Interview Questions link:

http://www.journaldev.com/2696/spring-interview-questions-and-answers

http://howtodoinjava.com/2013/03/19/inversion-of-control-ioc-and-dependency-injection-di-patterns-in-spring-framework-and-related-interview-questions/

http://www.tutorialspoint.com/spring/spring_interview_questions.htm

http://www.javacodegeeks.com/2014/05/spring-interview-questions-and-answers.html

http://www.javatpoint.com/spring-interview-questions

http://crackaninterview.com/spring-interview-questions/

http://www.withoutbook.com/Technology.php?tech=23

http://www.journaldev.com/2696/spring-interview-questions-and-answers

http://howtodoinjava.com/2013/03/20/different-spring-3-ioc-containers-with-example/

http://www.javatpoint.com/ioc-container

http://howtodoinjava.com/2013/03/19/inversion-of-control-ioc-and-dependency-injection-di-patterns-in-spring-framework-and-related-interview-questions/

http://www.journaldev.com/2410/spring-dependency-injection-example-with-annotations-and-xml-configuration

IoC Container

The IoC container is responsible to instantiate, configure and assemble the objects. The IoC container gets informations from the XML file and works accordingly. The main tasks performed by IoC container are:

- to instantiate the application class
- to configure the object
- to assemble the dependencies between the objects

There are two types of IoC containers. They are:

1. **BeanFactory**

ApplicationContext

1. **Inversion of Control** (IoC) is the mechanism to achieve loose-coupling between Objects dependencies. To achieve loose coupling and dynamic binding of the objects at runtime, the objects define their dependencies that are being injected by other assembler objects. Spring IoC container is the program that injects dependencies into an object and make it ready for our use.

   Spring Framework IoC container classes are part of `org.springframework.beans` and `org.springframework.context` packages and provides us different ways to decouple the object dependencies.

   Some of the useful ApplicationContext implementations that we use are;

   - `AnnotationConfigApplicationContext`: For standalone java applications using annotations based configuration.

- `ClassPathXmlApplicationContext`: For standalone java applications using XML based configuration.
- `FileSystemXmlApplicationContext`: Similar to ClassPathXmlApplicationContext except that the xml configuration file can be loaded from anywhere in the file system.
- `AnnotationConfigWebApplicationContext` and `XmlWebApplicationContext` for web applications.