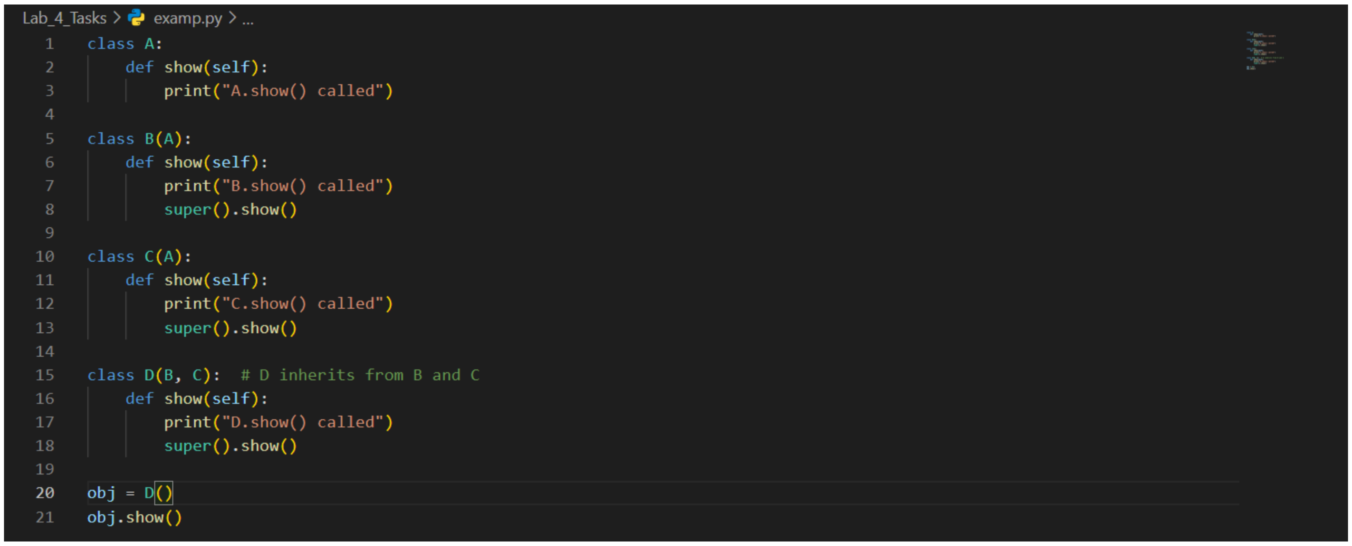
SUPER( ) PYTHON REPORT

1. How super() Handles Multiple Inheritance in Python

The super() function is used to call methods from a parent class. In the case of multiple inheritance. When used in a class, super() doesn't simply refer to the immediate parent — it refers to the next nearest class. This makes super() ensure that all classes in a hierarchy get a chance to execute their methods.

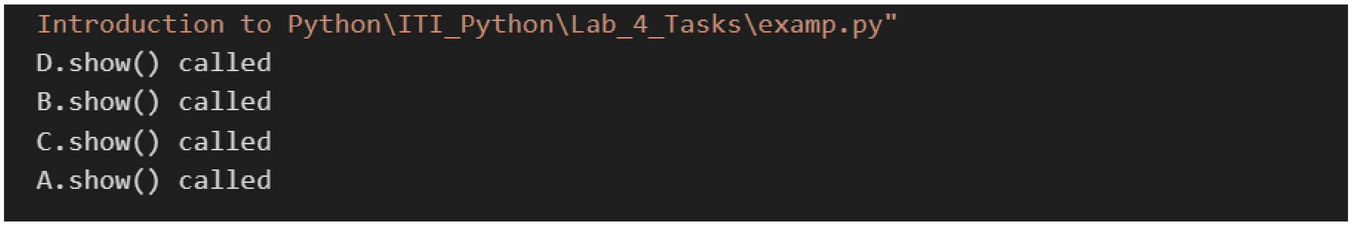
Example: Multiple Inheritance



Lab_4_Tasks > examp.py > ...

```python
class A:
    def show(self):
        print("A.show() called")

class B(A):
    def show(self):
        print("B.show() called")
        super().show()

class C(A):
    def show(self):
        print("C.show() called")
        super().show()

class D(B, C):  # D inherits from B and C
    def show(self):
        print("D.show() called")
        super().show()

obj = D()
obj.show()
```

Output:



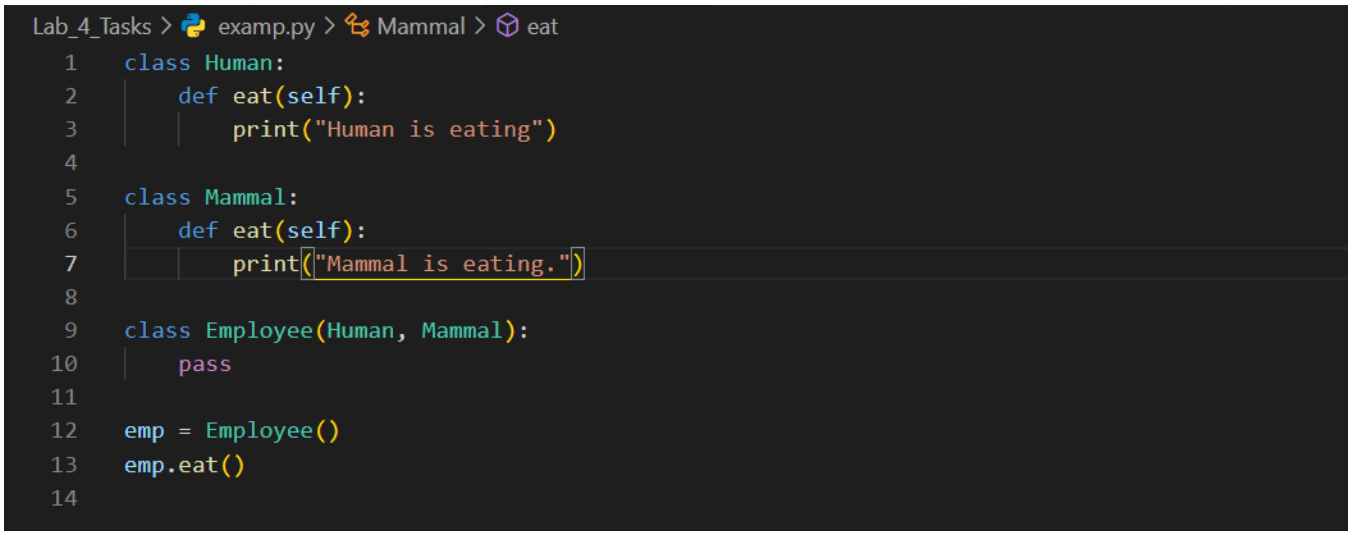
```
Introduction to Python\ITI_Python\Lab_4_Tasks\examp.py"
D.show() called
B.show() called
C.show() called
A.show() called
```

Explanation:

- The method resolution order for class D is D → B → C → A → object.
- Each super().show() call moves to the next class.
- This ensures that each class gets one opportunity to handle the method, without duplication or skipping.

2. Same Method in Inheritance with Multiple Parents (Overriding):

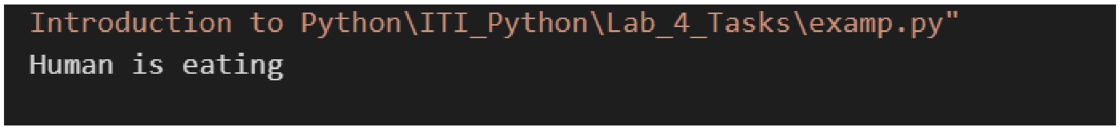
Suppose we have two parent classes: Human and Mammal, both define a method named eat(), but with different implementations. A child class Employee inherits from both.



Lab_4_Tasks > examp.py > Mammal > eat

```python
class Human:
    def eat(self):
        print("Human is eating")

class Mammal:
    def eat(self):
        print("Mammal is eating.")

class Employee(Human, Mammal):
    pass

emp = Employee()
emp.eat()

```

Output:



```
Introduction to Python\ITI_Python\Lab_4_Tasks\examp.py"
Human is eating
```

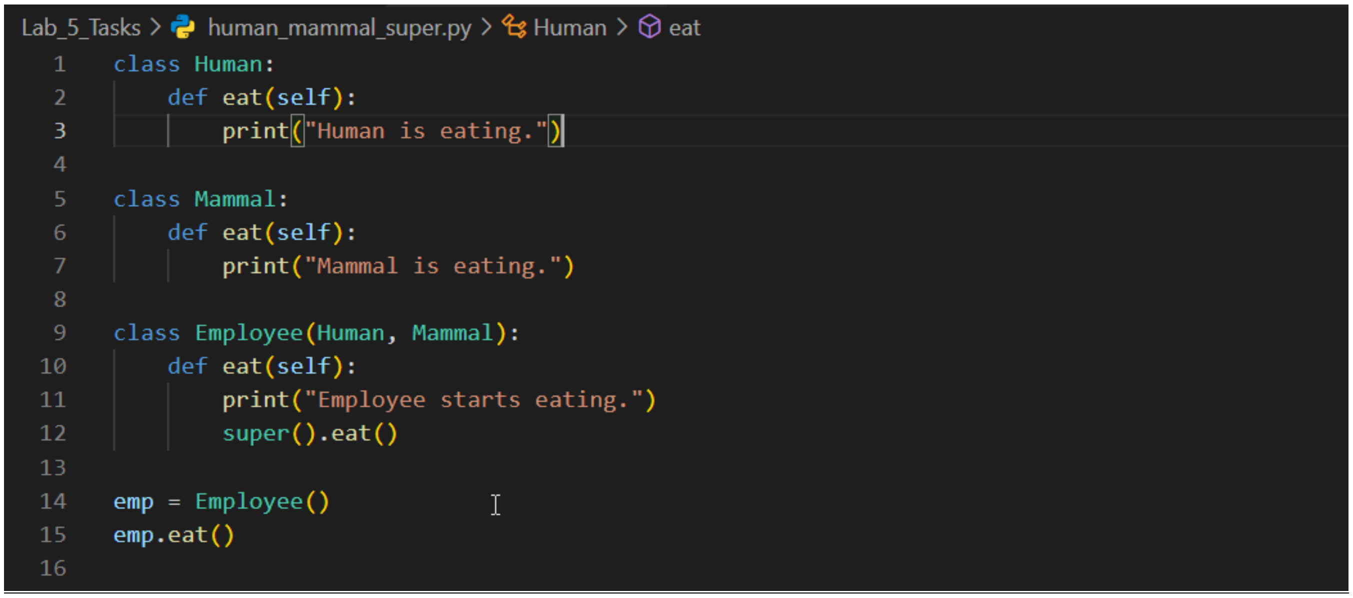
Explanation:

Python uses Method Resolution Order (MRO) to decide which method to call. The order is based on the class definition:

So, when emp.eat() is called:

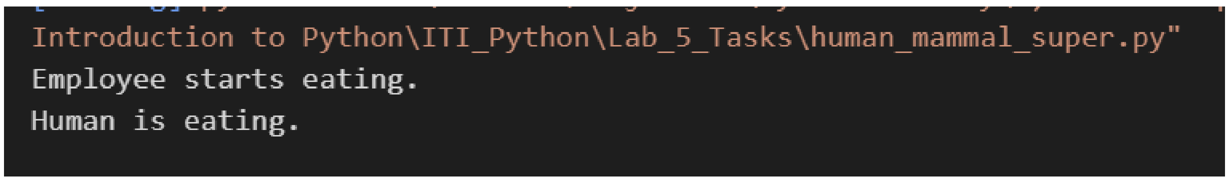
- Python checks Employee → Human → Mammal → object.
- Since Human has eat(), it is invoked, and Mammal's method is ignored unless explicitly called.

Using super( ) in Inheritance with Multiple Parents (Overriding):



```python
class Human:
    def eat(self):
        print("Human is eating.")

class Mammal:
    def eat(self):
        print("Mammal is eating.")

class Employee(Human, Mammal):
    def eat(self):
        print("Employee starts eating.")
        super().eat()

emp = Employee()
emp.eat()
```

Output:



```
Introduction to Python\ITI_Python\Lab_5_Tasks\human_mammal_super.py"
Employee starts eating.
Human is eating.
```

Explanation:

- Employee.eat() calls super().eat() → goes to Human.eat()
- This follows: Employee → Human → object