Inheritance Questions in Python

1. Single Inheritance:
Create a base class Person with a method display_name(). Inherit it in a class Student and call the method.

2. Multilevel Inheritance:
Design 3 classes: Animal → Mammal → Dog, where each class has its own method and Dog inherits all behaviors.

3. Multiple Inheritance:
Create two classes Flyable and Swimmable, each with a method. Derive a class Duck from both and call both methods.

4. Hierarchical Inheritance:
Define a parent class Vehicle, and create two child classes Car and Bike. Show how each inherits from Vehicle.

5. Use super() in a derived class to call a parent class's method. What happens if both classes have the same method name?

6. What is Method Resolution Order (MRO) in multiple inheritance? Demonstrate using a diamond problem structure.

7. Define a constructor in the base class. In the derived class, call it using super().__init__() and add new attributes.

8. Can you override a method in Python? Write a base class Shape with a method area() and override it in Circle.

Polymorphism Questions in Python

9. Method Overriding:

Write a base class Animal with method speak(). Create subclasses Dog, Cat that override speak().

10. Polymorphic Behavior:
Create a list of objects of Dog, Cat, Cow, each inheriting from Animal. Iterate and call speak() method.

11. Simulated Method Overloading:
Python doesn't support method overloading directly. Show how you can use default or *args to mimic it.

12. Write a class Calculator with a method add() that supports 2 and 3 arguments using default parameters or *args.

13. Can you override the __str__() method in Python? Create a class Book that returns a custom string when printed.

14. Demonstrate polymorphism using duck typing. Write a function start_engine(vehicle) that takes any object with a method start().

15. How does polymorphism help in writing more generic functions in Python? Provide a small real-world code snippet.