Inheritance in Python without using super()

We can use inheritance in Python without explicitly using the **super()** function, but it's important to note that not using **super()** might lead to some limitations, and it may not handle certain cases, especially when dealing with multiple inheritance.

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

    def speak(self):
        print(f"{self.name} makes a sound")

class Dog(Animal):
    def __init__(self, name, breed):
        # Call the constructor of the parent class directly
        Animal.__init__(self, name)
        self.breed = breed

    def bark(self):
        print(f"{self.name} barks")


dog = Dog("Buddy", "Labrador")

# Accessing attributes and methods from the Animal class through
inheritance
dog.speak()       # Output: Buddy makes a sound

# Accessing attributes and methods from the Dog class
dog.bark()        # Output: Buddy barks
```

Explanation:

1. Animal Class: This is the base class, representing a generic animal. It has an **__init__** method to initialize the **name** attribute and a **speak** method to print a generic sound.
2. Dog Class (Inherits from Animal): The **Dog** class is the derived class that inherits from the **Animal** class. It has its own constructor **(__init__)**, where it calls the constructor of the parent class (**Animal.__init__(self, name)**) to initialize the name attribute. Additionally, it introduces its own attribute breed.
3. Usage: An instance of the **Dog** class is created, and we demonstrate how it can access attributes and methods from its parent class (**Animal**). The speak method from the **Animal class** is called through **inheritance**. The **bark** method is specific to the **Dog** class.

In this example, the **Dog class** exhibits single inheritance, inheriting from a single parent class (**Animal**). The direct invocation of the parent class's constructor is used to ensure that the initialization of the name attribute is handled by the **Animal** class. This is a straightforward example to illustrate the concept of single inheritance in Python.