1. What is a Constructor?

   A constructor is a special method in a class that is automatically invoked when an object of the class is created. It is used to initialize the object and set up the initial state by assigning values to the instance variables.It has the same name as the class.It does not have a return type.

2. What is Constructor Chaining?

   Constructor Chaining occurs when a constructor calls another constructor of the same class or the parent class. It is used to reuse code within constructors.

   - Within the same class: Using this() to call another constructor of the same class.
   - From the parent class: Useingsuper() to call a constructor of the superclass.

```
class Parent {
    Parent() {
        System.out.println("Parent Constructor");
    }
}


class Child extends Parent {
    Child() {
        this("Calling another constructor");
        System.out.println("Child Constructor");
    }
    Child(String message) {
        super();
        System.out.println(message);
```

```
    }
}
```

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

   No, a subclass constructor cannot be directly called from a superclass constructor. Constructors flow from parent to child, meaning the super() call in the subclass invokes the superclass constructor, not the other way around.

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

   If we define a return type for a method with the same name as the class, it will no longer be treated as a constructor. Instead, it will be a regular method.

5. What is No-arg constructor?

   A no-argument constructor is a constructor that does not take any parameters. It is used to create objects with default values.

   Example :

```
class A {
        int a;
        int b;

        public A() {
                this.a = 0;
                this.b = 0;
        }
}
```

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

   No-argument Constructor: We can include custom logic in no-argument constructor for the initialization purpose.
   Default Constructor: It is implicitly provided by the compiler if no other constructor is defined by the programmer.

7. When do we need Constructor Overloading?

   Constructor Overloading is used when we want to create objects in multiple ways with different initializations (customized initialization). It involves defining multiple constructors with different parameter lists.

   Example :

```
class Person {
    String name;
    int age;

    Person() {
        this.name = "Unknown";
        this.age = 0;
    }

    Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
}
```

8. What is Default constructor Explain with an Example

   A default constructor is a constructor provided by the compiler when no other constructors are explicitly defined. It initializes the object with default values (e.g., null for objects, 0 for numbers, false for booleans, etc.)