Topic Related Questions

**1.** **What is inheritance in Java?**

**Ans:-** OOP feature where a subclass acquires fields/methods of a superclass. Promotes code reuse and method overriding. extends keyword used.

**2.** **Why use this keyword?**

**Ans:-**

this – Refers to current object. Used to access instance variables, call other constructors (this(...)), pass current object, avoid naming conflicts.

**3.** **What is method overriding vs overloading?**

**Ans:-**

 **Overriding**: Same method signature in subclass, changes behavior, runtime binding, needs @Override.

 **Overloading**: Same method name but different parameters or types, compile-time binding.

**4.** **What is object instantiation?**

**Ans:** Creating an instance of a class using new, which allocates memory and calls the constructor.

**5.** **Explain single vs multiple inheritance.**

**Ans:-**  **Single**: One class inherits from another.

 **Multiple**: Class inherits from multiple classes — not supported with classes in Java (avoids diamond problem), but possible with multiple interfaces.

**6.** **What is encapsulation?**

**Ans:-** Bundling data (fields) and methods, restricting direct access using private and controlling it with getters/setters. Improves security and flexibility.

**7.** **What is constructor overloading?**

**Ans:-**Defining multiple constructors in the same class with different parameter lists to create objects in different ways.

**8.** **Can we override static methods?**

**Ans:-** Static methods can’t be overridden; they can only be hidden in subclasses. Static binding occurs at compile-time.

**9.** **What is runtime polymorphism?**

**Ans:-** Method call resolved at runtime based on actual object type, achieved through method overriding and dynamic method dispatch.

**10.** **Difference between class and object?**

**Ans:-** **Class**: Blueprint/template with variables and methods, no memory until object creation.

 **Object**: Real instance of a class stored in heap memory, created using new.