

Spring core Part 3

1. What is the scope of inner beans.

- a. Singleton
- b. Prototype**
- c. Request
- d. None Of the above

2. Inner Beans can be retrieved by it's name.

- a. True
- b. False**

```
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

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

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

a) XML configuration

b) Annotations

c) None of the mentioned

d) All of the mentioned

6. Ways to declare bean from a static field?

a) FieldRetrievingFactoryBean

b) util:contant

c) All of the mentioned

d) None of the mentioned

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

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.

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

a) Introspection

b) Events

c) Persistence

d) Serialization

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

a) Refreshable bean

b) Beanshell

c) Scripting Beans

d) Inline Script

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

A Static factory method

B Non Static factory method

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

A Static method

B Main method

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>

14. Which interface specifies afterPropertiesSet() method ?

A> org.springframework.beans.factory.InitializingBean

B> org.springframework.applicationcontext.InitializingBean

C> org.springframework.beans.factory.DisposableBean

D> org.springframework.beans.factory.InitializingBean

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"/>

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

A> destroy()

B> registerShutdownHook()

C> shutdown()

D> registerDestroy()

17. Which interface defines callBackMethods?

A> BeanPostProcessor

B> BeanProcessor

C> BeanFactory

D> ApplicationContext

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"/>

19. Where we can define default init-method?

A>xml file

B>controller class

C>service class

D>component class

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