**Spring Core**

Q) How to allow prototype behavior into Singleton class? same like other fields how we will make immutable to make them as immutable like list, map , custom objects?

I need to get prototype class from singleton. I found that **method injection** is the way to go, but I don't really know how to use spring **@Lookup** annotation.

Let’s say we have Employee class and that we made as Singleton class by following

* Making class final and private constructor
* Create static reference variable of the same class instance with private final modifiers
* Static block which will return Singleton class object and which has synchronized block with double checking mechanisam to handle multiple threads.
* Deep cloning to all the mutable fields.
* No setter methods
* We are returning new objects with existing data through setter methods
* Same thing we are doing in while creating the object through constructor
* Implement clonable interface and handle CloneNotSupportedException inside your single class, if anybody tried to clone your object it will give the original reference when it was created during first time, because you will overide clone method by giving super.clone();

Let’s say we have Student object inside Employee class and that is not prototype and that is designed like Prototype annotation form spring, that means if any body ask spring it will provide new object every time and our Employee object is loosing Singleton behavior ?

How we will overcome this problem?

We have an annotation like **@Lookup,** if you provide that it will check if object is not exist then only it will create new object else it will return existing object reference only.