1. A constructor in programming is a special method in a class that gets called when an object of the class is created. It is used to initialize the object's attributes and set its initial state.

2. Constructor chaining is a technique in which one constructor calls another constructor in the same class or a superclass. This is often used to reuse code for common initialization tasks.

3. No, you cannot directly call a subclass constructor from a superclass constructor. Constructors are responsible for initializing the current class, and they cannot create or initialize objects of subclasses. However, a subclass constructor will implicitly call its superclass constructor.

4. Constructors do not have return types, not even void. If you attempt to specify a return type for a constructor, it will result in a compilation error.

5. A "no-arg constructor" is a constructor that takes no arguments. It is a constructor with no parameters, used to create an object with default initial values.

6. A "no-argument constructor" and a "default constructor" essentially refer to the same thing. They are constructors that don't take any parameters and are provided by the compiler if no constructor is explicitly defined in a class.

7. Constructor overloading is needed when you want to create multiple constructors in a class with different sets of parameters. This allows objects of the class to be instantiated in various ways, providing flexibility to the developers.

8. A default constructor is a constructor that is automatically provided by the compiler if no other constructor is defined in a class. It initializes the object's attributes with default values. Here's an example in Java:

java

public class MyClass {

// Default constructor

public MyClass() {

// Initialization code here

}

}