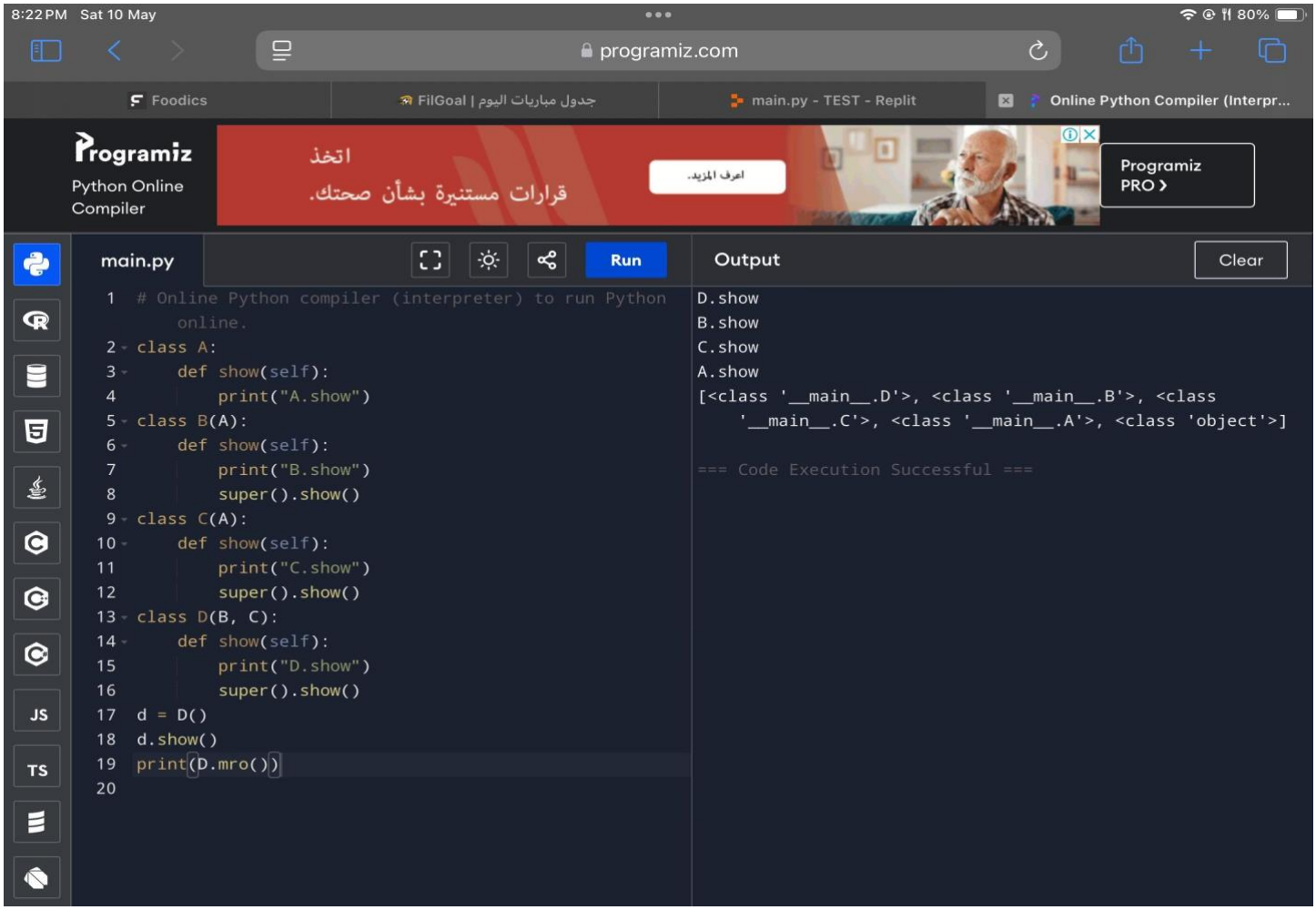
Abdall Khaled Alaassar

How super() Handles Multiple Inheritance

In multiple inheritance, the super() function follows the Method Resolution Order (MRO) to determine which class's method to call.

Python uses the C3 Linearization Algorithm to compute the MRO.

This ensures that super() doesn't just call the parent directly, but follows a predictable and consistent order across all inherited classes.

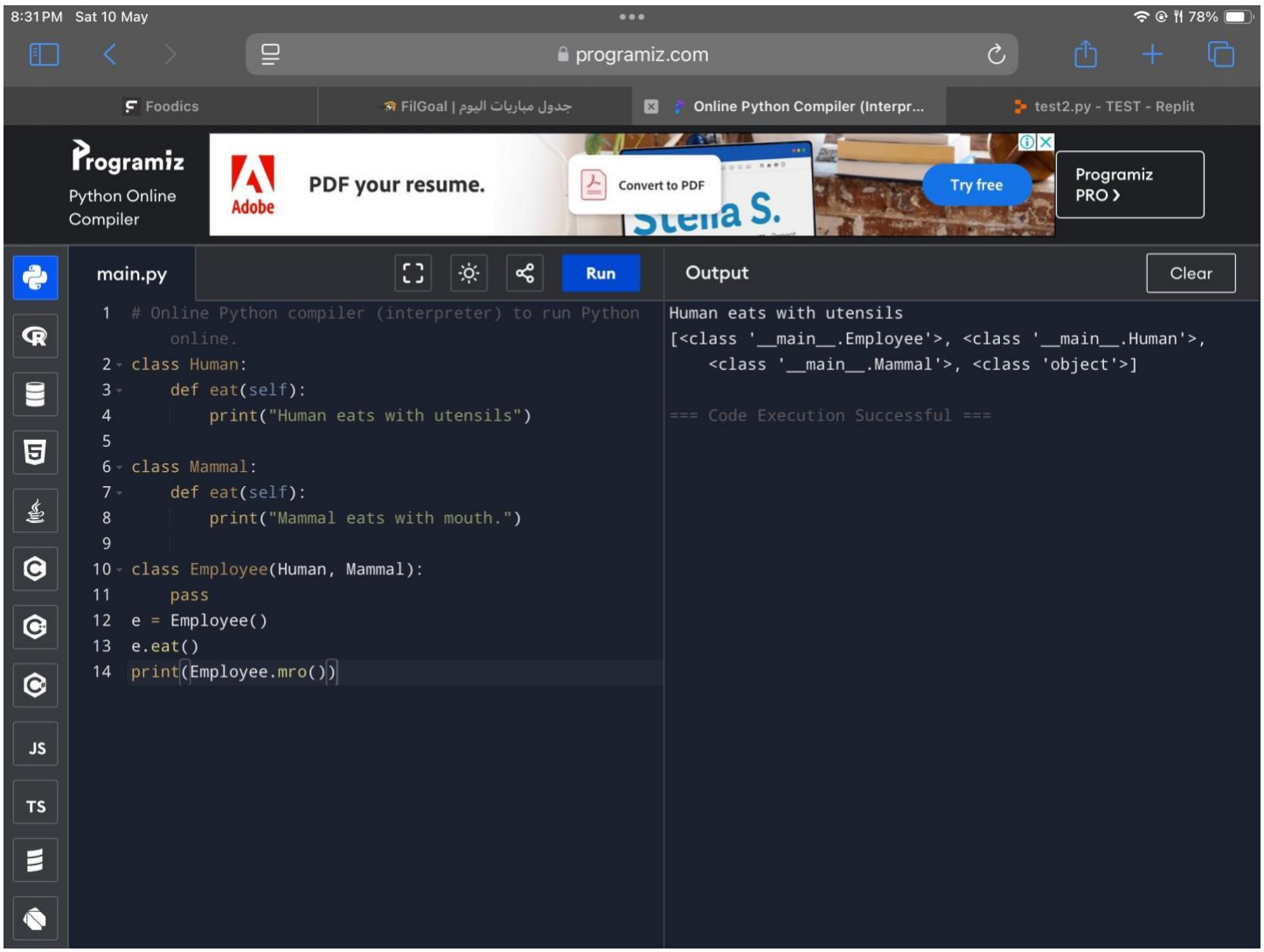


Explanation:

- super() in class D refers to B, which in turn calls C, then A.
- This chain is based on the MRO: D -> B -> C -> A -> object.

2. Method Conflict: Same Method in Multiple Parents

If two parent classes define the same method (e.g., eat) with different implementations, the method that gets executed in the child class depends on the MRO.



Explanation:

- Employee inherits from Human and Mammal, in that order.
- So, the MRO is: Employee -> Human -> Mammal -> object.
- Therefore, e.eat() calls Human.eat().

Conclusion

- Python uses MRO to resolve method calls in multiple inheritance.
- The super() function follows this MRO, not the immediate parent.
- Method conflicts are resolved by the order in which base classes are listed.

Resources

- https://docs.python.org/3/library/functions.html#super
- https://www.python.org/download/releases/2.3/mro/
- https://www.geeksforgeeks.org/super-keyword/