

1. What is a Constructor.

Sol. A constructor in Java is a special method that is used to initialize objects.

The constructor is called when an object of a class is created.

2. What is Constructor Chaining.

Sol. Constructor chaining is the process of calling a sequence of constructors. We can do it in two ways: by using `this()` keyword for chaining constructors in the same class. by using `super()` keyword for chaining constructors from the parent class.

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

Sol. A subclass can call a constructor defined by its superclass by use of the following form of `super`: `super(parameter-list)`; Here, `parameter-list` specifies any parameters needed by the constructor in the superclass.

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

Sol. Since constructor can only return the object to class, it's implicitly done by java runtime and we are not supposed to add a return type to it. If we add a return type to a constructor, then it will become a method of the class. This is the way java runtime distinguish between a normal method and a constructor.

5. What is No-arg constructor.

Sol. Here, the constructor does not accept any parameters. Hence, it is known as a no-arg constructor. Notice that we have declared the constructor as `private`. Once a constructor is declared `private`, it cannot be accessed from outside the class.

6. How is a No-argument constructor different from the default constructor.

Sol. If we do not create any constructor, the Java compiler automatically create a no-arg constructor during the execution of the program. This constructor is called default constructor.

7. When do we need Constructor overloading.

Sol. As constructor overloading enables the creation of the object of a specific class in several ways, it is most commonly used in Java programs based on the requirement of the programmer. With the use of constructor overloading, objects can be initialized with different data types.