Understanding MRO and Inheritance in Python

# 1. Class Hierarchy Overview

In this code, we’re working with a layered inheritance setup. At the top, we have a base class called Creature. Animal inherits from Creature. Then both Dog and Cat inherit from Animal. Finally, Bulldog inherits from both Dog and Cat. This structure forms what’s known as a diamond-shaped inheritance pattern.

Here’s a quick visual of the hierarchy:

Creature  
 |  
 Animal  
 / \  
Dog Cat  
 \ /  
 Bulldog

# 2. How Method Resolution Order Works

When you create an instance of the Bulldog class and call the `speak()` method, Python needs to figure out which version of `speak()` to use. Since multiple classes in the hierarchy define that method, Python uses something called the Method Resolution Order, or MRO, to decide the order in which it checks the classes.

For this example, the MRO for Bulldog is:

<class '\_\_main\_\_.Bulldog'>, <class '\_\_main\_\_.Dog'>, <class '\_\_main\_\_.Cat'>, <class '\_\_main\_\_.Animal'>, <class '\_\_main\_\_.Creature'>, <class 'object'>

# 3. Why "Dog barks" Is Printed

When we run `b = Bulldog(); b.speak()`, Python starts at Bulldog, then moves to Dog. Dog has a `speak()` method, so Python calls it and stops searching. That’s why you see 'Dog barks' printed on the screen. Even though Cat also has a `speak()` method, it’s never reached in this case.

# 4. What’s Special About This Order?

Python uses something called C3 Linearization to build the MRO. It’s a fancy way to make sure that:  
- Each class only shows up once in the order.  
- Child classes come before their parents.  
- The order respects the way classes are listed when you define them, like `Bulldog(Dog, Cat)`.

# 5. Final Thoughts

This example is a great way to see how Python handles multiple inheritance. Even if the structure looks a bit complex, Python has clear rules to figure things out. The key takeaway is: Python always follows a consistent order, and knowing it helps you avoid surprises when working with classes that share method names.