Java Interview Concept

- Cloning in Java
- Shallow Cloning vs Deep Cloning
- Why No Deep Cloning for String & Wrapper Classes?

What is Cloning?

- Creating an exact copy of an existing object in Java
- Using clone() method from Object class

Types of Cloning

- 1. Shallow Cloning
- 2. Deep Cloning

Shallow Cloning

- - Default behavior of clone()
- - Copies primitive fields & references of objects
- - Changes to referenced object reflects in clone

Deep Cloning

- - Manual process
- - Copies primitives + creates new copies of referenced objects
- - Achieved by overriding clone() method or using libraries

Example: Shallow vs Deep Cloning

- Shallow Copy: copies reference
- Deep Copy: copies object content

Why No Deep Cloning for String & Wrapper Classes?

- - String & Wrapper classes are immutable
- - Value can't be changed once created
- - So, shallow cloning is enough

Example

```
class Employee implements Cloneable {
 String name;
 Integer age;
  protected Object clone() {
      return super.clone();
  }
}
```

Summary

- Deep Cloning → Needed for mutable objects
- Shallow Cloning → Enough for immutable objects (String, Integer, etc.)