1. What is a Constructor?

A constructor in Java is a special type of method that is used to initialize objects. It is called when an object of a class is created. Constructors have the same name as the class and do not have a return type, not even `void`.

2. What is Constructor Chaining?

Constructor chaining refers to the process of calling one constructor from another constructor within the same class or in the parent class. It allows you to reuse code from one constructor in another constructor.

3. Can we call a subclass constructor from a superclass constructor?

No, a subclass constructor cannot be called directly from a superclass constructor. Constructors are not inherited in Java, so you cannot use `super()` to call a subclass constructor from a superclass constructor.

4. What happens if you keep a return type for a constructor?

If you declare a return type for a constructor, it essentially becomes a regular method and not a constructor. Constructors should not have a return type, not even `void`. If you add a return type, it will not be considered a constructor, and it will not initialize an object when called.

5. What is No-arg constructor?

A no-argument constructor is a constructor that takes no parameters. It doesn't accept any arguments. If you don't provide any constructor for a class, the Java compiler automatically creates a default no-argument constructor.

6. How is a No-argument constructor different from the Default Constructor?

A no-argument constructor and a default constructor essentially mean the same thing. A no-argument constructor is one that doesn't take any parameters. The default constructor is the one that is automatically provided by the Java compiler if you don't define any constructor in your class.

7. When do we need Constructor Overloading?

Constructor overloading is used when you want to create multiple constructors in a class, each with a different set of parameters. This allows objects of the class to be initialized in different ways. Constructor overloading provides flexibility when creating objects by allowing different ways to initialize them based on the parameters passed.

8. What is Default constructor? Explain with an Example

A default constructor is a no-argument constructor provided by Java if you don't define any constructors in your class. It initializes the object with default values. For example:

public class MyClass {

int value;

// Default constructor provided by Java if no constructors are defined

public MyClass() {

value = 10; // default initialization

}

public static void main(String[] args) {

MyClass obj = new MyClass();

System.out.println("Default Value: " + obj.value);

}

}

Output: Default Value: 10