1. What is dependency injection?

**A It is a design pattern that implements the Inversion of control (IoC) pattern for software applications.**

**B** It is one of the Spring modules.

**C** It is a technique to get dependencies from any project.

**D** It is used to promote loose coupling in code.

Answer

**A**

**Dependency Injection is a design pattern that implements the Inversion of Control pattern for software applications.**

2. What types of dependency injection does Spring support?

**A Based on the constructor and setters**

**B** Based on the constructor, setters, and getters

**C** Based on setters, getters, and properties

**D** Based on the constructor, setters, and properties

Answer

**A**

**Spring supports constructor-based and setters-based injections.**

3. Which of the following statements is correct regarding the Spring Framework?

**A** The Spring Framework is a heavy-weight solution.

**B** The Spring Framework is a light-weight solution.

**C** Both **A** and **B** are true.

**D** None of the above

Answer

**B**

**There are many reasons why spring is a lightweight framework.**

* **Spring provides you with different modules and allows you to use whatever works best for you. Ideally, the spring JAR file is only 2-3 MB.**
* **If you compare Spring with EJB, you have to write much less code and configurations. The beauty of Spring is that you can focus on the business logic whereas in EJB, you have to write a lot of code along with the business logic which makes it cumbersome and tightly coupled.**
* **With Spring, you play with POJO which does not depend on a Framework and improves the testability of your code.**
* **Spring offers seamless integration with frameworks, third-party libraries, etc.**

4. AOP is part of Core Container in the Spring Framework?

**A** True

**B** False

Answer

**B**

**AOP (Aspect-oriented programming) is not part of Core Container, is one of the key components of the Spring Framework, is a programming approach that allows properties of a program to determine how they are compiled into an executable program. AOP complements the OOP rules by also providing modularity. AOP breaks down the logic of the program into distinct parts called “concerns”. This increases modularity by cross-cutting concerns.**

5. The Expression Language is part of the Core Container in the Spring?

**A** True

**B** False

Answer

**A**

**SpEL stands for Spring Expression Language which is part of Core Container. It is a powerful expression language that supports queries and manipulation of an object graph at bean creation or runtime. It is similar to other expression languages such as JSP EL, OGNL, MVEL and JBoss EL, etc., with some additional features such as method calling and basic string modeling.**

6. Can we integrate spring with struts?

**A** Yes

**B** No

Answer

**A**

**Spring is a popular web framework for easy integration with many popular web libraries. So the question is: why do we need Spring when we have Struts? Spring is more than an MVC framework: it offers many other benefits that are not available in Struts.**

7. Spring is a \_\_\_\_\_\_\_\_\_\_\_ framework?

**A** free

**B** open source

**C** under license

**D** proprietary

Answer

**B**

**The Spring Framework is open source.**

8. What are the different types of Bean injection?

**A** constructor and setter

**B** constructor and getter

**C** getter and setter

**D** setter, getter and constructor

Answer

**A**

**Spring supports both setter and constructor injection.**

9. Controller in Spring is a(n)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_?

**A** abstract class

**B** concrete class

**C** final class

**D** interface

Answer

**D**

**Another way to create a controller in Spring framework is to have a class implement the Controller interface. For example:**

**import** *javax.servlet.http.HttpServletRequest*;

**import** *javax.servlet.http.HttpServletResponse*;

**import** *org.springframework.web.servlet.ModelAndView*;

**import** *org.springframework.web.servlet.mvc.Controller*;

**public** **class** ControllerExample **implements** Controller **{**

@Override

**public** ModelAndView handleRequest**(**HttpServletRequest rq,

HttpServletResponse rp**)** **throws** Exception **{**

System.out.println**(**"Welcome to StackHowTo"**)**;

**return** new ModelAndView**(**"StackHowTo"**)**;

**}**

**}**

10. Which exception class is bound to all the exceptions thrown in Spring applications?

**A** ArrayIndexOutofBound

**B** DataAccessException

**C** NullPointerException

**D** SpringException

Answer

**B**

**DataAccessException is an exception defined by the Spring framework. There are two things to note about DataAccessException. First of all, this is an exception. Therefore, application code that uses a data access object is not required to wrap every call with a try-catch block, as is the case in JDBC entity beans. and EJB 2.x. Second, DataAccessException is useful because it encapsulates the specific exception classes used by the underlying persistence technology and thus keeps the rest of the application independent of the persistence layer.**

1. Dependency injection or IOC is a \_\_\_\_\_\_\_\_\_\_\_\_\_?

**A Design Pattern**

**B** Framework

**C** Java Module

**D** ORM Framework

Answer

**A**

**In software engineering, dependency injection is a technique by which one object provides dependencies to another object. A dependency is a usable object (a service). An injection is the passage of a dependency to a dependent object (a client) that would use it.**

2. Beans defined in the spring framework are by default \_\_\_\_\_\_?

**A** Abstract

**B** Singleton

**C** Final

**D** Initialized

Answer

**B**

**The word “Singleton” in Spring is used for a bean scope, which means that the bean will only be created once for the entire application. Singleton usually stands for the GOF (Gang of Four) pattern. It is an object oriented model ensuring that there will only be one instance of a class.**

3. Which of the following is not a Spring module?

**A** AOP

**B** O/R Integration

**C** Spring MVC

**D** HTML/JSP

Answer

**D**

**Spring framework includes many modules such as core, beans, context, expression language, AOP, Aspects, JDBC, ORM, OXM, JMS, Transaction, Web, Servlet, Struts, etc.**

**Source :**[**docs.spring.io**](https://docs.spring.io/spring/docs/3.0.0.M4/reference/html/ch01s02.html)

4. Which of the following statements is true?

**A** ApplicationContext implements the BeanFactory

**B** **ApplicationContext inherits from BeanFactory**

**C** BeanFactory inherits from ApplicationContext

**D** BeanFactory implements ApplicationContext

Answer

**B**

* **ApplicationContext is a more preferred way than BeanFactory**
* **In newer versions of Spring, BeanFactory is replaced by ApplicationContext. But BeanFactory still exists for backward compatibility**
* **ApplicationContext extends BeanFactory**

**has the following advantages:**

* 1. **It supports internationalization of text messages**
  2. **It supports event publishing for registered listeners**
  3. **Access to resources such as URLs and files**

5. How to get the DAO object in the spring framework?

**A** Use the “new” keyword

**B** Using Spring Dependency Injection

**C** Both **A** and **B** are true.

**D** None of the above

Answer

**B**

**If a class A expects a data access object (DAO) to receive data from a database, you can easily create another test object(mock) for a database connection and inject that object into A to test A without having a database connection. Dependency injection based design is possible with the Java standard. Spring simply simplifies the use of dependency injection by providing a standard way to provide the configuration and by managing the references of the created objects.**

6. In which version of spring have the features of Java 5 been introduced?

**A** Spring 1.0

**B** Spring 2.0

**C** Spring 2.5

**D** Spring 3.0

Answer

**D**

**The main API of the Spring 3.0 framework uses Java 5; therefore, Java5 or later is required to run Spring 3.0 applications. Java 5 features, such as generic types, annotations, etc, can be used in Spring 3.0 framework applications. Spring 3.0 is fully compatible with JEE1.4 and JEE5 models.**

7. In which version of spring has Spring Expression Language been supported?

**A** Spring 1.0

**B** Spring 2.0

**C** Spring 2.5

**D** Spring 3.0

Answer

**D**

**Spring 3.0 introduces Spring Expression Language (SpEL), a powerful expression language that supports querying and manipulating object graph at runtime. With SpEL, Spring applications can overcome the limitation of using only fixed values in configuration files. It can be used for bean definitions in XML-based and annotation-based configurations.**

8. Which class does the IoC container represent?

**A** **ApplicationContext**

**B** ServletContext

**C** RootContext

**D** WebApplicationContext

Answer

**A**

**The IoC (Inversion of Control) container is responsible for instantiating, configuring, and aggregating objects. The IoC container obtains information from the XML file and operates accordingly. The main tasks performed by the IoC container are as follows:**

* **Instantiate the application class**
* **Configure the object**
* **Aggregate dependencies between objects**

9.How to use “idref” in the spring framework?

**A** With setter method

**B** With getter method

**C** With setter method and constructor argument

**D** With getter method and constructor argument

Answer

**C**

**In the Spring framework, idref consists of transmitting the identifier of a bean to another bean. Example:**

**<idref bean="PointA">**

**. idrefs can be considered of type Strings.**

**When using “idref”, the parameters of the Setter method or the constructor that we define/inject must be of type Strings.**

10. What component is part of the data access layer in the Spring Framework?

**A** Aspects

**B** JMS

**C** Beans

**D** Core

**E** Web

**F** Servelet

**G** Expression language

Answer

**B**

**Java applications that use Java Message Service (JMS) are called JMS clients. A JMS client can create, send, receive, and read messages. The clients who send messages are called producers and those who receive messages are called consumers.**

2. How to enable annotations in Spring?

**A** Add

<annotation-context: config />

 to the bean configuration.

**B** Add

<annotation-config />

 to the bean configuration.

**C** Add

<annotation-context-config />

 to the bean configuration.

**D** Add

<context: annotation-config />

 to the bean configuration.

Answer

**D**

**Add**

**<contexte: annotation-config />**

**to the bean configuration to enable the use of annotations.**

3. Which ORM does Spring support?

**A** Hibernate

**B** iBatis

**C** JPA

**D** All the answers are true

**E** None of the above

Answer

**D**

**Spring supports most ORMs, including Hibernate, JDO, TopLink, iBATIS and JPA.**

4. A bean must have an id attribute in the bean configuration file?

**A** True

**B** False

Answer

**B**

**The id is not a mandatory attribute in the bean configuration file.**

5. The \_\_\_\_\_\_\_\_ class can be extended to create a custom event in Spring.

**A** SpringEvent

**B** Event

**C** ApplicationEvent

**D** None of the above

Answer

**C**

**ApplicationEvent is used to create custom events.**

6. How to use <ref> tag in the Spring framework?

**A**<ref> tag is used with the bean ID.

**B**<ref> tag is used with a String value.

**C**Both**A**and**B**are true.

**D**None of the above

Answer

**A**

**In Spring, we must use the <ref> tag to inform the Spring container of the object’s dependency.**

**In Spring, beans can “access” each other by specifying their references in the same configuration file or in a different configuration file. In Spring, we can write several configuration xml files. Our associated bean can be in the same xml or in another xml file.**

**Example : person-bean.xml**

**<?xml version="1.0" encoding="UTF-8"?>**

**<beans>**

**<bean** id="personDetails" class="com.jwt.spring.PersonImpl"**/>**

**</beans>**

**Now, in the XML file below, we refer to “personDetails” configured in the person-bean.xml file. We must therefore use the “ref” tag with the “bean” attribute i.e.**

**<ref bean =" personDetails "/>**

**.**

**company-bean.xml**

**<?xml version="1.0" encoding="UTF-8"?>**

**<beans>**

....

**<bean** id="companyId" class="com.jwt.spring.Company"**>**

**<property** name="perdetails"**>**

**<ref** bean="personDetails"**/>**

**</property>**

**</bean>**

....

**</beans>**

7. How to define a bean in Spring?

**A** Use only the following

**<property** **/>**

**B** Use only the following

**<constructor-arg** **/>**

**C** Use

**<property** **/>**

 or

**<constructor-arg** **/>**

.

**D** None of the above

Answer

**C**

**Example:**

**<bean** id="objetB" scope="request" class="a.b.c.classeB"**/>**

**<bean** id="objetA" scope="request" class="a.b.c.classeA"**>**

**<constructor-arg** ref="objetB" **/>**

<!-- OR -->

**<property** name="bRef" ref="objetB" **/>**

**</bean>**

8. Which property is replaced by

p-namespace

 in the Spring framewrok?

**A**

**<property** **/>**

**B**

**<constructor-arg** **/>**

   
   
Answer

**A**

**In spring, p-namespace is an XML shortcut to inject a dependency into the bean. p-namespace replaces the <property> tag of XML. p-namespace has no XSD definition and exists only in the spring core. We can directly assign the attribute name of the class with p-namespace in the bean tag. We can use p-namespace instead of the <property> tag in spring XML. It is easy and clear to use, which will increase the readability of the XML context. Suppose we have the following <bean> definition in XML.**

**<bean** id="per" class="x.y.z.Person"**>**

**<property** name="name" value="Alex"**/>**

**<property** name="address" value="California"**/>**

**</bean>**

**We can change the <property> tag using p-namespace as follows.**

**<bean** id="per" class="x.y.z.Person" p:name="Alex" p:address="California"**/>**

9. Which property is replaced by

c-namespace

 in the Spring framewrok?

**A**

**<property** **/>**

**B**

**<constructor-arg** **/>**

Answer

**B**

**c-namespace was introduced in spring 3.1. It replaces the old style of constructor-arg. The bean that needs to be configured with c-namespace must have a constructor to accept these arguments.**

**Example – The old style :**

**<bean** id="std" class="x.y.z.Student"**>**

**<constructor-arg** name="name" value="BOB"**/>**

**<constructor-arg** name="age" value="25"**/>**

**</bean>**

**We can replace constructor-arg using c-namespace as follows.**

**<bean** id="std" class="x.y.z.Student" c:name="BOB" c:age="25"**/>**

10. What is the purpose of “ApplicationContextAware” in Spring?

**A** The dependency injection is performed.

**B** Makes a bean aware of the container.

Answer

**B**

**The Bean implementing the “ApplicationContextAware” interface can get the current context of the application and can be used to call any service from the application context.**

1. How many proxy types are there in the Spring framework?

**A** One

**B** Two

**C** Three

**D** Four

Answer

**B**

1. **Static**
2. **Dynamic**

2. What are the features offered by Spring?

**A** Resource management

**B** Resource unwrapping

**C** Exception handling

**D** Transaction participation

**E** All the answers are true

Answer

**E**

3. Choose the correct option:

**A** The first version was written by Rod Johnson, released in October 2002.

**B** The framework was first released under the Apache 2.0 license in June 2003.

**C** The Spring 1.2.6 framework won a productivity award and an innovation award in 2006.

**D** All the answers are true

Answer

4. Choose the correct option:

**A** The Spring framework is an open source application.

**B** The Spring framework is a Java platform.

**C** The Spring framework is used by the .NET framework.

**D** All the answers are true

Answer

**D**

5. Spring is an MVC framework based on requests?

**A** Yes

**B** No

Answer

**A**

6. What does MVC mean for Spring?

**A** Model view Controller

**B** Middle view Controller

**C** Module view Controller

**D** None of the above

Answer

**A**

7. What does AOP mean to Spring?

**A** Aspect Oriented Programs

**B** Aspect Oriented Programming

**C** Aspect Oriente Programming

**D** None of the above

Answer

**B**

8. How to externalize constants from a Spring configuration file or a Spring annotation in the .properties file?

**A** Using the tag

util:constant

**B** By declaring the post-processor of the ConstantPlaceholderConfigurer bean

**C** Using the tag

contexte:property-placeholder

**D** Using the namespace

c:

Answer

**C**

1. **The**

**<util:constant static-field="constant\_name"/>**

**tag is used to reference a Java constant or enumeration in a spring configuration file.**

1. **ConstantPlaceholderConfigurer does not exist. You can think of the PropertyPlaceholderConfigurer post-processor.**
2. **The**

**<context:property-placeholder location = "file: /maApp.properties" />**

**tag enables placeholder replacement $ {…}, resolved from the specified property file.**

1. **The c: namespace is intended to simplify the constructor syntax (since Spring 3.1) and does not provide such a feature.**

9. How many dynamic proxy types are available in Spring?

**A** One

**B** Two

**C** Three

**D** Four  
   
   
Answer

**B**

1. **JDK Dynamic Proxy**
2. **CGLIB Dynamic Proxy**

10. Select the correct statement to reference a Spring configuration file inside the com.stackhowto.myapp package in the example below?

ApplicationContext context = new

ClassPathXmlApplicationContext**(**"classpath:/com.stackhowto.myapp.config.xml"**)**;

**A** The classpath: prefix can be omitted

**B** The name of the package with the dot is not well formatted

**C** The slash before **com.stackhowto** can be omitted

**D** All the answers are true

Answer

**D**

## What does AOP stand for in Spring?

a) Advanced Object Programming

b) Aspect-Oriented Programming

c) Application-Oriented Programming

d) Asynchronous Operation Protocol

Click to View Answer and Explanation

### Answer:

b) Aspect-Oriented Programming

### Explanation:

AOP stands for Aspect-Oriented Programming, which deals with cross-cutting concerns in applications.

## 2. Which of the following can be considered a cross-cutting concern?

a) Business logic

b) Authentication

c) Data validation

d) Feature toggles

Click to View Answer and Explanation

### Answer:

b) Authentication

### Explanation:

Cross-cutting concerns are aspects that affect multiple parts of an application. Authentication, logging, and transactions are typical examples.

## 3. In Spring AOP, what is an "advice"?

a) A recommendation given by the Spring team

b) A method that handles a cross-cutting concern

c) An error thrown by the AOP process

d) A configuration recommendation

Click to View Answer and Explanation

### Answer:

b) A method that handles a cross-cutting concern

### Explanation:

An advice is an action taken by an aspect for a particular join point (like method execution).

## 4. Which type of advice runs after the advised method completes without throwing any exception?

a) Before advice

b) Around advice

c) After returning advice

d) After throwing advice

Click to View Answer and Explanation

### Answer:

c) After returning advice

### Explanation:

"After returning advice" is executed after the advised method completes successfully without throwing any exception.

## 5. Which of the following is NOT a type of advice in Spring AOP?

a) Around advice

b) Below advice

c) Before advice

d) After advice

Click to View Answer and Explanation

### Answer:

b) Below advice

### Explanation:

"Below advice" isn't a type of advice in Spring AOP. The main advice types are Before, After, After returning, After throwing, and Around.

## 6. In Spring AOP, what defines a point where an advice should be executed?

a) Aspect

b) Join point

c) Pointcut

d) Advisor

Click to View Answer and Explanation

### Answer:

c) Pointcut

### Explanation:

Pointcut defines the expressions that determine which join points (like method executions) should be matched for advice execution.

## 7. Which of the following is NOT true about Spring AOP?

a) It supports only method execution join points

b) It uses byte-code weaving

c) It is implemented using proxies

d) It's integrated with Spring's transaction management

Click to View Answer and Explanation

### Answer:

b) It uses byte-code weaving

### Explanation:

Spring AOP is proxy-based and does not use byte-code weaving. Full-fledged AOP frameworks like AspectJ use byte-code weaving.

## 8. Which annotation can be used to define an aspect in Spring AOP?

a) @AspectDef

b) @AspectJ

c) @PointcutDef

d) @Aspect

Click to View Answer and Explanation

### Answer:

d) @Aspect

### Explanation:

The @Aspect annotation is used in Spring AOP to define a class as an aspect.

## 9. Which advice runs irrespective of a method's successful execution or an exception being thrown?

a) Before advice

b) Around advice

c) After advice

d) After returning advice

Click to View Answer and Explanation

### Answer:

c) After advice

### Explanation:

"After advice" runs after a method's execution, regardless of its outcome.

## 10. What's the primary module for AOP in Spring?

a) spring-core

b) spring-aop

c) spring-aspect

d) spring-beans

Click to View Answer and Explanation

### Answer:

b) spring-aop

### Explanation:

The spring-aop module provides the AOP integrations for Spring applications.

## 11. Which AOP framework does Spring AOP integrate with for advanced AOP features?

a) JAspect

b) AspectC

c) AspectJ

d) WeaveJ

Click to View Answer and Explanation

### Answer:

c) AspectJ

### Explanation:

Spring AOP provides integration with AspectJ for advanced AOP use-cases beyond its proxy-based model.

## 12. What does the "weaving" process in AOP refer to?

a) Creating XML configurations

b) The process of linking aspects with other application types

c) Generating proxies

d) Scanning components

Click to View Answer and Explanation

### Answer:

b) The process of linking aspects with other application types

### Explanation:

Weaving is the process through which aspects are integrated into the application, either at compile-time, load-time, or runtime.

## 13. For a method execution, which advice wraps the execution and can decide whether to proceed with the execution?

a) Before advice

b) Around advice

c) After advice

d) After returning advice

Click to View Answer and Explanation

### Answer:

b) Around advice

### Explanation:

"Around advice" surrounds the advised method execution and provides developers with the capability to customize behavior before and after the method invocation.

## 14. In Spring AOP, what is the purpose of the "target" object?

a) It defines the AOP proxy

b) It contains metadata about the AOP process

c) It is the actual object being advised

d) It represents the aspect instance

Click to View Answer and Explanation

### Answer:

c) It is the actual object being advised

### Explanation:

In Spring AOP, the "target" object is the original object, and the advised object is the proxy instance.

## 15. Which of the following ensures that an advice runs regardless of the outcome of the join point?

a) @AfterReturning

b) @Around

c) @After

d) @Before

Click to View Answer and Explanation

### Answer:

c) @After

### Explanation:

The @After annotation ensures that the corresponding advice runs no matter how the join point completes (either normally or due to an exception).