**1**. What is the scope of inner beans.

a. Singleton

b. Prototype

c. Request

d. None Of the above

**Ans**-- b

**2**. Inner Beans can be retrieved by it’s name.

a. True

b. False

**Ans**-- b

**3**. public​ abstract​ class​ Product {

public​ static​ final​ Product AAA = new​ Battery("AAA", 2.5);

public​ static​ final​ Product CDRW = new​ Disc("CD-RW", 1.5);

...

}

<beans ...>

<bean id="aaa" class​="org.springframework.beans.factory.config.

FieldRetrievingFactoryBean">

<property name="staticField">

<value>com.shop.Product.AAA</value>

</property>

</bean>

<bean id="cdrw" class​="org.springframework.beans.factory.config.

FieldRetrievingFactoryBean">

<property> name="staticField"

valuecom.shop.Product.CDRW/value

</property>

</bean>

</beans>

B) Product aaa = com.shop.Product.AAA;

Product cdrw = com.shop.Product.CDRW;

a)A and B are equivalent

b) A and B provides different functionality

c) Runtime Error in A

d) Exception in B

**ans**-- a

4.Is this bean configuration correct?

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:util="http://www.springframework.org/schema/util"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/util/spring-util-3.0.xsd"

util:constant id="aaa"

static-field="com.shop.Product.AAA" />

util:constant id="cdrw"

static-field="com.shop.Product.CDRW" />

</beans>

a)Yes

b) No

**ans**-- b

**5**. The Spring Expression Language can be accessed by:-

a) XML configuration

b) Annotations

c) None of the mentioned

d) All of the mentioned

**Ans**-- d

**6**. Ways to declare bean from a static field?

a) FieldRetrievingFactoryBean

b) util:contant

c) All of the mentioned

d) None of the mentioned

**ans**-- c

**7**. What will be the output of the code?

public​ class​ CreatePro {

String ProductId;

public​ CreatePro(String ProductId) {

this​.ProductId = ProductId;

}

public​ static​ Product creation\_Product(String productId) {

System.out.println("Bean Created");

if​ ("aaa".equals(productId)) {

return​ new​ Battery("AAA", 2.5);

} else​ if​ ("cdrw".equals(productId)) {

return​ new​ Disc("CD-RW", 1.5);

}

}

}

<beans ...>

<bean id="aaa" class​="CreatePro"

factory-method="createProduct">

<constructor-arg value="aaa" />

</bean>

<bean id="cdrw" class​="CreatePro"

factory-method="createProduct">

<constructor-arg value="cdrw" />

</bean>

</beans>

a) BeanCreationException

b) Bean Created

c) ClassPathException

d) None of the mentioned

**Ans**-- a

**8**. For collection merging, which Principles we have to follow:

1. The beans must and should follow bean inheritance.

2. The bean collection implementation classes must be the same.

3. The collection implementation class's generic type must be the

same.

a) 1 And 2

b) Only 2

c) All three

d) None of the above.

**Ans**-- c

9. Which of the following is not feature of Beans?

a) Introspection

b) Events

c) Persistence

d) Serialization

**Ans**-- d

10. . Dynamic-language-backed bean with some configurations

a) Refreshable bean

b) Beanshell

c) Scripting Beans

d) Inline Script

**Ans**—a

11. Which factory method is used in singleton designed pattern?

A Static factory method

B Non Static factory method

**Ans**> A

12>In which method we can use factory-method?

A Static method

B Main method

**Ans**> A

13>Choose the correct syntax for a non-static factory method?

A <bean class="com.factotymethod.A" factory-method="getB"></bean>

B <bean id="b" class="com.factotymethod.A" factory-method="getB"

></bean>

C <bean id="b" class="com.factotymethod.A" factory-method="getB"

factory-bean="a"></bean>

D <bean name="bean1" class="com.factotymethod.A"

factory-method="getB" factory-bean="a"></bean>

**Ans**> C

14> Which interface specifies afterPropertiesSet() method ?

A> org.springframework.beansfactory.InitializingBean

B> org.springframework.applicationcontext.InitializingBean

C> org.springframework.beans.factory.DisposableBean

D> org.springframework.beans.factory.InitializingBean

**Ans**> D

15>Choose the correct syntax for init-method in xml file?

A><bean name = "exampleBean" class = "examples.ExampleBean"

init-method = "init"/>

B><bean id = "exampleBean" name = "exampleBean" init-method =

"init"/>

C><bean id = "exampleBean" class = "examples.ExampleBean"

init-method = "init"/>

D><bean id = "exampleBean" class = "examples.ExampleBean" method =

"init"/>

**Ans**> C

16>Which method will shutdown and call the relevant destroy methods?

A> destroy()

B> registerShutdownHook()

C> shutdow()

D> registerDestroy()

**Ans**> B

17>Which interface defines callBackMethods?

A> BeanPostProcessor

B> BeanProcessor

C> BeanFactory

D> ApplicationContext

**Ans**> A

18>Choose the correct syntax for destroy-method in xml file?

A><bean name = "exampleBean" class = "examples.ExampleBean"

destroy-method = "destroy"/>

B><bean id = "exampleBean" name = "exampleBean" destroy-method =

"destroy"/>

C><bean id = "exampleBean" class = "examples.ExampleBean"

destroy-method = "destroy"/>

D><bean id = "exampleBean" class = "examples.ExampleBean" method =

"destroy"/>

**Ans**> C

19> Where we can define default init-method?

A>xml file

B>controller class

C>service class

D>componenet class

**Ans**> A

20>What are the annotations used in the replacement of init and destroy

methods?

A>@Construct and @Destroy

B>@init and @destroy

C>@PostConstruct and @PreDestroy

D>@PostInit and @PreDestroy

**Ans**>C

Q 21. What types of Dependency injection does spring supports?

a) Constructor based, Setter based

b) Constructor based, Setter based, Getter Based

c) Setter based, Getter based, Properties based

d) Constructor based, Setter based, Properties based

**Ans**:A

Q 22. How can you inject Java Collection in Spring?

a) Using list, set, map or props tag.

b) Using lit, set, map or collection tag.

c) Using list, set, props or collection tag.

d) Using list, collection, map or props tag.

**Ans**:A

Q 23. Declaring a bean from a static field requires a built-in factory bean

FieldRetrievingFactoryBean and fully qualified field name or instance field

is specified in the list property.

a) True

b) False

**Ans**:B

Q 24. As an alternative to specifying the field name in the staticField

property explicitly, you can set it as the bean name of

FieldRetrievingFactoryBean.

a) True

b) False

**Ans**:A

Q25. Declaring bean form object properties can be done using:-

a) PropertyPathFactoryBean

b) util:constant

c) None of the mentioned

d) All of the mentioned

**Ans**:A

Q26. The Spring Expression Language can be accessed by:-

a) XML configuration

b) Annotations

c) None of the mentioned

d) All of the mentioned

**Ans**:D

Q27. Which annotation is used as a substitute of initialization method?

a) @PostConstruct

b) @PreDestroy

c) None of the mentioned

d) All of the mentioned

**Ans**:A

Q28.Which annotation is used as a substitute of destroy method?

a) @PostConstruct

b) @PreDestroy

c) None of the mentioned

d) All of the mentioned

**Ans**:B

Q29. Which configuration can be used for Dependency Injection?

a) XML Configuration

b) Annotation Configuration

c) Java Based Configuration

d) All of the mentioned

**ANS**:D

30. Beans can be created by which of the following properties?

a) Static factory-method

b) Instance Factory-Method

c) All of the mentioned

d) None of the mentioned

**Ans**:C

31)If two classes depend on minute details of each other, a sudden

change in one class will initiate subsequent change in the dependent

class, the classes are said to be :

a)strongly dependent

b)loosely dependent

c)tightly coupled

d)loosely coupled

**answer**:c

32)Controller in spring is a

a)abstract class

b)concrete class

c)final class

d)an interface

**answer**:d

33)Lifecycle of a bean is managed by

a)BeanFactory

b)BeanSystemFactory

c)Spring Container

d)FileSystemResource

**answer**:c

34)POJO is an ordinary Java object, not bound by any special restriction

and not requiring any class path.

a)True

b)False

**answer**:a

35)Components of a Spring Application

a)Interface,Bean class,AOP,Bean Configuration File,User program

b)Context,AOP,Bean Configuration File,Interface,User program

c)Interface,Bean class,JMS,AOP,Bean Configuration Fild)Bean class,AOP,Core,Interface,User program

**answer**:a

36)What is the measure of interdependence with in the classes?

a)Cohesion

b)Encapsulation

c)Coupling

d)Interdependency injection

**answer**:c

37)In Spring framework, Spring Container uses which mechanism to inject

objects required for main method

a)Setter

b)Dependency injection

c)Constructor

d)Setter and getter

**answer**:b

38)Spring promotes loose coupling through

a)ORM

b)AOP

c)IOC

d)SPEL

**answer**:c

39)Expression language is used in springs are not

a)Yes

b)No

**answer**:a

40)Which of these configuration is used for key method?

a)new ClassPathXmlApplicationContext("com/learn/config.xml");

b)new FileSystemXmlApplicationContext("C:/learn/config.xml");

c)new FileSystemXmlApplicationContext("./learn/config.xml");

d)none

**answer**:a

41.What is singleton scope?

A. This scopes the bean definition to a single instance per Spring IoC

container.

B. This scopes the bean definition to a single instance per HTTP Request.

C. This scopes the bean definition to a single instance per HTTP Session.

D. This scopes the bean definition to a single instance per HTTP

Application/ Global session.

**Ans**:​This scopes the bean definition to a single instance per Spring IoC

container.

42.Which one is the default scope of the beans?

A. Prototype

B. Session

C. Request

D. Singleton

**Ans**​:Singleton

43. Which scope creates a new bean instance each time when requested?

A. Singleton

B. Prototype

C. Session

D. Request

**Ans**:​Prototype

44.Session Creates a single bean instance per HTTP request, only valid in

the context of a web application?

A. True

B. False

**Ans**:​False

45.Which annotation is used as a substitute of initialization method?

A. @PostConstruct

B. @PreDestroy

C. None of the mentioned

D. All of the mentioned

**Ans**:​@PostConstruct

46.Which of the following are considered valid beans?

A. Singleton

B. Prototype

C. All of the mentioned

D. None of the mentioned

**Ans**:​All of the mentioned

47.Which attribute is used to set the scope of the bean?

A. setScope

B. scope

C. getScope

D. none of the mentioned

**Ans**:​scope

48.What is byType mode of autowiring?

A. Default setting which meas no autowiring and you should use explicit

bean reference for wiring.

B. Autowiring by property name. Spring tries to match and wire its

properties with the beans defined by the same names in the configuration

file.

C. Spring first tries to wire using autowire by constructor, if it does not

work, Spring tries to autowire by byType.

D. Autowiring by property type. Spring tries to match and wire a property if

its type matches with exactly one of the beans name in configuration file.

**Ans**​:Autowiring by property type. Spring tries to match and wire a property

if its type matches with exactly one of the beans name in configuration

file.

49.How to auto-inject into field a bean by its name? select one or more

response.

A. With the name attribute of the @Autowired annotations.

B. By using the single @Qualifier annotation.

C. By using both @Autowired and @Qualifier annotation.

**Ans**:​By using both @Autowired and @Qualifier annotation.

50.What are different Autowire types?

A. byName,byType,destructor and autodetect.

B. byName,byMethod,constructor and autodetect.

C. byName,btType,constructor and autocorrect.

D. byName,byType,constructor and autodetect.

**Ans**:​byName,byType,constructor and autodetect.

Ques51: Is Partial dependency possible by Constructor method?

a.True

b. False

**Ans**: b

Ques 52: Which of the following element is not used inside the

constructor-arg element?

a.Set

b.Map

c.List

d.Vector

**Ans**: d

Ques 53: Which is the most flexible injection method?

a.Setter Injection

b.Constructor Injection

c.Both a & b.

**Ans**: a

Ques 54: IOC or Dependent Injection is a...

a. Design Pattern

b. Frame work

c. Annotation

**Ans**. A

Ques 55: What are different types of Bean Injections?

a. setter and getter

b. setter , getter and constructor

c. setter and constructor

d.None

**Ans**: c

Ques 56:How to auto-inject into a field a bean by its name? Select one or

more responses

a. By using both the @Autowired and @Qualifier annotation

b. with the name of the @Autowired annotation

c. by using the single @Qualifier annotation

**Ans**. A

Ques 57. What is the other name of Dependency Injection

a.Inversion of control

b. Polymorphism

c.AOP

d.Container

**Ans**.a

Ques58:What statement is not correct in live environment? select a unique

answer

a. Constructor and properties autowiring in the same bean are not

compatible

b. A bean should have a default or no-arg constrctor

c. The <constructor-arg> should take type, name and index o reduce

ambiguity.

d. All of the above

**Ans**: a,b.

Ques59: Whenever the dependency-check fails, Spring throws which

exception

a.UnsatisfiedDependencyException

b. DependencyException

c.DataAccess exception

**Ans**. A

Ques60: Default mode of dependency-check

a.Simple

b.Object

c. all

d.none

**Ans**: d

61.Which stereotype annotation marks java class as a bean?

a.@Controller

b.@Component

c.@Repository

d.@Configuration

**Ans**:b

62.Which annotation is used as a Generic StereoType?

a.@Component

b.@Controller

c.@Service

d.@Repository

**Ans**:a

63. Autowiring is implicitly enabled in context:component-scan.

a.True

b.False

**Ans**:a

64.Match the following annotations with respective layers they are used.

1.@Repository a.Presentation Layer

2.@Service b.Persistence Layer

3.@Controller c.Service Layer

A.1-a,2-b,3-c

B.1-c,2-a,3-b

C.1-b,2-c,3-a

**Ans**:c

65.What should be enabled in spring configuration file if we want to

implement scanning of stereotype annotations?

a.<context:component-scan base-package="">

b.<context:component base-package="">

c.<context:component-scan base="">

d.<component-scan base-package="">

**Ans**:a

66.The mechanism through initialization and destruction callback methods

can be invoked are:

a.By implementing initializing Bean and Disposable Bean interfaces.

b.By giving init-method and destroy-method attribute values in

configuration file.

c.both

d.none

**Ans**:c

67.The initializing Bean interface is present in package.

a.org.springframework.beans.factory.InitializingBean.

b.org.springframework.beans.beanfactory.InitializingBean.

c.org.springframework.beans.InitializingBean.

d.org.springframework.web.factory.InitializingBean.

**Ans**:a

68.The initialization method provided by Initializing Bean interface is:

a.afterPropertiesSet()

b.afterSet()

c.init()

d.init-method()

**Ans**:a

69.@PostConstruct and @PreDestroy annotations are implemented by using

this in configuration file.

a.<context:annotation-config/>

b.<context:component-scan base-package=" "/>

c.<context:annotation/>

d.<context:configuration/>

**Ans**:a

70.Stereotype annotations cannot be implemented on:

a.Concrete classes.

b.Interfaces.

c.Abstract classes.

**Ans**:b

71. Method used to process bean before initialization callback

a) scope

b) postProcessAfterInitialization()

c) postProcessBeforeInitialization()

d) it’s own constructor

**Answer**:C

Explanation: You can process every bean before initialization callback

method by implementing the postProcessBeforeInitialization() and

methods.

72. Method used to process bean after initialization callback

a) scope

b) getBean

c) postProcessAfterInitialization()

d) it’s own constructor

**Answer**: c

Explanation: You can process every bean after initialization callback

method by implementing the postProcessAfterInitialization() and

methods

73. It’s possible to replace the original bean instance with a

brand-new instance in your bean post processor

a) True

b) False

**Answer**: a

Explanation: Both the postProcessBeforeInitialization() and

postProcessAfterInitialization() methods must return an instance for

the bean being processed.

74. In application context, BeanPost Processors are registered using

addBeanPostProcessors method

a) True

b) False

**Answer**: b

Explanation: Using an application context, the registration will be as

simple as declaring an instance of the processor in the bean

configuration file.

75. 1. Which version of Java introduced annotation?

a) Java 5

b) Java 6

c) Java 7

d) Java 8

**Answer**: a

Explanation: Annotation were introduced with Java 5 version.

76. Java-based configuration lets you configure Spring without \_\_\_.

A) Annotations

B) Xml

C) Both A and B.

D) None of the above

**ANS**:B

77. Choose the correct syntax to access base package location to

scan classes.

A) <context:component-scan base package/>

B) <context:annotation-config>

C) <tx:annotation driven>

D) <mvc:annotation-driven>

**Ans**:A

78. Which class acts as IoC Container?

A) ServletContext

B) DispatcherServlet

C) ApplicationContext

D) None of the above.

**Answer**:C

Exp: ApplicationContext class acts as IoC Container.

Hide Answer

79. A bean can have more than one name using multiple id

attributes?

a) True

b) False

**Answer**: a

Explanation: Beans are allowed to have more than one ids.

80. Bean’s naming convention:-

starts with lowercase, camelcase from then on.?

a) True

b) False

**Answer**: a

Explanation: Beans follow naming conventions.