Topic Related Questions

**1.What is Abstraction?**

**Ans:-** Abstraction means hiding complex implementation details and showing only essential features.  
➡ Achieved using **abstract classes** and **interfaces** in Java.  
➡ Example: abstract void draw();

**2.** **Difference Between Interface and Abstract Class**

**Ans:-**

| Feature | Interface | Abstract Class |
| --- | --- | --- |
| Methods | Only abstract (Java 7), default/static allowed (Java 8+) | Abstract + concrete allowed |
| Variables | Public static final only | Any access modifier |
| Inheritance | Multiple interfaces allowed | Only single class allowed |
| Constructor | Not allowed | Can have constructors |

**3.** **Explain Polymorphism with Example**

**Ans:-** Polymorphism allows one method to behave differently for different objects.  
➡ **Compile-time**: Method overloading  
➡ **Runtime**: Method overriding

Animal a = new Dog();

a.sound(); // Output: Bark

**4. What is Method Overriding?**

**Ans:** Redefining a parent class method in a subclass with the same name and parameters.  
➡ Used in runtime polymorphism.

**5.** **Explain “IS-A” vs “HAS-A”?**

**Ans:-**  **IS-A**: Inheritance – subclass is a type of superclass.  
Dog IS-A Animal

 **HAS-A**: Composition – class has another class as a member.  
Car HAS-A Engine

**6.** **Why Use Inheritance?**

**Ans:-** To reuse code, enhance readability, and establish a parent-child relationship.  
➡ Enables **polymorphism** and improves maintainability.

**7.** **What is Dynamic Binding?**

**Ans:-** The method call is resolved at runtime, not compile-time.  
➡ Happens with overridden methods via parent class reference.

**8.** **What is Constructor Chaining?**

**Ans:-** Calling one constructor from another constructor.  
➡ this() for same class, super() for parent class constructor.

**9.** **How to Implement Encapsulation?**

**Ans:-** Keep variables private, and access them using public getters/setters.  
➡ Ensures data protection and control.

**10.** **Explain super Keyword?**

**Ans:-** super is used to refer to parent class's constructor or methods.  
➡ Helps access overridden methods or hidden fields.