**Core Java:**

**Deep Immutability**

An object is **deeply immutable** when it's impossible to change its state in any way after construction.

rules for creating a deeply immutable class:

1. Make the class final so it cannot be extended.
2. Make all fields private and final.
3. Don't provide any "setter" methods.
4. In the constructor, perform a **defensive copy** of any mutable input objects.
5. In any "getter" methods, return a **copy** or an **unmodifiable view** of any internal mutable objects.

**Defensive copy is required only for mutable object.**

**Shallow Copy**

Whenever we use default implementation of clone method we get shallow copy of object.

If only primitive type fields or Immutable objects are there then there is no difference between shallow and deep copy in Java.