* IOC Container Vs Bean Factory Vs Application Context

**Inversion Of Control**: The control of creating and instantiating the dependencies is transferred to the spring framework. We do not need to use the “new” keyword anymore. We just need to autowire the dependencies. Spring container/IOC Container will take it from there and will manage all the beans and their dependencies.

There are two implementations of IOC Container: Bean Factory & Application Context

**Bean Factory** provides basic management of beans and autowiring of dependencies via spring-core.jar. Most common implementation of BeanFactory is XmlBeanFactory

Resource res = new FileSystemResource("beans.xml");

XmlBeanFactory factory = new XmlBeanFactory(res);

or

ClassPathResource res = new ClassPathResource("beans.xml");

XmlBeanFactory factory = new XmlBeanFactory(res);

or

ClassPathXmlApplicationContext appContext = new ClassPathXmlApplicationContext(

new String[] {"applicationContext.xml", "applicationContext-part2.xml"});

// of course, an ApplicationContext is just a BeanFactory

BeanFactory factory = (BeanFactory) appContext;

Application Context = Bean Factory + Spring’s AOP features + i18n capabilities + WebApplicationContext

Spring recommends to use Application Context in most of the scenarios because Application context is the super set of Bean Factory. Though, we may use Bean Factory when memory usage is a concern