

```
public String getValue (String name)
```

```
{
```

```
    if (name.length() == 1)
```

```
    {
```

```
        s.o.p (name.charAt(length-1))
```

```
    }
```

```
    else
```

```
        return getValue (substring (0, name.length()-1)) +
```

```
        name.charAt (name.length()-1)
```

```
main
```

```
{
```

```
    getValue ("India");
```

a) India

b) I

c) NDia

d) aIdnI

return (Indi) + a;

(after (Indi) + i) + a

return (India) + a.

India

1) What is join point cut?

Ans: An Expression to identify joinpoints.

2) What is the scope of stateful bean in spring?

Ans: - prototype scope.

3) Where will prototype scope is used

stateful

stateless

Restful

Restless

4) What is Advice?

5) Dependency injection & IOC in hollywood principal?

ANS: - Don't call us, We'll call you

6) What are the ways to give configuration file.

XML Based

Annotation Based

Java Based

Annotation Config Application Config.

A) 1, 2, 3

B) 1, 3, 4

C) 2, 3, 4

7) How to inherit of bean?

Ans:- parent = "

① dependency • injection and ioc hollywood principle.

↳ don't call us, we'll call you.

② mvc → expansion.

↳ model view controller.

③ inner bean ~~config~~ scope

↳ prototype ↖
singleton.

<bean id = " " class = " " scope = "singleton",
prototype
required =

HelloWorld d = (HelloWorld) context.getBean
("helloworld")

HelloWorld s = (HelloWorld) context.getBean
("helloworld")

option:

<bean id = "helloworld" class = " " scope = "singleton"

① Which scope is valid in web aware application.

① req

② session

③ singleton

④ prototype

② What should be the ~~type~~^{scope} of jinner bean?

① singleton

② prototype

③ req

④ session

③ How will inject dep in constructor, which autowiring?
by constructor, by setter

④ Autodiscovery.

use of autodiscovery

→ reducing bean tag.

→ eliminating property.

→ eliminating constructor.

⑤ Hollywood principle.

DI

IOC

Don't call us we will call you.

Always customer is right.

⑥

⑦ When will you go for Java based Config → Slide.
— when settings doesn't change.
— ~~what~~ To inject dependencies during runtime.

or

⑧ Which annotation are there in JSR 250.

⑨ Which annotation for DTO classes.

① within methods. → @require require-new

② How do you ~~can~~ denote method annotation.

@Transaction

@Inject

@Autowired

@Resource.

③ Is There a way of programmatic transaction management

① yes, using @transactional. bean for processing.

② yes, using transaction template?

④ Use of transaction management

① logging → history.

② modularize the transaction using oop.

⑤ annotation for enabling transaction

ctx: annotation-driven >

ctx: annotation=true >

⑥ Challenges of transaction TMs

① error prone

② ← page 32 of slide.

⑦ How do you implement ~~the~~ Manual transaction management using Transaction Manager Interface.

⑧ Q 34 of spring cert dump

@Rule

TestName : testName

@Test
testA()

test.getName().equals("testA")

@Test
testB()

test.getName() == ()

}

$$C\% = 5.2 + "3.26" \cdot 1.8$$

String s1 = "india";

String s2 = "india";

s.o.p (s1 + " equals " + s2 + " equals " + " → " +
s1.equals(s2)) ;

⇒ india equals india → 1

⇒ india equals india → true

⇒ india equals india → 0

⇒ india equals india → false.

int a;

static int a;

int foo() { return a; }

static int foo() { return a; }

1 & 3

1 & 4

2 & 3

2 & 4

which is ~~not valid~~
two options cannot
be together

Spring

- 1.) The scope of inner bean? Singleton, Prototype, Request, Session
- 2.) Which one is used to eliminate/reduce the use of
<property> & <constructor-arg>?
Auto-discovery, Auto-detect, ...
- 3.) The default mode of autowire?
ByName,
ByType,
On, off
- 4.)

1) what is a Standard Interface?

- a) Serializable
- b) Collection
- c) System
- d) - - -

2) which one is not addressed in SPRING ^{Data Access}?

- a) Thread Safe
- b) Security and Authentication
- c) JDBC Template
- d) Object into DB

3) what can be the attributes in Inner bean?

- a) ID, class
- b) interface, inner
- c) class, inner
- d) inherit, parent

4) To autowire in the constructor what autowire type will be used.

a) by constructor, ~~by~~ @ Autowire ...
:
:

5) Name and Value which two are ~~are~~ any objects

a) Set

b) List

c) Map ✓

d) ~~Collection~~ Props

6) How you do ~~the~~ package enable the annotations

a) <Context: component scan>

b) <context: Annotation config>

c) ..

d) ..

7) which 3 ways to implement the Spel?

a) Annotation Config Application context

b) Java based,

c) XML based,

d) annotation based.

5) @ Controller is used in ?

- a) Data Repository.
- b) Spring MVC
- c) Component
- d) Service

6) How to declare a Java-based Configuration in Configuration meta-data ?

⇒ ANS: @ Configuration, above the class.

1) Which is a standard Java Interface?

- a) Serializable.
- b) System
- c) Collections.

2) Which is the proper inner bean declaration.

~~```
<bean id="call" class="com.Employee">
 <property name="id"/>
 <inner bean id="" class="">
 </inner bean>
</bean>
```~~

b) 

```
<bean id="emp" class="com.Employee">
```

Ans → 

```
 <property name="empid" value="001">
 <inner bean id="addr" class="com.Address">
 <property name="city" value="Colombato"/>
 </inner bean>
 </property>
</bean>
```

3) What happens when web-aware Spring Application Context is used.

- a) It returns nothing.
- b) It works properly.
- c) You cannot use it.

4) What is an advice?

- An action to be performed at the joinpoint.

8) when you will use the Java based config.

a) when Settings doesn't change

b) To inject dependencies in Runtime

c) To inject appropriate dependency according to algorithm

d)

9) when will the Spring will inject the dependencies

a) Runtime

b) Compile time.

10) what ~~you will~~ is the use of transaction management in Spring?

:

11) @Autowired

Object A;

```
if (A == null)
```

```
{
```

```
 sop("Hello");
```

```
}
```

```
else
```

```
{
```

```
 sop("world");
```

```
}
```

} some method.

12) what is the purpose of using Spring Inheritance?

a) Share common properties.

b) To use java class inheritance.

c) To use java inheritance in Spring.

- ① Inner bean scope - options  
Singleton, prototype, session, request
- ② How will you provide the configuration metadata to the container
- ③ use of `ClasspathXmlApplicationContext` and `FilepathXmlApplicationContext`
- ④ How we declare a transaction methods -  
`@Transactional`
- ⑤ Aop style - `@Aspect`
- ⑥ use of Autodiscovery - options  
Reduce the constructor args  
" " Property  
" " bean creation
- ⑦ Abbreviation of MVC → model-view-controller
- ⑧ How to mention the Java based configuration file → `@Configuration`
- ⑨ How to start the container without using xml - `@AnnotationConfigApplicationContext`

⑩ Annotation which is used for constructor-arg  
and property injection - options

autowire



possible valid attributes in an inner bean

id, class ✓  
interface, inner  
class, inner  
inherit, parent.

---

How to inherit a bean def.

- ① parent
- ② inter

---

purpose of using ~~spring~~ object inheritance in spring  
• sharing common props.

---

default mode of autowiring.

- byName
- byType
- on
- off
- ~~byType~~

---

How do you set autowire

autowire = true  
= "11"

= on

by default

---

Two major packages used in spring framework

org. sf. beans &

... application

beans & , context ✓

bean ... . properties.

property — . application



1) What is the keyword for inheritance in spring

a) inherit.

b) parent.

c) include.

Benefits of DI

- loose coupling

- Easy testing.

- Tight coupling

---

Same bean-name. declared twice -

→ scope =

---

pointed expression is for

① one or more advices

② one or more aspects

③ one or more joinpoint

④ one or more concerns

---

In web aware what kind of exception can occur

① `IllegalStateException`

② `NullPointerException`

③



① @Controller is used in which framework

MVC,

AOP

Service, Controller.

C

② How will you connect core module and Data access module (ORM)  
JMS, JPA, ORM, transaction

③ <sup>Centre.</sup> Heart of MVC - Spring  
dispatcher Servlet,

④ Config in XML to enable ~~the~~ Spring MVC.

<mvc:annotation-driven>

</mvc>

⑤ To scan the packages, for creating beans.  
<context:component-scan>

To enable annotation in Spring

context:annotation-config

~~Class Diff~~ behv

Use of ClassPath XML AC & File System AC

① classpath & absolute path of file

② " & relative path of file

③ Now.

④ Absolute path & relative path.

Q 3 ways to implement Spring Configuration.

① XML Based.

② Java Based.

③ Annotation Based.

④ Annotation Config AC.

1, 2 & 3

Q Instead of using XML, how do you start ~~AC~~ <sup>Container</sup>?

① Annotation Config AC.

---

Scopes of beans.

{ singleton, proto.  
- - - all pre.

---

which two scopes are valid for both web & standalone.

Singleton & prototype.

---

which scope are valid for only web aware applications.

Session, prototype, request, session, global

---

~~Look~~ dependency lookup - ?

---

which attribute cannot be inherited →

---

scope of inner bean.

↳ Session, prototype, etc.



1. In Spring, which concept is used to create authentication and logging?

Ans: AOP

2. Which Annotation is used for MVC?

Ans: @Controller, @RequestMapping

3) Which Package is useful for creating Spring beans?

4). Which one is useful for JDBC Template?

options:-

a) ThreadSafe

b) DataSource

c) Security and Authentication

d) Data Access Object

5) @Autowired

private MyBean mybeans;

if (mybeans == ~~true~~<sup>null</sup>)

{ s.o.p ("true");

}

else {

{ s.o.p ("false");

}

Ans:-

6)

// Not using Spring

private MyBean mybeans;

if (mybeans.equals(mybeans))

{ s.o.p ("true");

}

else { s.o.p ("false");

}



9) Autowired, Autodiscovery which by using this which one is reduced

- a) bean element      b) XML configuration.

10) Creation of Bean.

a) Context creat, Retrive bean from context, use Bean

b) Retrive and use bean.

d

11) Java Based Configuration which annotation is used.

a) @configuration.

b) @component .

12) `<bean id="user", class="Baseclass"`

`<property name="country, value="US" >`

`</bean>`

`<bean id="user1" class parent="Baseclass"`

`<property name="country`

`value="Japan">`

`<property name="type`

`value=1`

`</bean>`

a) Country Japan, type = 1

c) country US

b) country US, type = 1

1. Which standard interface used in Spring?  
a) Serializable b) Connection c) System
2. Which scope is used in Inner Bean?  
a) request b) singleton c) prototype d) session
3. What are the scopes in Bean?  
a) All the above b) singleton c) prototype d) request  
e) session f) Global session.
4. What is the advantage of spring  
Ans - loose coupling & code can be maintained easily.
5. In POJO Java model, which class extends in spring?
6. How to autoinject into a field a bean by its name?  
a) With name attribute of @Autowired annotat  
b) @Qualifier  
c) Both.
7. Which collections is used in Spring  
Ans: Map list set props
8. @Controller is used in which framework?  
a) MVC ✓ b) AOP c)

10. Always return single value  $\rightarrow$  Singleton.

11. Annotation used in Java-class  $\rightarrow$  @Configuration.

12. ~~How~~ How to enable configuration in Java class.

$\rightarrow$  <context: annotation-config>

$\rightarrow$  <context: component-scan>

13. Annotation used in DAO  $\rightarrow$  @Repository.

14. How to inject database Properties in JDBC template in web.xml?

15. Hollywood Principle.

16. Annotation in JSP-2.0.



## Spring:-

1. Centralized process in MVC : Dispatcher Servlet.

2. Whether there is method of } : Yes.  
Programmatic Transaction

3. Standard Interface:

4. @Transactional (Required)  
void add  
{  
  
}  
  
@Transactional (Required = new)  
void sub { ~~Required~~ ~~new~~  
{  
  
void add ()  
  
}

5. How to autowire by constructor?

6. ~~How~~ which Scope can be used in web-Aware?

7. How to activate autowire? autowire=" ", autowire="on",  
set autowire="true", default.

8. Which Scope can be used to get the answer?

a = " ";  
b = "null";

9. Named value pair of collection : <Map>



1) Annotations that follows MVC pattern.

- 1) @Autowired, @Qualifier.
- 2) @Controller, @RequestParam, @RequestMapping, ...
- 3) @Resource, @Inject

2) How will you inject dependencies in the Constructor, in XML file.

- \* by Name
- \* by Type.
- \* by Constructor.
- \* by autodetect.

3) How will you make database connection using Jdbc template.

- \* using Connection object.
- \* using DataSource bean.
- \* using Configuration XML file.

4) Configuration to be used in xmc to enable MVC pattern.

1) <MVC - Annotation - driven>

5) Annotation used for DAO classes.

@ Repository -

6)