

Spring Fundamentals

This is test to test the knowledge on Spring. The questions are based on new version of Spring 2.5 and Spring 3.0. and covers Spring Basics, core and transaction.

Category	Technical Skill / Interview questions
Keywords	Spring, Spring Core, spring interview questions
Tags	java spring
Questions	13 - Free questions
Test takers	3544
Average score	48.9
Rating	 (4/5)
Guru	j2eeGuru

Test : Spring Fundamentals

Question :1 I want to provide the MessageSource functionality to application, providing access to messages in i18n-style in my application. Which one should I use ?

Correct Answer	Your Selection	Answer	Explanation
✓	<input type="radio"/>	ApplicationContext	ApplicationContext interface, located in the org.springframework.context package provides MessageSource functionality
	<input type="radio"/>	Properties File	
	<input type="radio"/>	Servlet Context	

Question :2 What do you understand by Dependency Injection ?

Correct Answer	Your Selection	Answer	Explanation
✓	<input type="checkbox"/>	It is design pattern of how objects are related.	
	<input type="checkbox"/>	The property of the server on how it injects the object	
✓	<input type="checkbox"/>	Since all the dependency of an object has defined , a standalone application can be tested in totality by injecting the dependent objects.	

Question :3 What are the types of injection supported by Spring ?

Correct Answer	Your Selection	Answer	Explanation
	<input type="radio"/>	Constructor Injection only	
✓	<input type="radio"/>	Constructor Injection and Setter Injection	
	<input type="radio"/>	Constructor Injection , Setter Injection before 2.0 and Interface Injection with 2.0 and above version	
	<input type="radio"/>	Constructor Injection , Setter Injection and Interface Injection	

Question :4 Which of the following features are provided when using the spring templating classes such JdbcTemplate or HibernateTemplate ?

Correct Answer	Your Selection	Answer	Explanation
✓	<input type="checkbox"/>	Automatic Transaction Management	
✓	<input type="checkbox"/>	Exception Handling	
	<input type="checkbox"/>	Load Balancing	
	<input type="checkbox"/>	Cache Management	
✓	<input type="checkbox"/>	Resource look up	

Question :5

What is understood by **Inversion of Control**(IOC) ?

Correct Answer	Your Selection	Answer	Explanation
✓	<input type="checkbox"/>	<p>The object define their dependencies and the container injects the dependencies when it creates the bean</p>	<p>From Spring Documentation : The basic principle behind Dependency Injection (DI) is that objects define their dependencies (that is to say the other objects they work with) only through constructor arguments, arguments to a factory method, or properties which are set on the object instance after it has been constructed or returned from a factory method. Then, it is the job of the container to actually inject those dependencies when it creates the bean. This is fundamentally the inverse, hence the name Inversion of Control (IoC), of the bean itself being in control of instantiating or locating its dependencies on its own using direct construction of classes, or something like the Service Locator pattern.</p>
	<input type="checkbox"/>	<p>It is a unique behavior of Spring Framework</p>	
✓	<input type="checkbox"/>	<p>This is a generic concept which Spring implements</p>	

Question :6

What is a 'bean' in Spring ?

Correct Answer	Your Selection	Answer	Explanation
	<input type="radio"/>	<p>It is a special component of the IOC container</p>	
	<input type="radio"/>	<p>It is the regular java bean with some special features and capabilities in spring</p>	
✓	<input type="radio"/>	<p>A bean is simply an object that is instantiated, assembled and otherwise managed by a Spring IoC container; other than that, there is nothing special about a bean</p>	

Question :7 Can dependency injection be lazy instantiated ?

Correct Answer	Your Selection	Answer	Explanation
	<input type="radio"/>	No	
✓	<input type="radio"/>	Yes	By Specifying lazy="true" annotation on the bean, it can be instantiated only when needed

Question :8 Spring's Transaction management feature can fully replace all the features of EJB's Container managed transaction

Correct Answer	Your Selection	Answer	Explanation
✓	<input type="radio"/>	Yes except the global transactions	Unlike EJB CMT, which is tied to JTA, the Spring Framework's declarative transaction management works in any environment. It can work with JDBC, JDO, Hibernate or other transactions under the covers, with configuration changes only. It does have the capability of global transactions.
	<input type="radio"/>	False	
	<input type="radio"/>	True , no exceptions	

Question :9

An application server is required for Spring's Transaction mechanism to run.

Is this statement

Correct Answer	Your Selection	Answer	Explanation
	<input type="radio"/>	<p>True</p>	
✓	<input type="radio"/>	<p>False</p>	<p>Spring provides transactional capabilities which can run in any web server and do not need a full blown application server</p>

Question :10

Look at the code below .

```
<bean id = "helloWorldSample" ` class="com.helloworld.samples.HelloWorld" depends-on
= "setuphelloClassId">
<property name="myString">
Hello World Program
</property>
<property name="dependentClassId">
<ref bean = "firstDependentClass"/>
</property>
</bean>

<bean id = "dependentClass" class = "com.helloworld.dependent.DependentClass" destroy-
method="cleanupMethod">
<property name="dependentString">
</property>

</bean>

<bean id = "setuphelloClassId" class = "com.helloworld.config.setupHelloClass">
</bean>
```

Which of the following answers is correct ?

Correct Answer	Your Selection	Answer	Explanation
	<input type="radio"/>	The bean for helloWorldSample class depends on the setuphelloClassId class and the property "dependentClassId" is not a field for the Helloworld bean with a reference to the dependentClass class.	
✓	<input checked="" type="radio"/>	The bean definition for HelloWorld depends on the setupHelloClass and the property "dependentClassId" is a field for the Helloworld bean with a reference to the firstDependentClass bean.	
	<input type="radio"/>	The bean for HelloWorld class depends on the setupHelloClass property and the property class "dependentClassId" is a field for the Helloworld bean with a reference to the DependentClass property.	
	<input type="radio"/>	The bean definition for HelloWorld depends on the setuphelloClassId and the property "firstDependentClass" is a field for the Helloworld bean with a reference to the DependentClass.	

Question :11

The basic principle behind Dependency Injection(DI) is that the objects define their dependencies . What are the valid ways is which an object define its dependency ?

Correct Answer	Your Selection	Answer	Explanation
✓	<input type="checkbox"/>	Through Constructor arguments	
✓	<input type="checkbox"/>	Arguments to a factory method,	
✓	<input type="checkbox"/>	properties which are set on the object instance after it has been constructed or returned from a factory method	
	<input type="checkbox"/>	xml file	This is how you can define in spring

Question :12

Can you instantiate a static inner class in Spring ?

Correct Answer	Your Selection	Answer	Explanation
	<input type="radio"/>	No. Static classes are not supported	
	<input type="radio"/>	Yes. They are instantiated by default with the class	
✓	<input type="radio"/>	Yes. But you will have to use the binary name of the class	For example, if you have a class called Foo in the com.example package, and this Foo class has a static inner class called Bar, the value of the 'class' attribute on a bean definition would be com.example.Foo\$Bar

Question :13

What is dependency injection ?

Correct Answer	Your Selection	Answer	Explanation
	<input type="radio"/>	A design pattern where in dependent objects are defined in an xml for easy maintainability	
✓	<input type="radio"/>	This is a design pattern whereby collaborators(dependent objects) are passed into an object from the outside at the time application is initialized	If you need a decent insight into IoC and DI, please do refer to said article: http://martinfowler.com/articles/injection.html .
	<input type="radio"/>	The unique feature of spring which allows object to be injected during compile time.	