All Tests (https://rank.sanfoundry.com/spring-tests/)

Top Rankers (https://rank.sanfoundry.com/top-rankers-spring/)

Practice Test 1 (https://rank.sanfoundry.com/spring-practice-test-1/)

Practice Test 2 (https://rank.sanfoundry.com/spring-practice-test-2/)

Practice Test 3 (https://rank.sanfoundry.com/spring-practice-test-3/)

Practice Test 4 (https://rank.sanfoundry.com/spring-practice-test-4/)

Practice Test 5 (https://rank.sanfoundry.com/spring-practice-test-5/)

Practice Test 6 (https://rank.sanfoundry.com/spring-practice-test-6/)

Practice Test 7 (https://rank.sanfoundry.com/spring-practice-test-7/)

Practice Test 8 (https://rank.sanfoundry.com/spring-practice-test-8/)

Practice Test 9 (https://rank.sanfoundry.com/spring-practice-test-9/)

Practice Test 10 (https://rank.sanfoundry.com/spring-practice-test-10/)

Mock Test 1 (https://rank.sanfoundry.com/spring-mock-test-1/)

Mock Test 2 (https://rank.sanfoundry.com/spring-mock-test-2/)

Mock Test 3 (https://rank.sanfoundry.com/spring-mock-test-3/)

Mock Test 4 (https://rank.sanfoundry.com/spring-mock-test-4/)

Mock Test 5 (https://rank.sanfoundry.com/spring-mock-test-5/)

Mock Test 6 (https://rank.sanfoundry.com/spring-mock-test-6/)

Mock Test 7 (https://rank.sanfoundry.com/spring-mock-test-7/)

Mock Test 8 (https://rank.sanfoundry.com/spring-mock-test-8/)

Mock Test 9 (https://rank.sanfoundry.com/spring-mock-test-9/)

Mock Test 10 (https://rank.sanfoundry.com/spring-mock-test-10/)

« Prev Page (https://www.sanfoundry.com/spring-questions-answers-declaring-beans/)

Next Page (https://www.sanfoundry.com/spring-questions-answers-bean-post-processors/) »

# Spring Questions and Answers – Bean Scopes

This set of Java Spring Multiple Choice Questions & Answers (MCQs) focuses on “Bean Scopes”.

1. A bean can be requested by:-
a) getBean method
b) reference from another bean using autowiring, property etc
c) all of the mentioned
d) none of the mentioned
View Answer

Answer: c
Explanation: When a bean is requested by the getBean() method or a reference from other beans, Spring will decide which bean instance should be returned according to the bean scope.

2. Which attribute is used to set the scope of the bean?
a) setScope
b) scope

c) getScope

d) none of the mentioned

View Answer

Answer: b

Explanation: Scope attribute defines the scope of a bean.

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

a) Prototype

b) Session

c) Request

d) Singleton

View Answer

Answer: d

Explanation: This unique bean instance will be returned for all subsequent getBean() calls and bean references.

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

a) Singleton

b) Prototype

c) Session

d) Request

View Answer

Answer: b

Explanation: Creates a new bean instance each time when requested.

5. Session Creates a single bean instance per HTTP request, only valid in the context of a web application?

a) True

b) False

View Answer

Answer: b

Explanation: Session Creates a single bean instance per HTTP session, only valid in the context of a web application.

6. Which of the following are considered valid beans?

a) Singleton

b) Prototype

c) All of the mentioned

d) None of the mentioned

View Answer

Answer: c

Explanation: In Spring 1.x, singleton and prototype are the only two valid bean scopes, and they are specified by the singleton attribute (i.e., singleton=”true” or singleton=”false”), not the scope attribute.

7. What will be the output?

```
public class ShoppingCart
{
    private List<Product> items = new ArrayList<Product>();
    public void addItem(Product item)
    {
       items.add(item);
    }
    public List<Product> getItems()
    {
       return items;
    }
}
```

```
   <beans ...>
        <bean id="aaa" class="com.shop.Battery">
        <property name="name" value="AAA" />
        <property name="price" value="2.5" />
        </bean>
        <bean id="cdrw" class="com.shop.Disc">
        <property name="name" value="CD-RW" />
        <property name="price" value="1.5" />
        </bean>
        <bean id="dvdrw" class="com.shop.Disc">
        <property name="name" value="DVD-RW" />
        <property name="price" value="3.0" />
        </bean>
        <bean id="shoppingCart" class="com.shop.ShoppingCart" />
   </beans>
```

```
   import org.springframework.context.ApplicationContext;
   import org.springframework.context.support.ClassPathXmlApplicationContext;
   public class Main
   {
        public static void main(String[] args) {
        ApplicationContext context =
        new ClassPathXmlApplicationContext("beans.xml");
        Product aaa = (Product) context.getBean("aaa");
        Product cdrw = (Product) context.getBean("cdrw");
        Product dvdrw = (Product) context.getBean("dvdrw");
        ShoppingCart cart1 = (ShoppingCart) context.getBean("shoppingCart");
        cart1.addItem(aaa);
        cart1.addItem(cdrw);
        System.out.println("Shopping cart 1 contains " + cart1.getItems());
        ShoppingCart cart2 = (ShoppingCart) context.getBean("shoppingCart");
        cart2.addItem(dvdrw);
        System.out.println("Shopping cart 2 contains " + cart2.getItems());
        }
   }
```

a) Shopping cart 1 contains (AAA 2.5, CD-RW 1.5)
Shopping cart 2 contains (AAA 2.5, CD-RW 1.5, DVD-RW 3.0)

b) Shopping cart 1 contains (AAA 2.5, CD-RW 1.5)

Shopping cart 2 contains (DVD-RW 3.0)

c) BeanCreationException

d) None of the mentioned

View Answer

Answer: a

Explanation: Since the default scope is singleton, so items added by first bean instantiated will be used again by same bean if instantiated again.

8. In above question if scope of shoppingCart named bean is prototype, then what will be the output?
What will be the output?

a) Shopping cart 1 contains (AAA 2.5, CD-RW 1.5)
Shopping cart 2 contains (AAA 2.5, CD-RW 1.5, DVD-RW 3.0)

b) Shopping cart 1 contains (AAA 2.5, CD-RW 1.5)
Shopping cart 2 contains (DVD-RW 3.0)

c) BeanCreationException

d) None of the mentioned

View Answer

Answer: b

Explanation: Since the default scope is prototype, so items added by first bean instantiated will be removed by same bean if instantiated again, and will add new items listed.

9. Which interface is used to perform initialization of beans?

a) InitializingBean

b) Disposablebean

c) None of the mentioned

d) All of the mentioned

View Answer

Answer: a

Explanation: Spring allows your bean to perform initialization callback methods afterPropertiesSet() by implementing the InitializingBean and interfaces.

10. Which interface is used to perform destruction of beans?

a) InitializingBean

b) Disposablebean

c) None of the mentioned

d) All of the mentioned

View Answer

Answer: b

Explanation: Spring allows your bean to perform destroy callback methods destroy() by implementing the DisposableBean and interfaces.

11. Alternate way of initialization method is:-

a) init-method attribute

b) afterPropertiesSet

c) destroy-method attribute

d) none of the mentioned

View Answer

Answer: a

Explanation: A better approach of specifying the initialization callback methods is by setting the init-method attributes in your bean declaration.

12. Alternate way of destruction method is:-

a) init-method attribute

b) afterPropertiesSet

c) destroy-method attribute

d) none of the mentioned

View Answer

Answer: c

Explanation: A better approach of specifying the destroy callback methods is by setting the destroy-method attributes in your bean declaration.

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

a) @PostConstruct

b) @PreDestroy

c) None of the mentioned

d) All of the mentioned

View Answer

Answer: a

Explanation: Using JSR annotation.

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

a) @PostConstruct

b) @PreDestroy

c) None of the mentioned

d) All of the mentioned

View Answer

Answer: b
Explanation: Using JSR annotation.

15. Which configuration can be used for Dependency Injection?
a) XML Configuration
b) Annotation Configuration
c) Java Based Configuration
d) All of the mentioned

View Answer

Answer: d
Explanation: All of the above mentioned can be used for Dependency Injection.

**Sanfoundry Global Education & Learning Series – Java Spring.**

To practice all areas of Java Spring, here is complete set of 1000+ Multiple Choice Questions and Answers (https://www.sanfoundry.com/1000-spring-questions-answers/).

« Prev Page - Spring Questions and Answers – Declaring Beans (https://www.sanfoundry.com/spring-questions-answers-declaring-beans/)
» Next Page - Spring Questions and Answers – Bean Post Processors (https://www.sanfoundry.com/spring-questions-answers-bean-post-processors/)

## Deep Dive @ Sanfoundry:

1. **Java Programming Examples on Java.Lang (https://www.sanfoundry.com/java-programming-examples-java-lang/)**
2. **Java Questions and Answers (https://www.sanfoundry.com/java-questions-answers-freshers-experienced/)**
3. **C# Programming Examples on Threads (https://www.sanfoundry.com/csharp-programming-examples-on-threads/)**
4. **Spring Questions and Answers (https://www.sanfoundry.com/1000-spring-questions-answers/)**
5. **Spring Questions and Answers – AspectJ Aspects (https://www.sanfoundry.com/spring-interview-questions-answers/)**
6. **Spring Questions and Answers – Spring BlazeDS support (https://www.sanfoundry.com/spring-interview-questions-answers-experienced/)**
7. **Advanced Java Questions & Answers – Java Beans (https://www.sanfoundry.com/advanced-java-questions-answers-java-beans/)**
8. **Spring Questions and Answers – Accessing and Integrating Spring with known Frameworks (https://www.sanfoundry.com/spring-questions-answers-accessing-integrating-spring-questions-answers-known-frameworks/)**
9. **Spring Questions and Answers – Refreshing and Defining Beans from Scripts (https://www.sanfoundry.com/spring-questions-answers-refreshing-defining-beans-scripts/)**
10. **Spring Questions and Answers – Implementing and Injecting Beans with Scripting Languages (https://www.sanfoundry.com/spring-questions-answers-implementing-injecting-beans-scripting-languages/)**