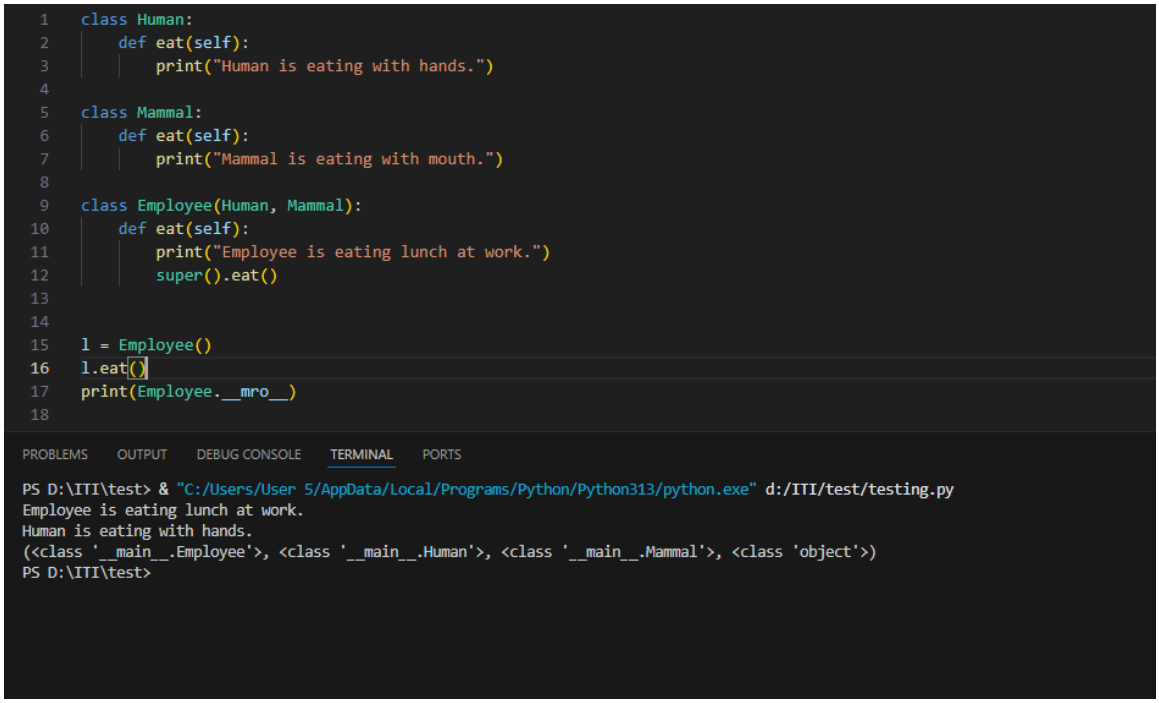
Multiple Inheritance in Python

- ## How super() Handles Multiple Inheritance

In Python, the super() function follows the Method Resolution Order to determine the next class whose method should be called. MRO is determined by the C3 linearization algorithm, which ensures a consistent order of method resolution. When multiple inheritance is involved, super() does not just call the immediate parent; instead, it calls the next class in the MRO chain.

- ## 2. Handling Same Method Names in Multiple Parents

If two parent classes have the same method (e.g., eat) but with different implementations, and the child class (e.g., Employee) calls the method, Python resolves which method to call using the MRO order. The method from the first parent in the inheritance list is called first, unless explicitly overridden in the child. Using super().method() ensures that the call continues to the next class in the MRO chain.



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

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

class Employee(Human, Mammal):
    def eat(self):
        print("Employee is eating lunch at work.")
        super().eat()


l = Employee()
l.eat()
print(Employee.__mro__)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\ITI\test> & "C:/Users/User 5/AppData/Local/Programs/Python/Python313/python.exe" d:/ITI/test/testing.py
Employee is eating lunch at work.
Human is eating with hands.
(<class '__main__.Employee'>, <class '__main__.Human'>, <class '__main__.Mammal'>, <class 'object'>)
PS D:\ITI\test>
```