

1. Container initialization
2. Bean class
3. Bean is given to the container (TOH container ne kya kiya --- Instantiate , Dependency Injection)

LATER.....

4. we ask the container to give us the bean object
5. we get that object and we check its properties

Exercise 1 = Add sysout in constructor and setters and observe the output.

Container calls the default constructor (**Instantiate**) + the setters (**Dependency Injection**)

Exercise 2 = Ask the container for one more MessageBean in the Demo class

See the contents of the next bean

OBSERVE !!! Though we call getBean twice constructor and setters are called only once!!!

Only object is there and its reference is returned every time

Change the senderName of bean1 using setter

See the values of both beans OBSERVE ----- sender of both beans changes !!!

Try doing it 3 times or 4 times ----- OBSERVE!!!!!!!

CONCLUSION :

Bean1----->**Obj**<-----bean2

SINGLETON pattern of INSTANTIATION ---- only one object created !!!

So if we want new object every time ---- can we tell the container ? YES through configuration.

Solution : we set the **scope attribute** in **bean tag** to **prototype**

Exercise : try changing scope to prototype . Run the Demo Observe : Two bean instances are created

Exercise : try changing scope to singleton. Run the Demo Observe : single bean instance are created

Exercise : we will add a property --- Address to the MessageBean
Address is another Bean

Programmer communicated with the Container through **beans.xml** !!!!!

XML based configuration !!!!

Java Based Configuration

We used AnnotationConfigApplicationContext to create the container

We passed a JavaConfig class to the container

Inside the Java Config class we added methods annotated with @Bean tag

Exercise --- write Demo4 and JavaConfig as discussed in class

Create multiple message beans Observe how many message beans are created
and how many Address beans are created

Annotation Based Configuration

--- **We do the configuration on the class itself !!!**

--- Let us write a class BookBean ---- name , cost , MyDateBean dateOfPublication

