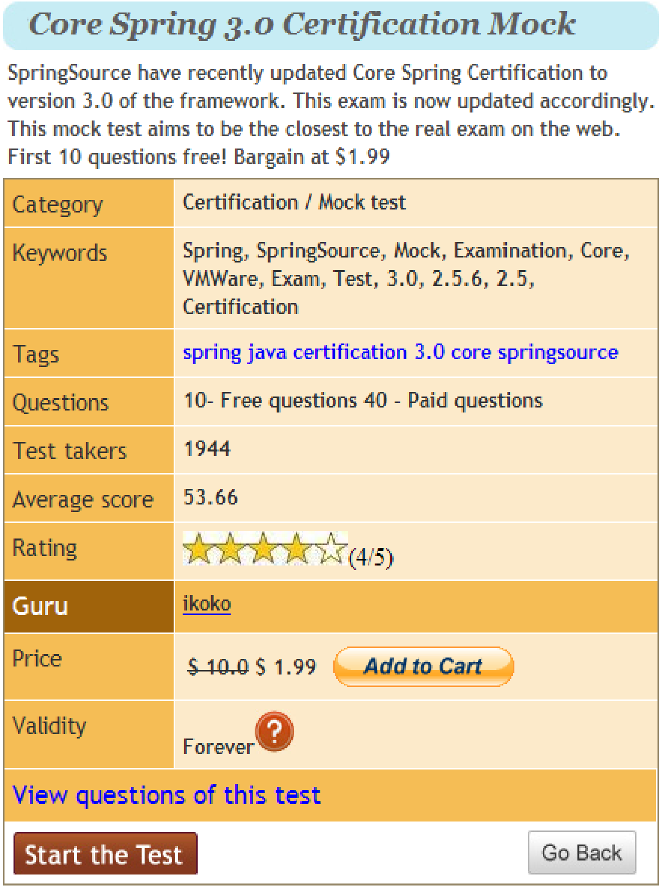
http://www.skill-guru.com/test/81/core-spring-3.0-certification-mock

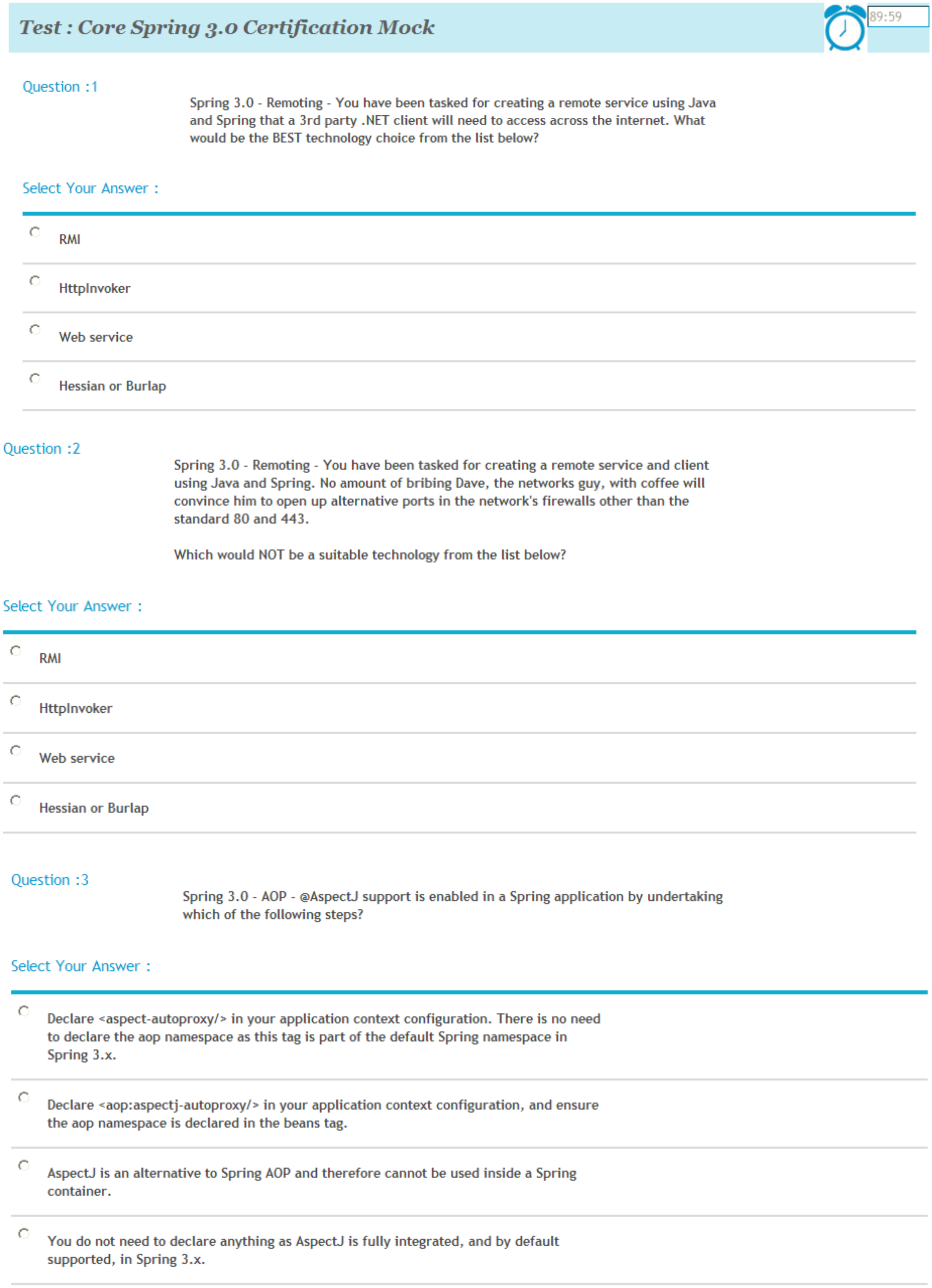


Core Spring 3.0 Certification Mock

SpringSource have recently updated Core Spring Certification to version 3.0 of the framework. This exam is now updated accordingly. This mock test aims to be the closest to the real exam on the web. First 10 questions free! Bargain at $1.99

Category	Certification / Mock test
Keywords	Spring, SpringSource, Mock, Examination, Core, VMWare, Exam, Test, 3.0, 2.5.6, 2.5, Certification
Tags	spring java certification 3.0 core springsource
Questions	10- Free questions 40 - Paid questions
Test takers	1944
Average score	53.66
Rating	★★★★☆ (4/5)
Guru	ikoko
Price	~~$ 10.0~~ $ 1.99 Add to Cart
Validity	Forever ?

View questions of this test



Test : Core Spring 3.0 Certification Mock

89:59

Question :1

Spring 3.0 - Remoting - You have been tasked for creating a remote service using Java and Spring that a 3rd party .NET client will need to access across the internet. What would be the BEST technology choice from the list below?

Select Your Answer :

○ RMI

○ HttpInvoker

○ Web service

○ Hessian or Burlap

Question :2

Spring 3.0 - Remoting - You have been tasked for creating a remote service and client using Java and Spring. No amount of bribing Dave, the networks guy, with coffee will convince him to open up alternative ports in the network's firewalls other than the standard 80 and 443.

Which would NOT be a suitable technology from the list below?

Select Your Answer :

○ RMI

○ HttpInvoker

○ Web service

○ Hessian or Burlap

Question :3

Spring 3.0 - AOP - @AspectJ support is enabled in a Spring application by undertaking which of the following steps?

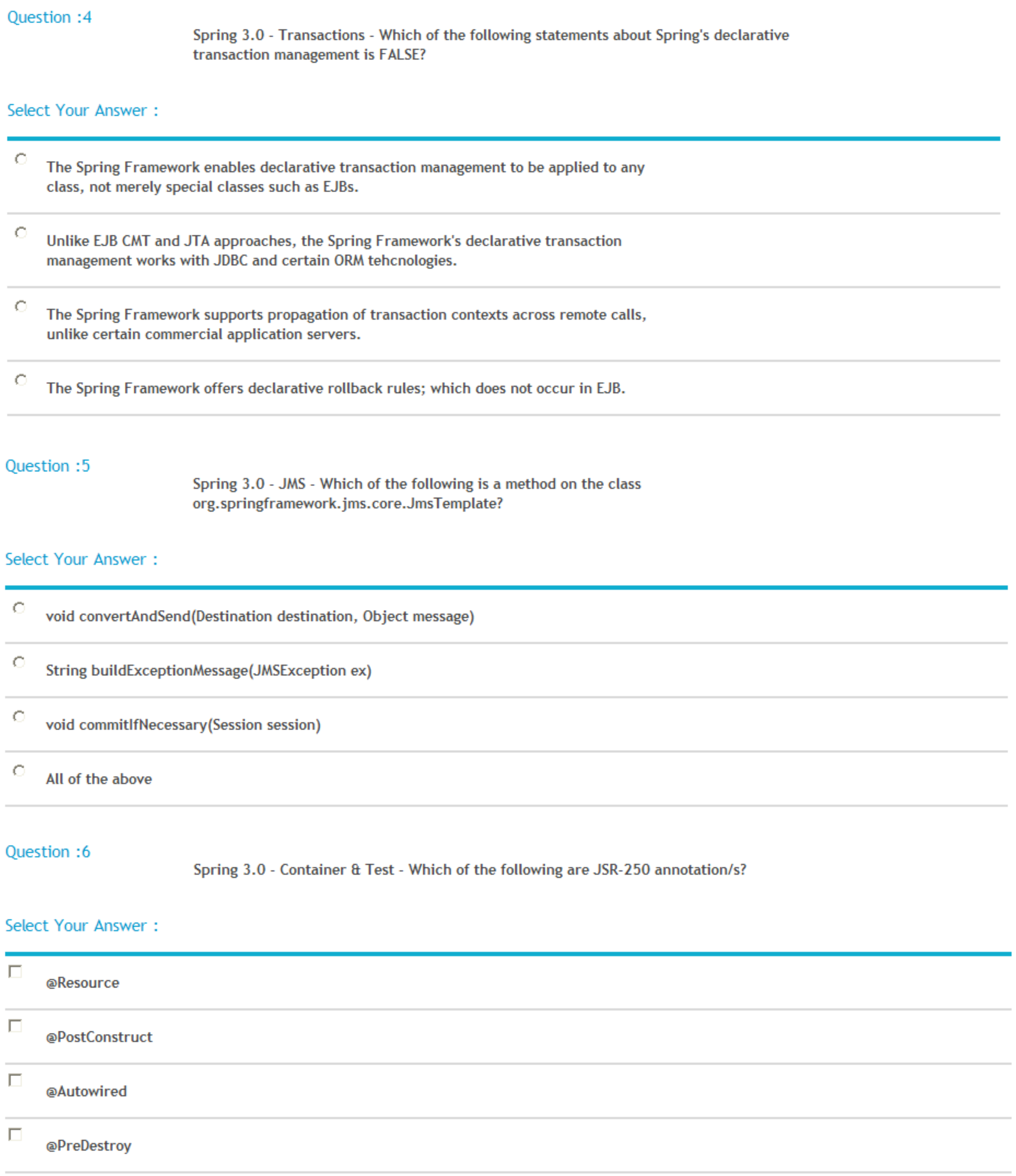
Select Your Answer :

○ Declare <aspect-autoproxy/> in your application context configuration. There is no need to declare the aop namespace as this tag is part of the default Spring namespace in Spring 3.x.

○ Declare <aop:aspectj-autoproxy/> in your application context configuration, and ensure the aop namespace is declared in the beans tag.

○ AspectJ is an alternative to Spring AOP and therefore cannot be used inside a Spring container.

○ You do not need to declare anything as AspectJ is fully integrated, and by default supported, in Spring 3.x.



Question :4

Spring 3.0 - Transactions - Which of the following statements about Spring's declarative transaction management is FALSE?

Select Your Answer :

○ The Spring Framework enables declarative transaction management to be applied to any class, not merely special classes such as EJBs.

○ Unlike EJB CMT and JTA approaches, the Spring Framework's declarative transaction management works with JDBC and certain ORM tehcnologies.

○ The Spring Framework supports propagation of transaction contexts across remote calls, unlike certain commercial application servers.

○ The Spring Framework offers declarative rollback rules; which does not occur in EJB.

Question :5

Spring 3.0 - JMS - Which of the following is a method on the class org.springframework.jms.core.JmsTemplate?

Select Your Answer :

○ void convertAndSend(Destination destination, Object message)

○ String buildExceptionMessage(JMSException ex)

○ void commitIfNecessary(Session session)

○ All of the above

Question :6

Spring 3.0 - Container & Test - Which of the following are JSR-250 annotation/s?

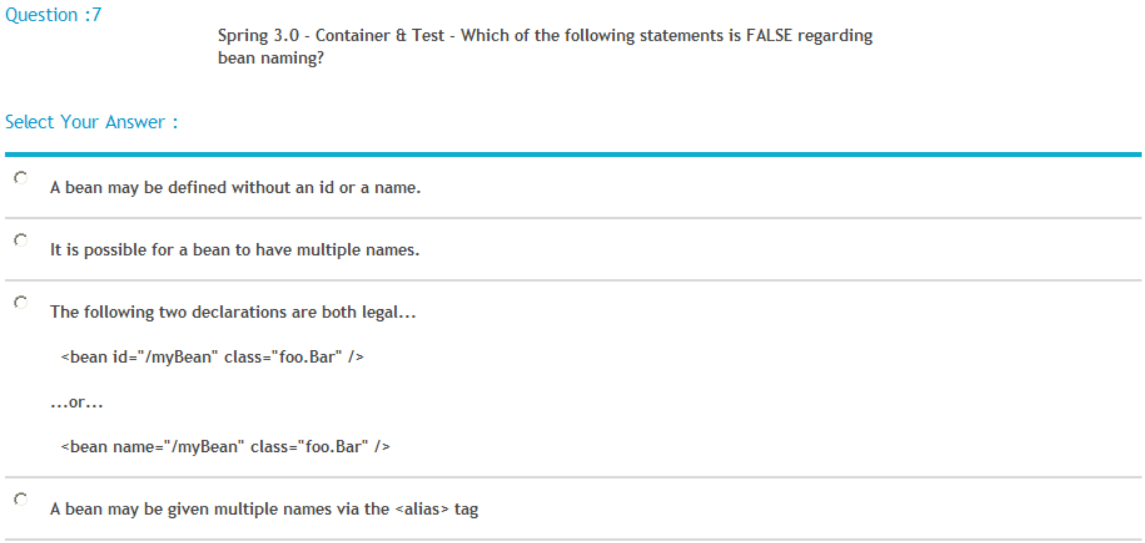
Select Your Answer :

□ @Resource

□ @PostConstruct

□ @Autowired

□ @PreDestroy



Question :7

Spring 3.0 - Container & Test - Which of the following statements is FALSE regarding bean naming?

Select Your Answer :

○ A bean may be defined without an id or a name.

○ It is possible for a bean to have multiple names.

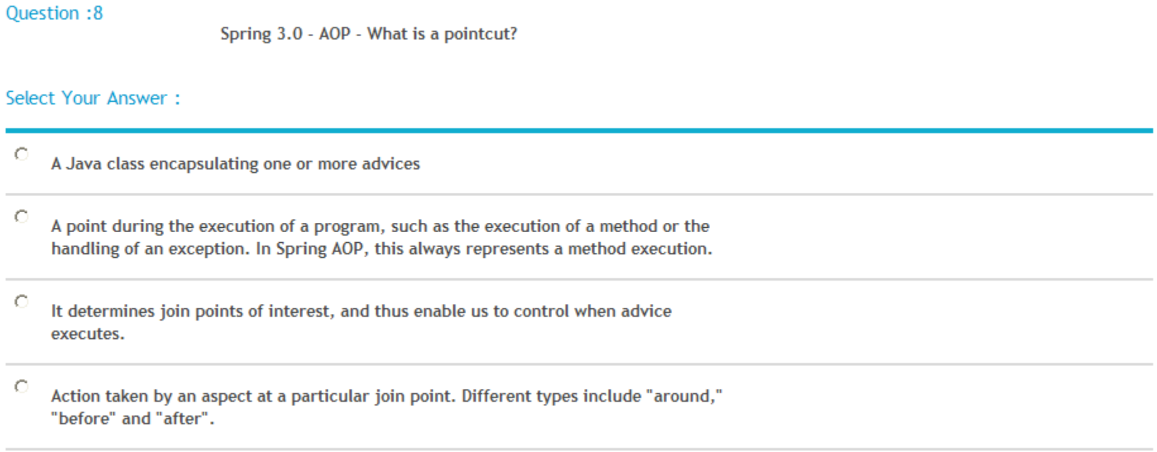
○ The following two declarations are both legal...

```
<bean id="/myBean" class="foo.Bar" />
```

...or...

```
<bean name="/myBean" class="foo.Bar" />
```

○ A bean may be given multiple names via the <alias> tag



Question :8

Spring 3.0 - AOP - What is a pointcut?

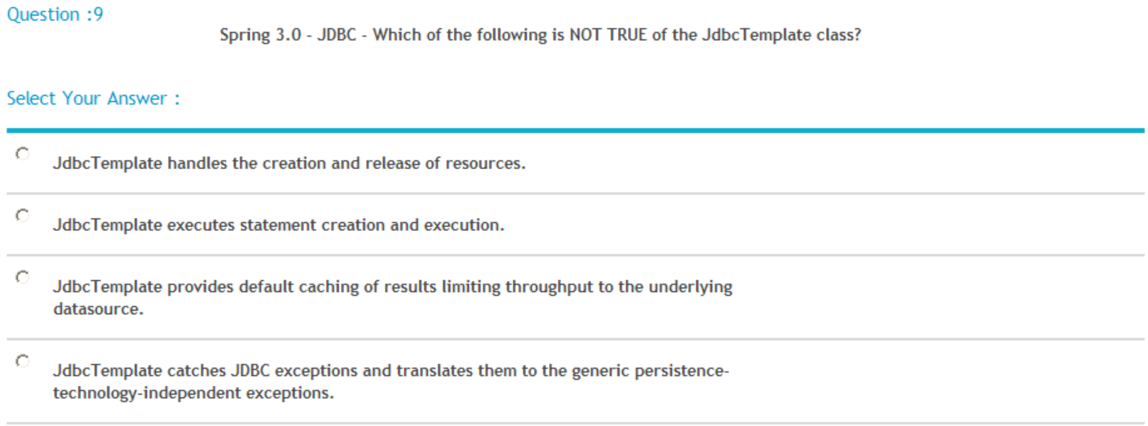
Select Your Answer :

○ A Java class encapsulating one or more advices

○ A point during the execution of a program, such as the execution of a method or the handling of an exception. In Spring AOP, this always represents a method execution.

○ It determines join points of interest, and thus enable us to control when advice executes.

○ Action taken by an aspect at a particular join point. Different types include "around," "before" and "after".



Question :9

Spring 3.0 - JDBC - Which of the following is NOT TRUE of the JdbcTemplate class?

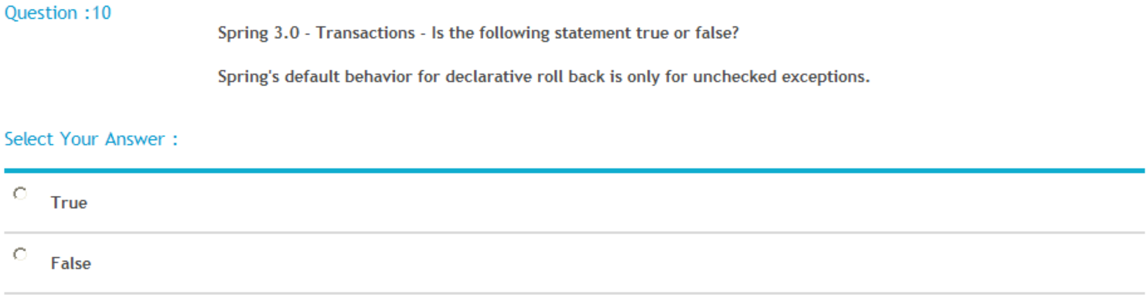
Select Your Answer :

○ JdbcTemplate handles the creation and release of resources.

○ JdbcTemplate executes statement creation and execution.

○ JdbcTemplate provides default caching of results limiting throughput to the underlying datasource.

○ JdbcTemplate catches JDBC exceptions and translates them to the generic persistence-technology-independent exceptions.



Question :10

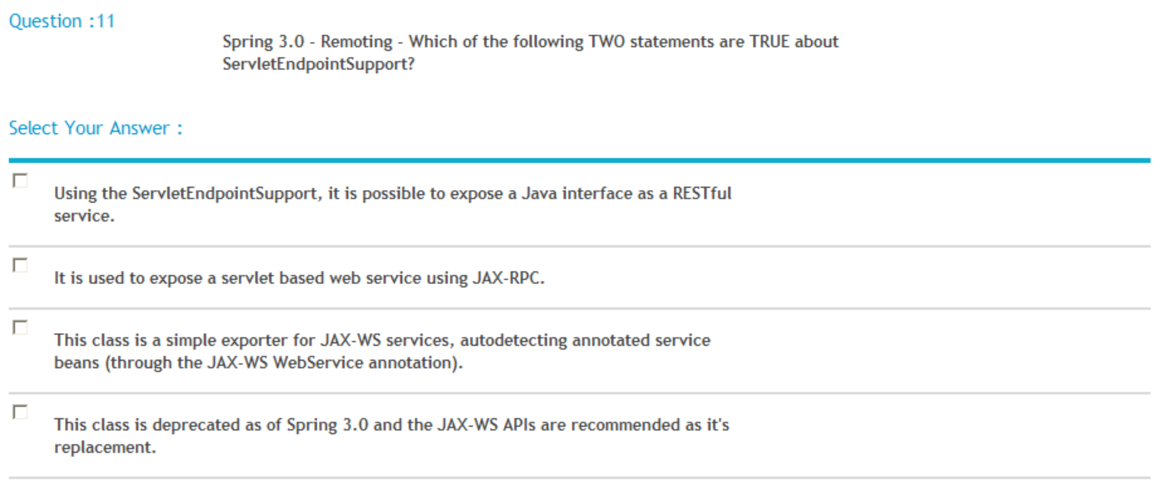
Spring 3.0 - Transactions - Is the following statement true or false?

Spring's default behavior for declarative roll back is only for unchecked exceptions.

Select Your Answer :

○ True

○ False



Question :11

Spring 3.0 - Remoting - Which of the following TWO statements are TRUE about ServletEndpointSupport?

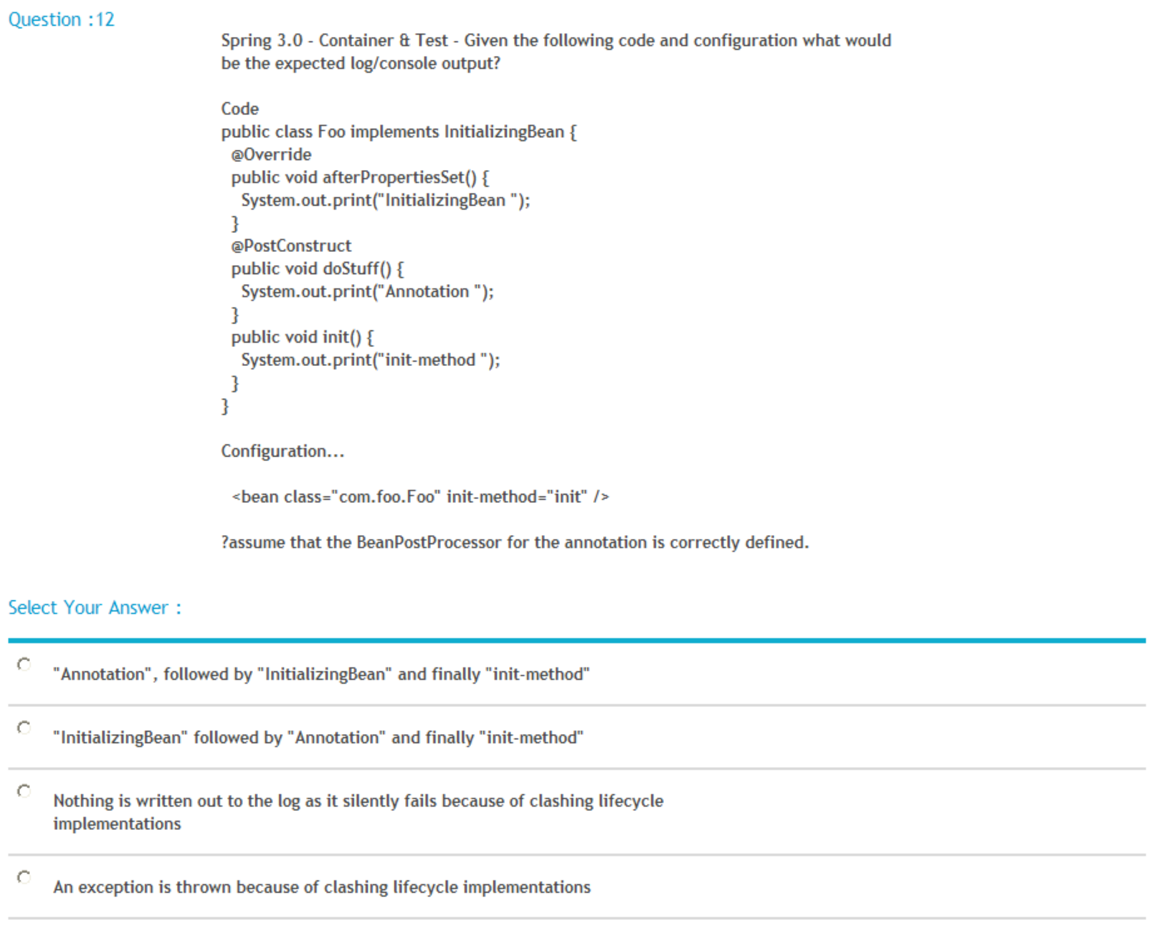
Select Your Answer :

☐ Using the ServletEndpointSupport, it is possible to expose a Java interface as a RESTful service.

☐ It is used to expose a servlet based web service using JAX-RPC.

☐ This class is a simple exporter for JAX-WS services, autodetecting annotated service beans (through the JAX-WS WebService annotation).

☐ This class is deprecated as of Spring 3.0 and the JAX-WS APIs are recommended as it's replacement.



Question :12

Spring 3.0 - Container & Test - Given the following code and configuration what would be the expected log/console output?

```
Code
public class Foo implements InitializingBean {
  @Override
  public void afterPropertiesSet() {
    System.out.print("InitializingBean ");
  }
  @PostConstruct
  public void doStuff() {
    System.out.print("Annotation ");
  }
  public void init() {
    System.out.print("init-method ");
  }
}

Configuration...

  <bean class="com.foo.Foo" init-method="init" />
```

?assume that the BeanPostProcessor for the annotation is correctly defined.

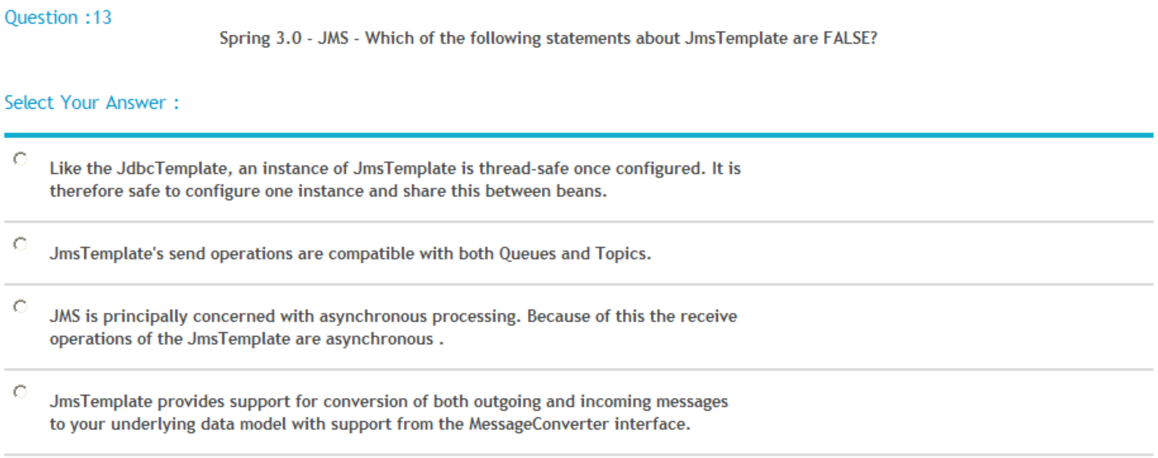
Select Your Answer :

○ "Annotation", followed by "InitializingBean" and finally "init-method"

○ "InitializingBean" followed by "Annotation" and finally "init-method"

○ Nothing is written out to the log as it silently fails because of clashing lifecycle implementations

○ An exception is thrown because of clashing lifecycle implementations



Question :13

Spring 3.0 - JMS - Which of the following statements about JmsTemplate are FALSE?

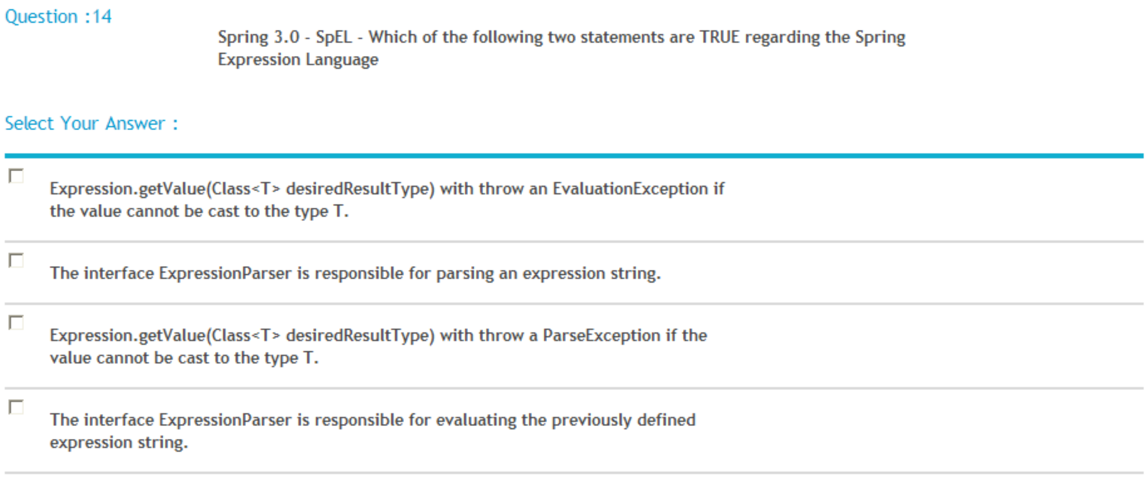
Select Your Answer :

○ Like the JdbcTemplate, an instance of JmsTemplate is thread-safe once configured. It is therefore safe to configure one instance and share this between beans.

○ JmsTemplate's send operations are compatible with both Queues and Topics.

○ JMS is principally concerned with asynchronous processing. Because of this the receive operations of the JmsTemplate are asynchronous .

○ JmsTemplate provides support for conversion of both outgoing and incoming messages to your underlying data model with support from the MessageConverter interface.



Question :14

Spring 3.0 - SpEL - Which of the following two statements are TRUE regarding the Spring Expression Language

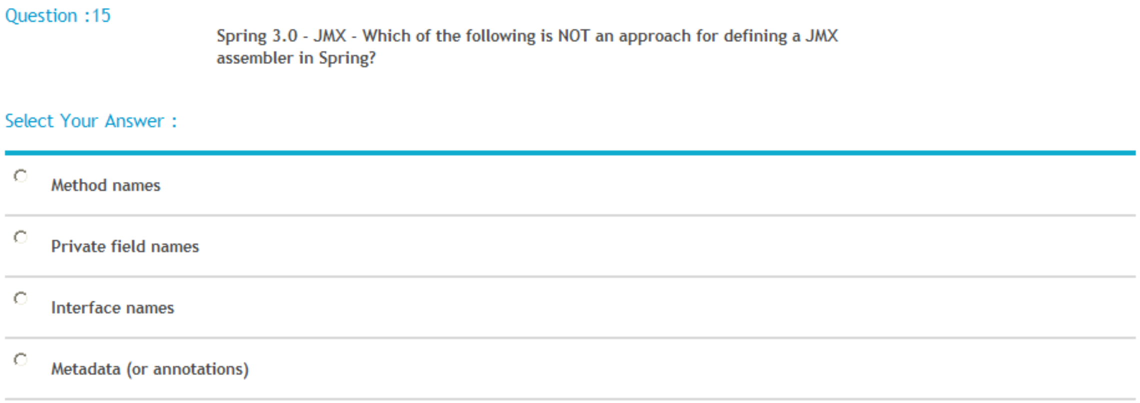
Select Your Answer :

□ Expression.getValue(Class<T> desiredResultType) with throw an EvaluationException if the value cannot be cast to the type T.

□ The interface ExpressionParser is responsible for parsing an expression string.

□ Expression.getValue(Class<T> desiredResultType) with throw a ParseException if the value cannot be cast to the type T.

□ The interface ExpressionParser is responsible for evaluating the previously defined expression string.



Question :15

Spring 3.0 - JMX - Which of the following is NOT an approach for defining a JMX assembler in Spring?

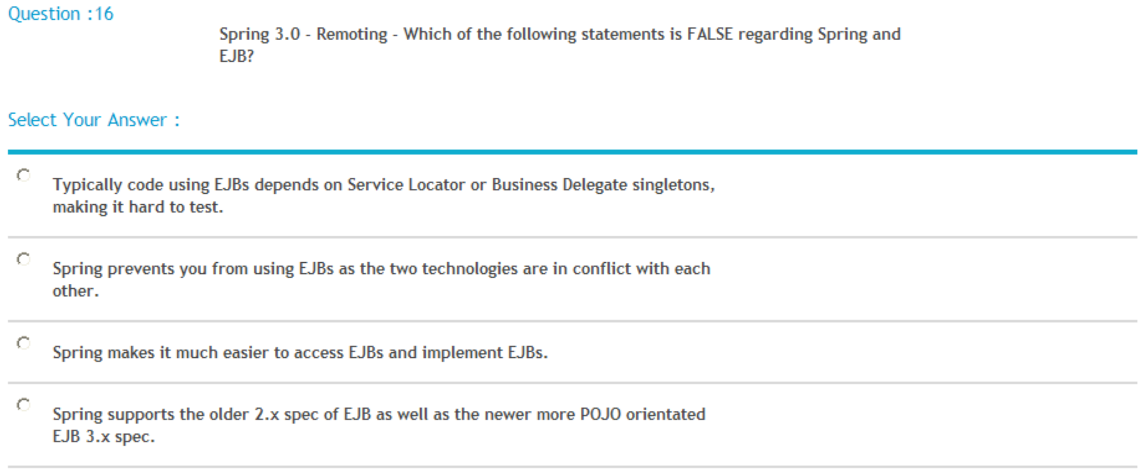
Select Your Answer :

○ Method names

○ Private field names

○ Interface names

○ Metadata (or annotations)



Question :16

Spring 3.0 - Remoting - Which of the following statements is FALSE regarding Spring and EJB?

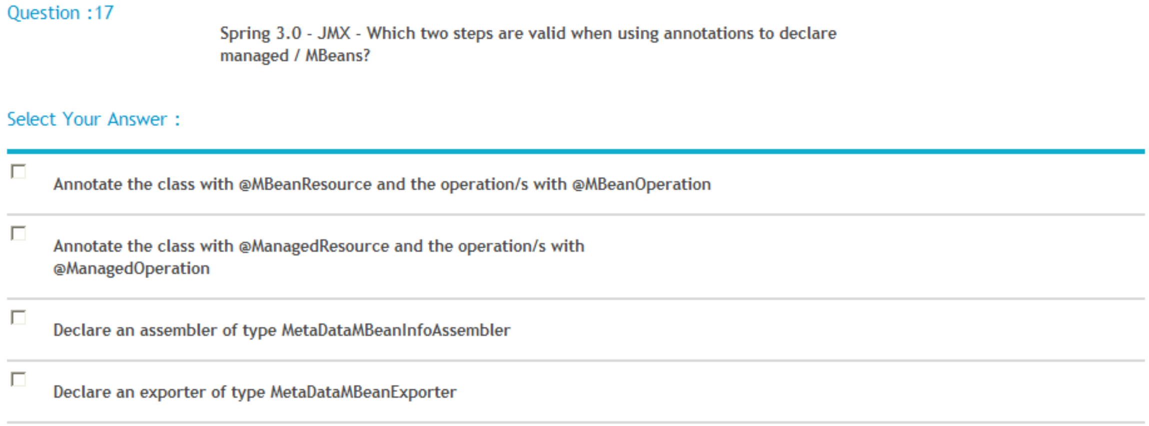
Select Your Answer :

○ Typically code using EJBs depends on Service Locator or Business Delegate singletons, making it hard to test.

○ Spring prevents you from using EJBs as the two technologies are in conflict with each other.

○ Spring makes it much easier to access EJBs and implement EJBs.

○ Spring supports the older 2.x spec of EJB as well as the newer more POJO orientated EJB 3.x spec.



Question :17

Spring 3.0 - JMX - Which two steps are valid when using annotations to declare managed / MBeans?

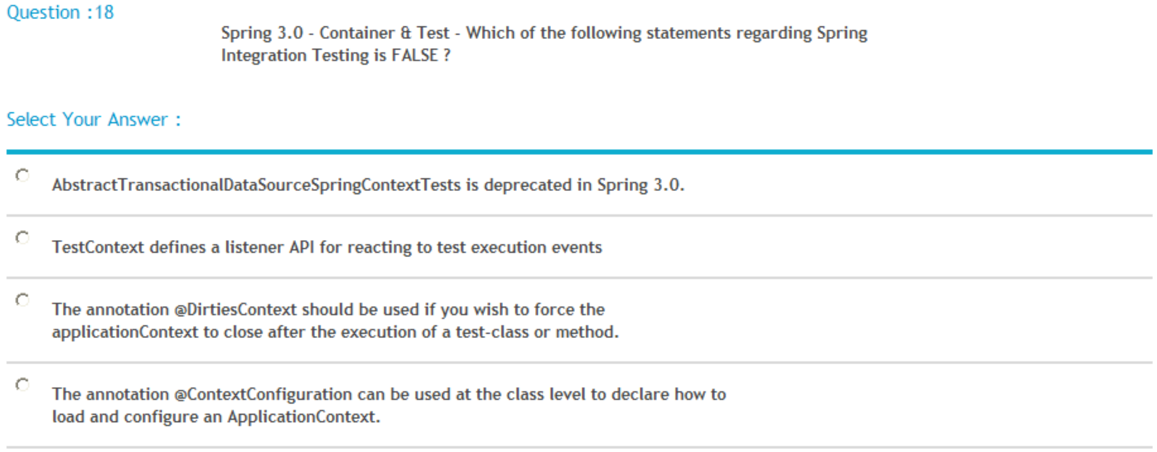
Select Your Answer :

□ Annotate the class with @MBeanResource and the operation/s with @MBeanOperation

□ Annotate the class with @ManagedResource and the operation/s with @ManagedOperation

□ Declare an assembler of type MetaDataMBeanInfoAssembler

□ Declare an exporter of type MetaDataMBeanExporter



Question :18

Spring 3.0 - Container & Test - Which of the following statements regarding Spring Integration Testing is FALSE ?

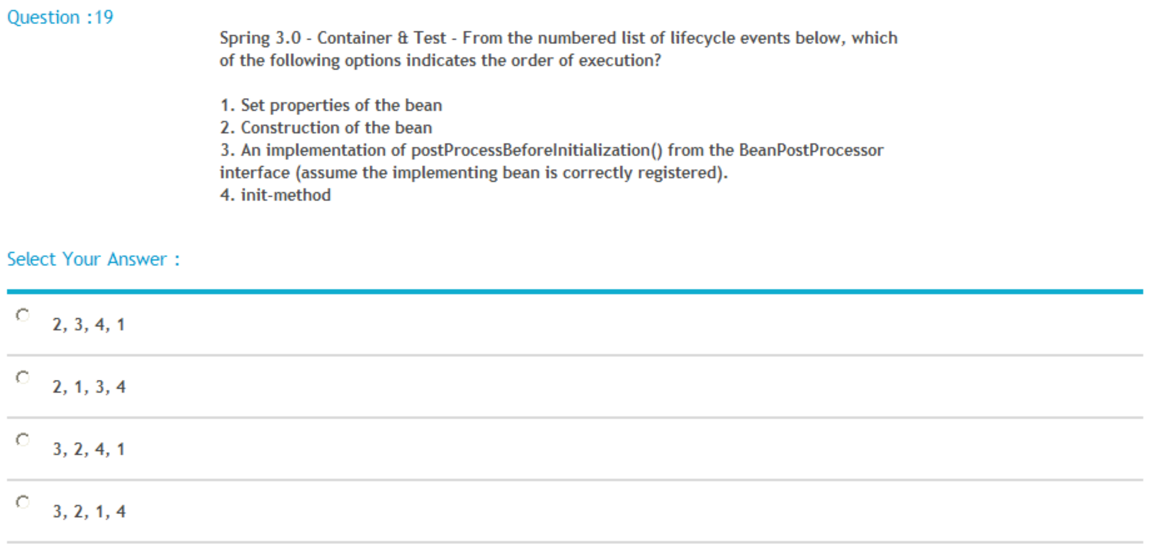
Select Your Answer :

○ AbstractTransactionalDataSourceSpringContextTests is deprecated in Spring 3.0.

○ TestContext defines a listener API for reacting to test execution events

○ The annotation @DirtiesContext should be used if you wish to force the applicationContext to close after the execution of a test-class or method.

○ The annotation @ContextConfiguration can be used at the class level to declare how to load and configure an ApplicationContext.



Question :19

Spring 3.0 - Container & Test - From the numbered list of lifecycle events below, which of the following options indicates the order of execution?

1. Set properties of the bean
2. Construction of the bean
3. An implementation of postProcessBeforeInitialization() from the BeanPostProcessor interface (assume the implementing bean is correctly registered).
4. init-method

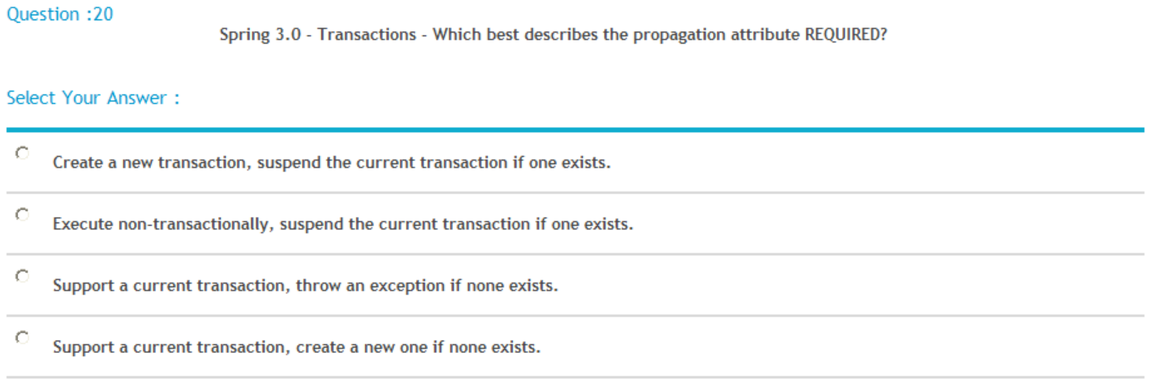
Select Your Answer :

○ 2, 3, 4, 1

○ 2, 1, 3, 4

○ 3, 2, 4, 1

○ 3, 2, 1, 4



Question :20

Spring 3.0 - Transactions - Which best describes the propagation attribute REQUIRED?

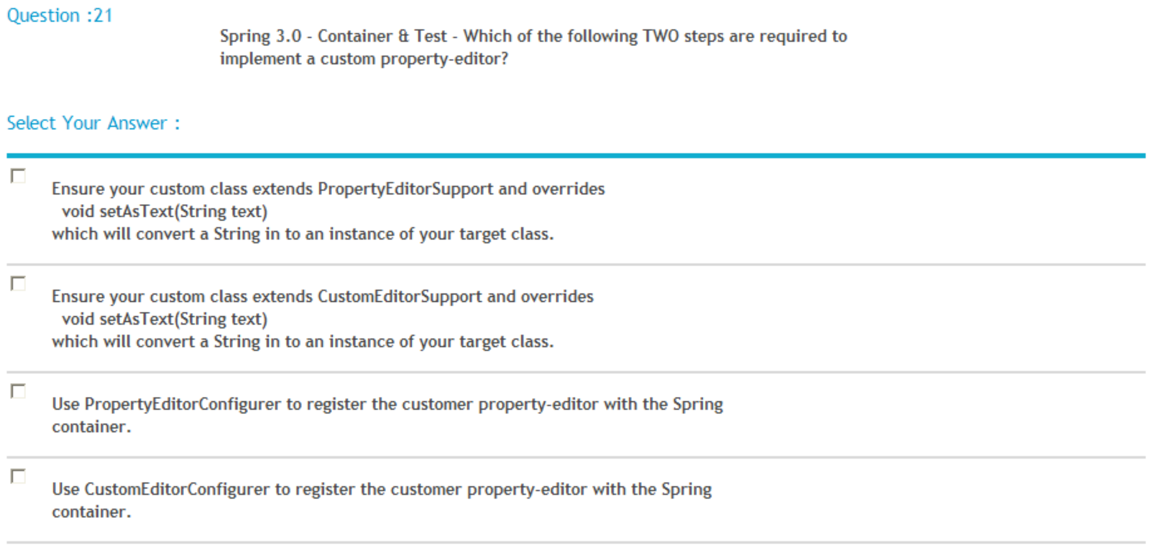
Select Your Answer :

○ Create a new transaction, suspend the current transaction if one exists.

○ Execute non-transactionally, suspend the current transaction if one exists.

○ Support a current transaction, throw an exception if none exists.

○ Support a current transaction, create a new one if none exists.



Question :21

Spring 3.0 - Container & Test - Which of the following TWO steps are required to implement a custom property-editor?

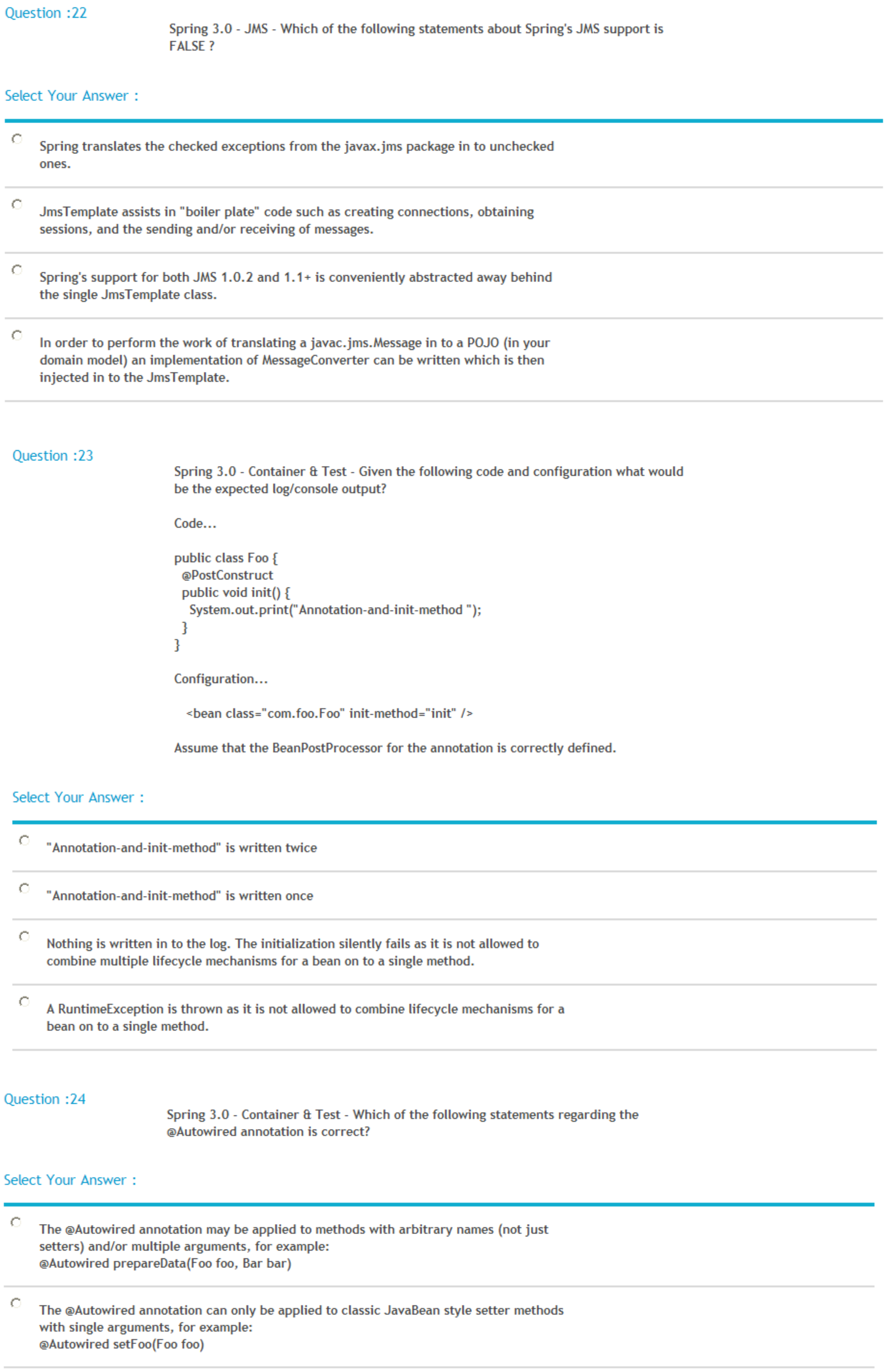
Select Your Answer :

□ Ensure your custom class extends PropertyEditorSupport and overrides
void setAsText(String text)
which will convert a String in to an instance of your target class.

□ Ensure your custom class extends CustomEditorSupport and overrides
void setAsText(String text)
which will convert a String in to an instance of your target class.

□ Use PropertyEditorConfigurer to register the customer property-editor with the Spring container.

□ Use CustomEditorConfigurer to register the customer property-editor with the Spring container.



Question :22

Spring 3.0 - JMS - Which of the following statements about Spring's JMS support is FALSE ?

Select Your Answer :

○ Spring translates the checked exceptions from the javax.jms package in to unchecked ones.

○ JmsTemplate assists in "boiler plate" code such as creating connections, obtaining sessions, and the sending and/or receiving of messages.

○ Spring's support for both JMS 1.0.2 and 1.1+ is conveniently abstracted away behind the single JmsTemplate class.

○ In order to perform the work of translating a javac.jms.Message in to a POJO (in your domain model) an implementation of MessageConverter can be written which is then injected in to the JmsTemplate.

Question :23

Spring 3.0 - Container & Test - Given the following code and configuration what would be the expected log/console output?

Code...

```
public class Foo {
  @PostConstruct
  public void init() {
    System.out.print("Annotation-and-init-method ");
  }
}
```

Configuration...

```
   <bean class="com.foo.Foo" init-method="init" />
```

Assume that the BeanPostProcessor for the annotation is correctly defined.

Select Your Answer :

○ "Annotation-and-init-method" is written twice

○ "Annotation-and-init-method" is written once

○ Nothing is written in to the log. The initialization silently fails as it is not allowed to combine multiple lifecycle mechanisms for a bean on to a single method.

○ A RuntimeException is thrown as it is not allowed to combine lifecycle mechanisms for a bean on to a single method.

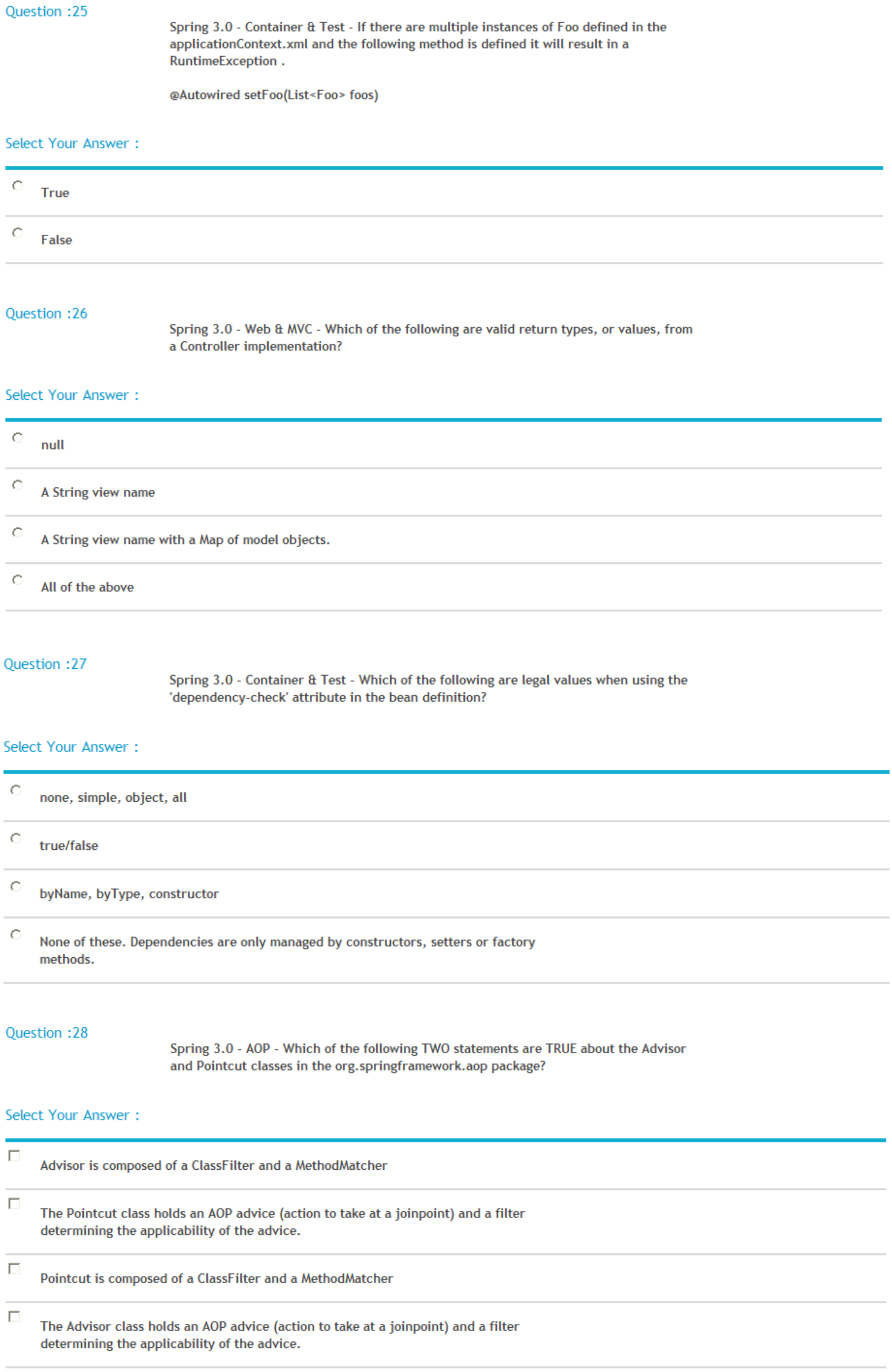
Question :24

Spring 3.0 - Container & Test - Which of the following statements regarding the @Autowired annotation is correct?

Select Your Answer :

○ The @Autowired annotation may be applied to methods with arbitrary names (not just setters) and/or multiple arguments, for example:
@Autowired prepareData(Foo foo, Bar bar)

○ The @Autowired annotation can only be applied to classic JavaBean style setter methods with single arguments, for example:
@Autowired setFoo(Foo foo)



Question :25

Spring 3.0 - Container & Test - If there are multiple instances of Foo defined in the applicationContext.xml and the following method is defined it will result in a RuntimeException .

@Autowired setFoo(List<Foo> foos)

Select Your Answer :

○ True

○ False

Question :26

Spring 3.0 - Web & MVC - Which of the following are valid return types, or values, from a Controller implementation?

Select Your Answer :

○ null

○ A String view name

○ A String view name with a Map of model objects.

○ All of the above

Question :27

Spring 3.0 - Container & Test - Which of the following are legal values when using the 'dependency-check' attribute in the bean definition?

Select Your Answer :

○ none, simple, object, all

○ true/false

○ byName, byType, constructor

○ None of these. Dependencies are only managed by constructors, setters or factory methods.

Question :28

Spring 3.0 - AOP - Which of the following TWO statements are TRUE about the Advisor and Pointcut classes in the org.springframework.aop package?

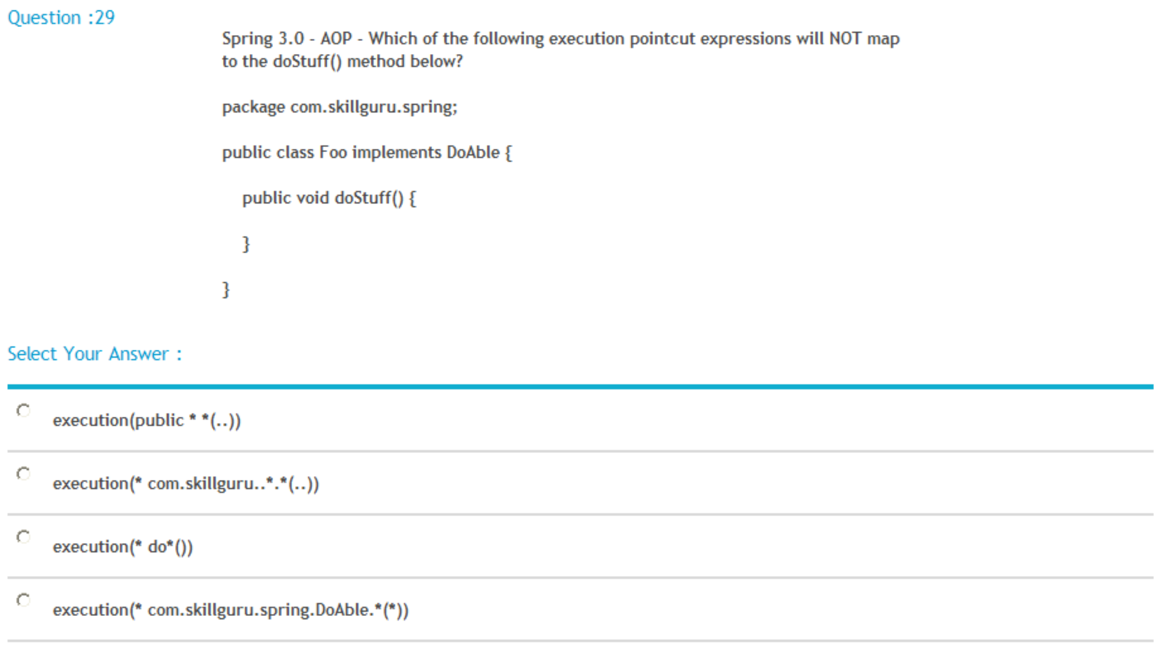
Select Your Answer :

□ Advisor is composed of a ClassFilter and a MethodMatcher

□ The Pointcut class holds an AOP advice (action to take at a joinpoint) and a filter determining the applicability of the advice.

□ Pointcut is composed of a ClassFilter and a MethodMatcher

□ The Advisor class holds an AOP advice (action to take at a joinpoint) and a filter determining the applicability of the advice.



Question :29

Spring 3.0 - AOP - Which of the following execution pointcut expressions will NOT map to the doStuff() method below?

```
package com.skillguru.spring;

public class Foo implements DoAble {

    public void doStuff() {

    }

}
```

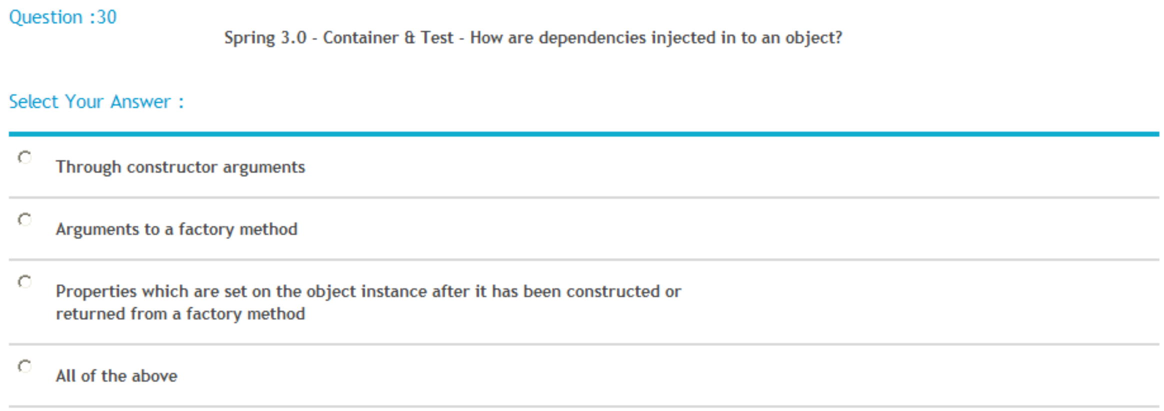
Select Your Answer :

○ execution(public * *(..))

○ execution(* com.skillguru..*.*(..))

○ execution(* do*())

○ execution(* com.skillguru.spring.DoAble.*(*))



Question :30

Spring 3.0 - Container & Test - How are dependencies injected in to an object?

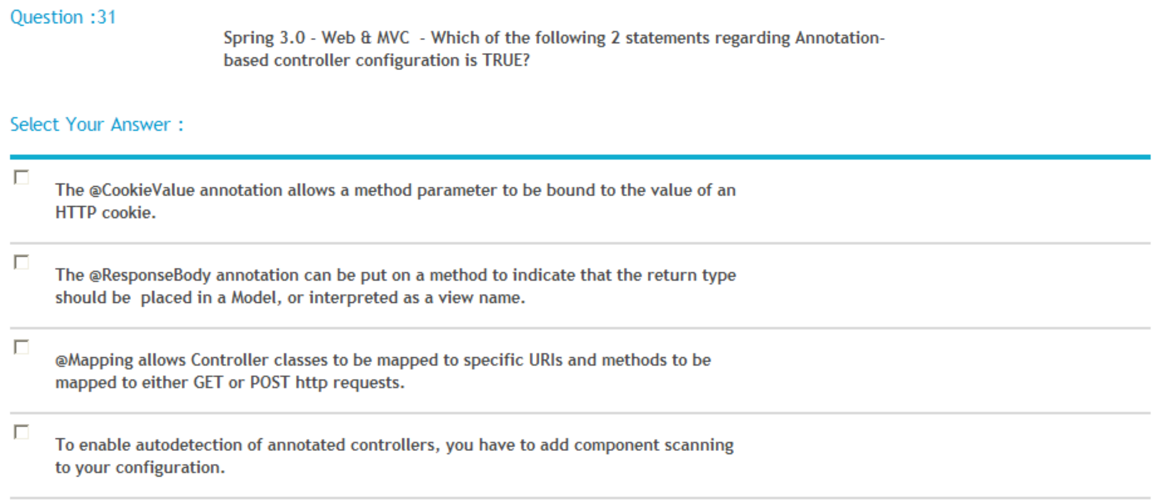
Select Your Answer :

○ Through constructor arguments

○ Arguments to a factory method

○ Properties which are set on the object instance after it has been constructed or returned from a factory method

○ All of the above



Question :31

Spring 3.0 - Web & MVC - Which of the following 2 statements regarding Annotation-based controller configuration is TRUE?

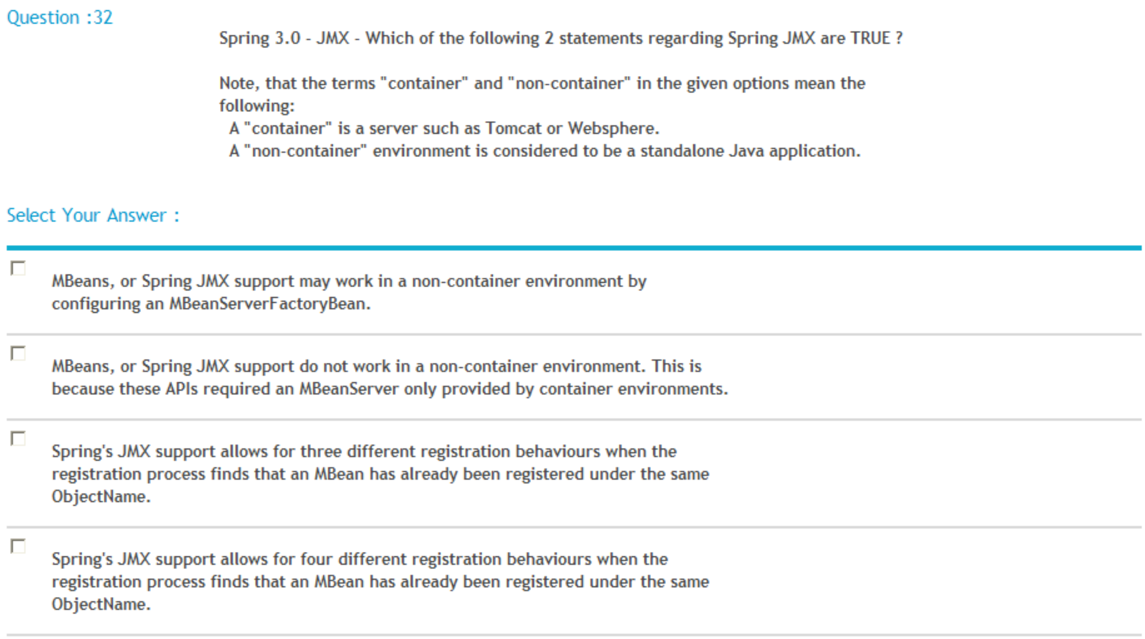
Select Your Answer :

□ The @CookieValue annotation allows a method parameter to be bound to the value of an HTTP cookie.

□ The @ResponseBody annotation can be put on a method to indicate that the return type should be placed in a Model, or interpreted as a view name.

□ @Mapping allows Controller classes to be mapped to specific URIs and methods to be mapped to either GET or POST http requests.

□ To enable autodetection of annotated controllers, you have to add component scanning to your configuration.



Question :32

Spring 3.0 - JMX - Which of the following 2 statements regarding Spring JMX are TRUE ?

Note, that the terms "container" and "non-container" in the given options mean the following:
A "container" is a server such as Tomcat or Websphere.
A "non-container" environment is considered to be a standalone Java application.

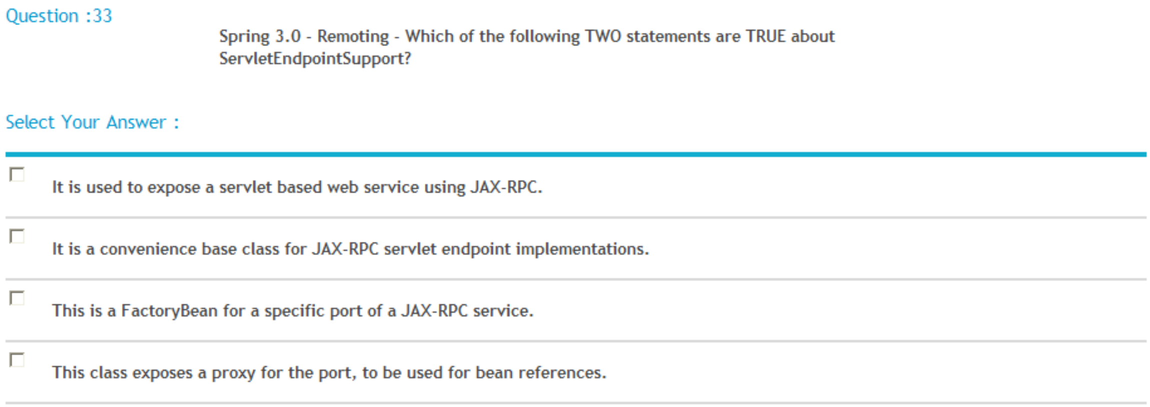
Select Your Answer :

□ MBeans, or Spring JMX support may work in a non-container environment by configuring an MBeanServerFactoryBean.

□ MBeans, or Spring JMX support do not work in a non-container environment. This is because these APIs required an MBeanServer only provided by container environments.

□ Spring's JMX support allows for three different registration behaviours when the registration process finds that an MBean has already been registered under the same ObjectName.

□ Spring's JMX support allows for four different registration behaviours when the registration process finds that an MBean has already been registered under the same ObjectName.



Question :33

Spring 3.0 - Remoting - Which of the following TWO statements are TRUE about ServletEndpointSupport?

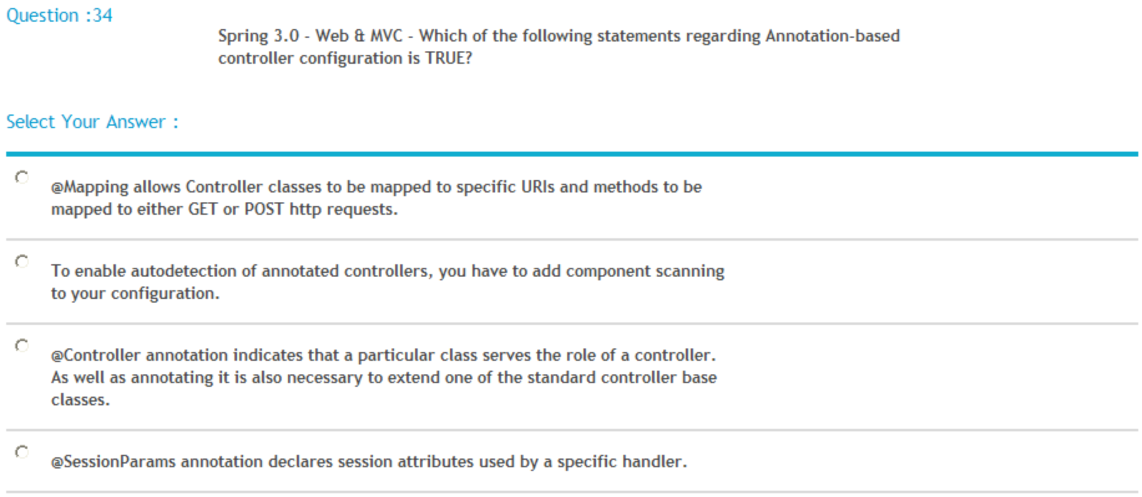
Select Your Answer :

□ It is used to expose a servlet based web service using JAX-RPC.

□ It is a convenience base class for JAX-RPC servlet endpoint implementations.

□ This is a FactoryBean for a specific port of a JAX-RPC service.

□ This class exposes a proxy for the port, to be used for bean references.



Question :34

Spring 3.0 - Web & MVC - Which of the following statements regarding Annotation-based controller configuration is TRUE?

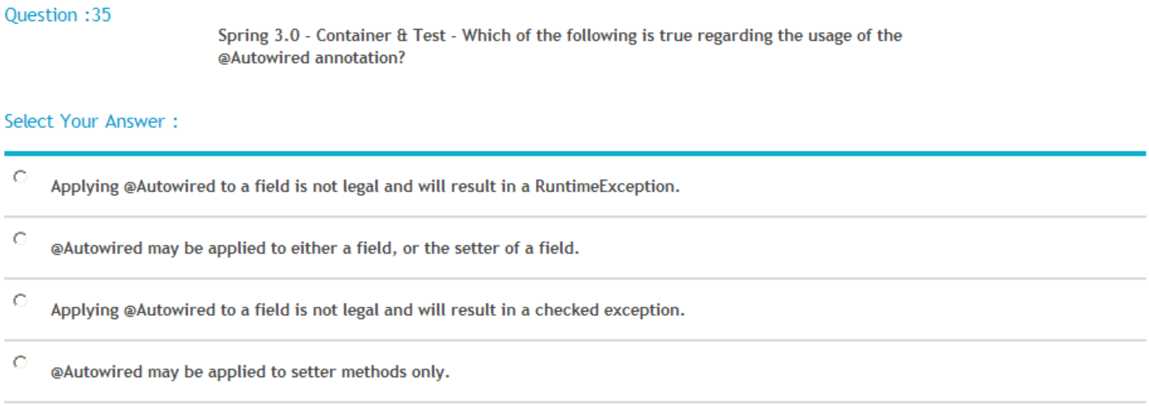
Select Your Answer :

○ @Mapping allows Controller classes to be mapped to specific URIs and methods to be mapped to either GET or POST http requests.

○ To enable autodetection of annotated controllers, you have to add component scanning to your configuration.

○ @Controller annotation indicates that a particular class serves the role of a controller. As well as annotating it is also necessary to extend one of the standard controller base classes.

○ @SessionParams annotation declares session attributes used by a specific handler.



Question :35

Spring 3.0 - Container & Test - Which of the following is true regarding the usage of the @Autowired annotation?

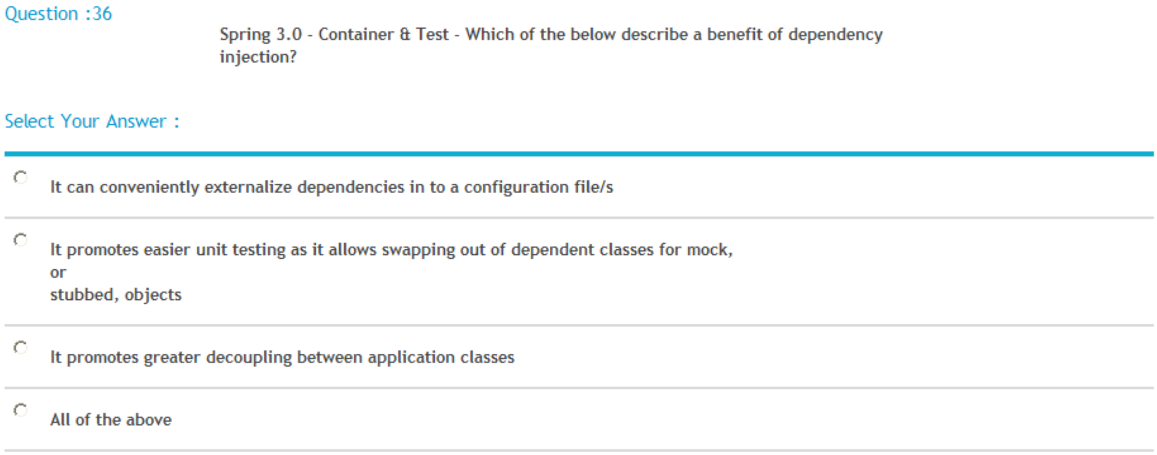
Select Your Answer :

○ Applying @Autowired to a field is not legal and will result in a RuntimeException.

○ @Autowired may be applied to either a field, or the setter of a field.

○ Applying @Autowired to a field is not legal and will result in a checked exception.

○ @Autowired may be applied to setter methods only.



Question :36

Spring 3.0 - Container & Test - Which of the below describe a benefit of dependency injection?

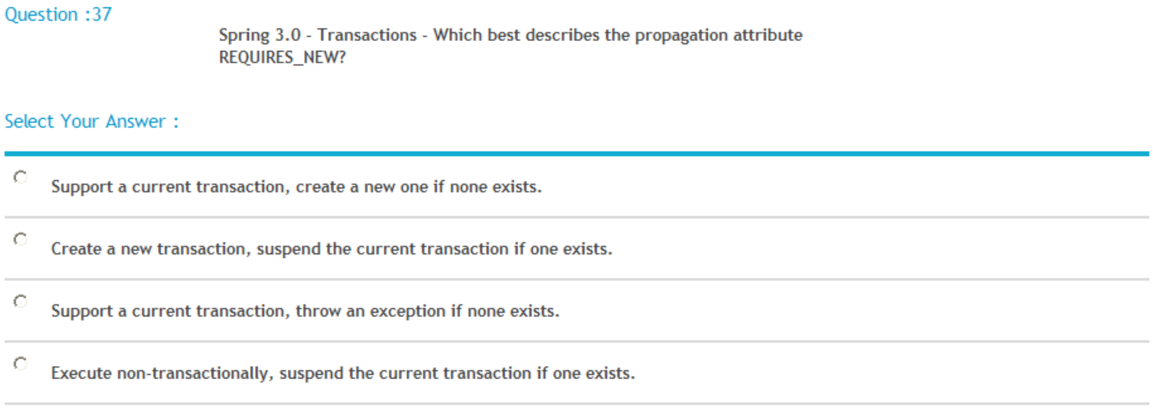
Select Your Answer :

○ It can conveniently externalize dependencies in to a configuration file/s

○ It promotes easier unit testing as it allows swapping out of dependent classes for mock,
or
stubbed, objects

○ It promotes greater decoupling between application classes

○ All of the above



Question :37

Spring 3.0 - Transactions - Which best describes the propagation attribute REQUIRES_NEW?

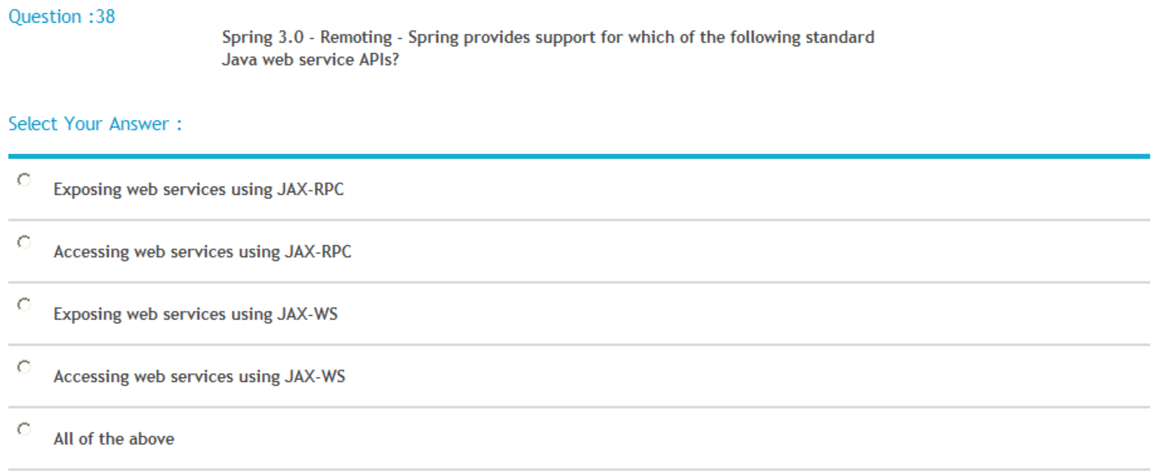
Select Your Answer :

○ Support a current transaction, create a new one if none exists.

○ Create a new transaction, suspend the current transaction if one exists.

○ Support a current transaction, throw an exception if none exists.

○ Execute non-transactionally, suspend the current transaction if one exists.



Question :38

Spring 3.0 - Remoting - Spring provides support for which of the following standard Java web service APIs?

Select Your Answer :

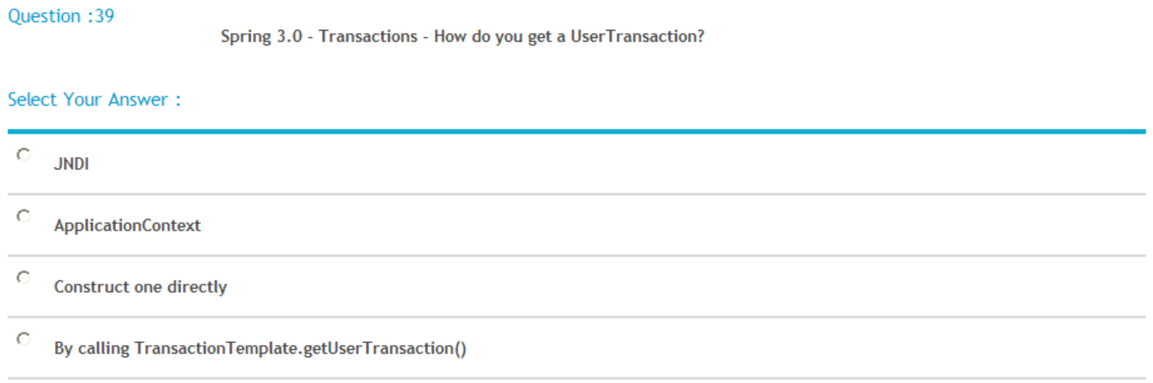
○ Exposing web services using JAX-RPC

○ Accessing web services using JAX-RPC

○ Exposing web services using JAX-WS

○ Accessing web services using JAX-WS

○ All of the above



Question :39

Spring 3.0 - Transactions - How do you get a UserTransaction?

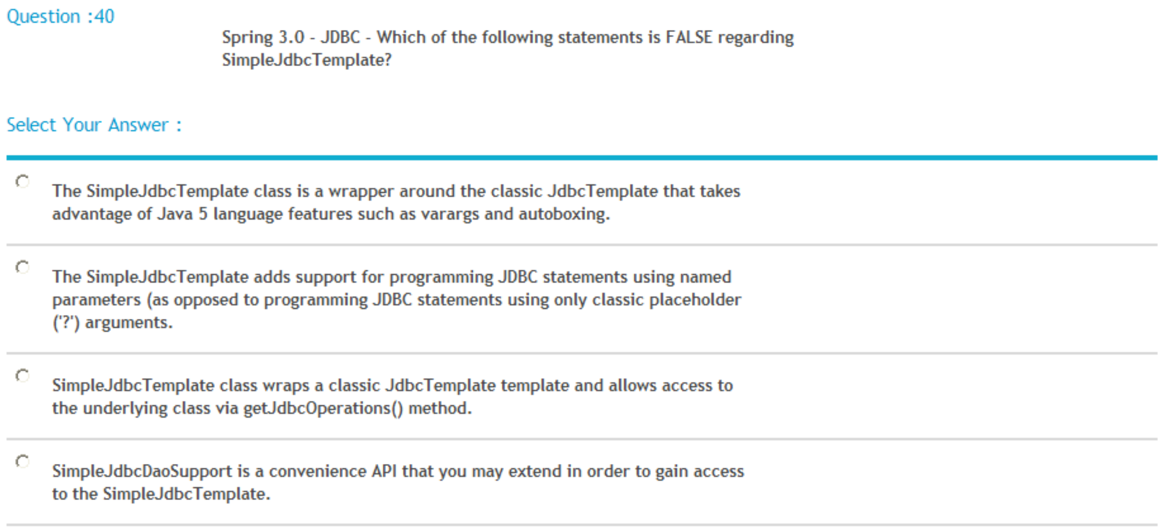
Select Your Answer :

○ JNDI

○ ApplicationContext

○ Construct one directly

○ By calling TransactionTemplate.getUserTransaction()



Question :40

Spring 3.0 - JDBC - Which of the following statements is FALSE regarding SimpleJdbcTemplate?

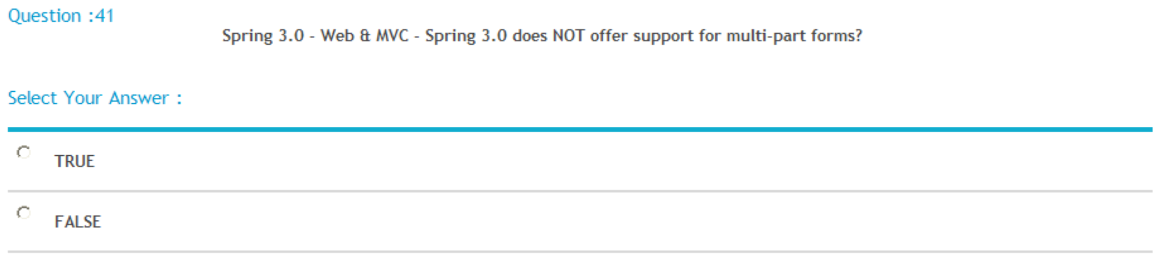
Select Your Answer :

○ The SimpleJdbcTemplate class is a wrapper around the classic JdbcTemplate that takes advantage of Java 5 language features such as varargs and autoboxing.

○ The SimpleJdbcTemplate adds support for programming JDBC statements using named parameters (as opposed to programming JDBC statements using only classic placeholder ('?') arguments.

○ SimpleJdbcTemplate class wraps a classic JdbcTemplate template and allows access to the underlying class via getJdbcOperations() method.

○ SimpleJdbcDaoSupport is a convenience API that you may extend in order to gain access to the SimpleJdbcTemplate.



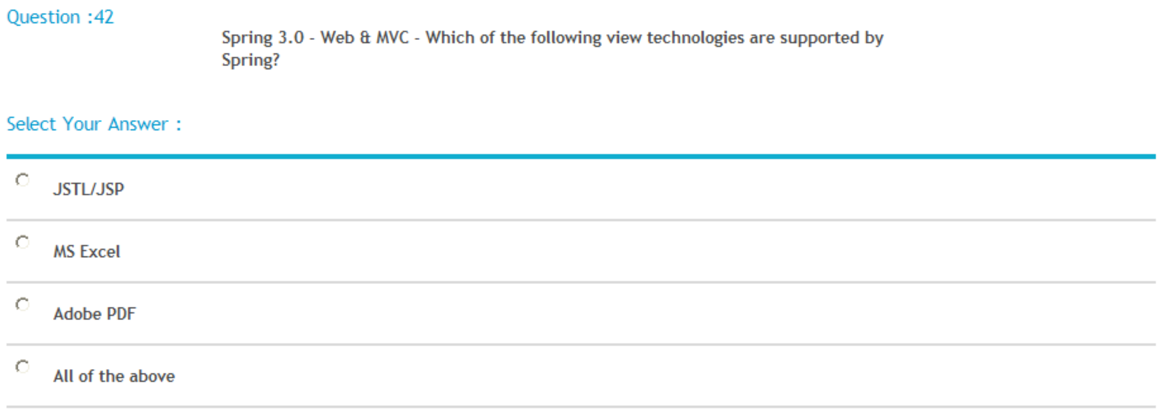
Question :41

Spring 3.0 - Web & MVC - Spring 3.0 does NOT offer support for multi-part forms?

Select Your Answer :

○ TRUE

○ FALSE



Question :42

Spring 3.0 - Web & MVC - Which of the following view technologies are supported by Spring?

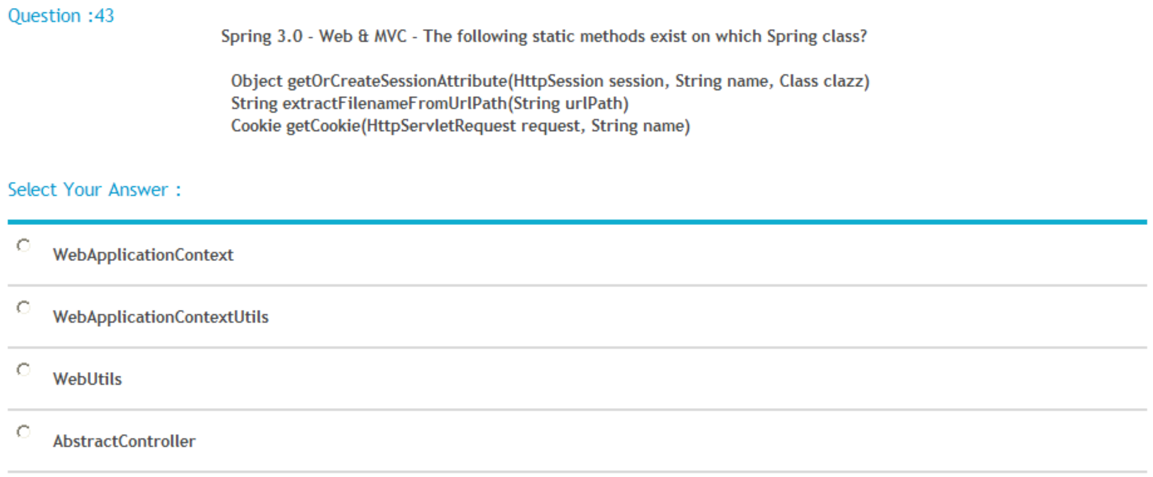
Select Your Answer :

○ JSTL/JSP

○ MS Excel

○ Adobe PDF

○ All of the above



Question :43

Spring 3.0 - Web & MVC - The following static methods exist on which Spring class?

Object getOrCreateSessionAttribute(HttpSession session, String name, Class clazz)
String extractFilenameFromUrlPath(String urlPath)
Cookie getCookie(HttpServletRequest request, String name)

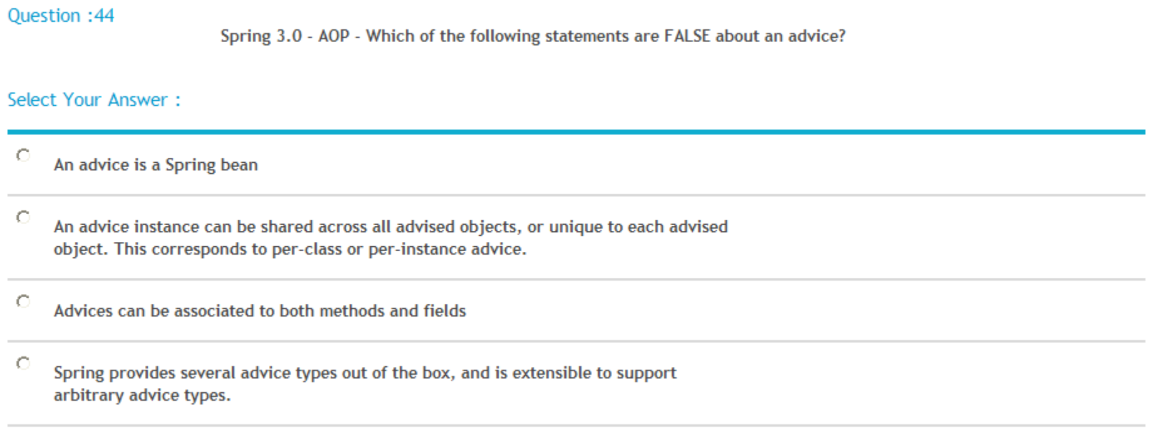
Select Your Answer :

○ WebApplicationContext

○ WebApplicationContextUtils

○ WebUtils

○ AbstractController



Question :44

Spring 3.0 - AOP - Which of the following statements are FALSE about an advice?

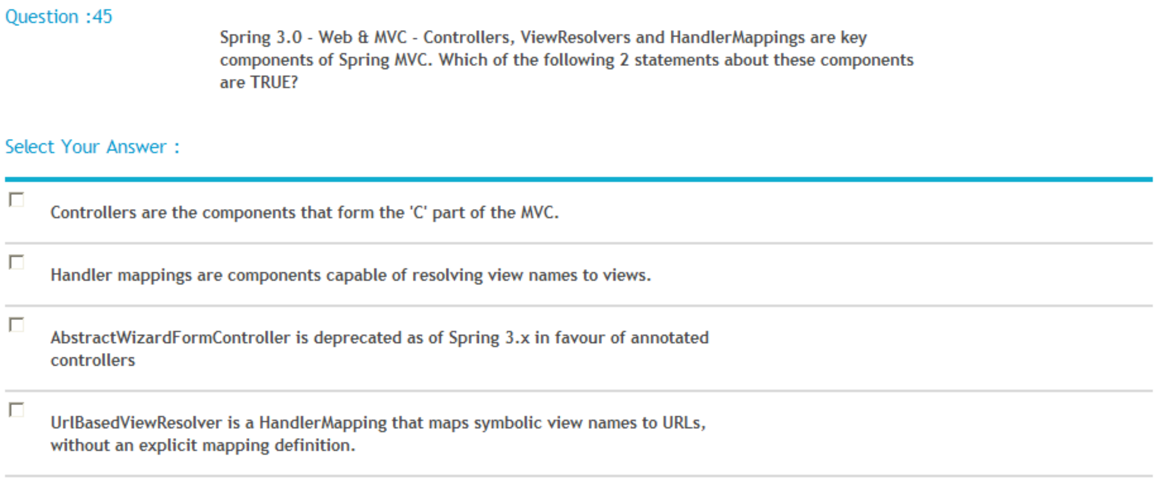
Select Your Answer :

○ An advice is a Spring bean

○ An advice instance can be shared across all advised objects, or unique to each advised object. This corresponds to per-class or per-instance advice.

○ Advices can be associated to both methods and fields

○ Spring provides several advice types out of the box, and is extensible to support arbitrary advice types.



Question :45

Spring 3.0 - Web & MVC - Controllers, ViewResolvers and HandlerMappings are key components of Spring MVC. Which of the following 2 statements about these components are TRUE?

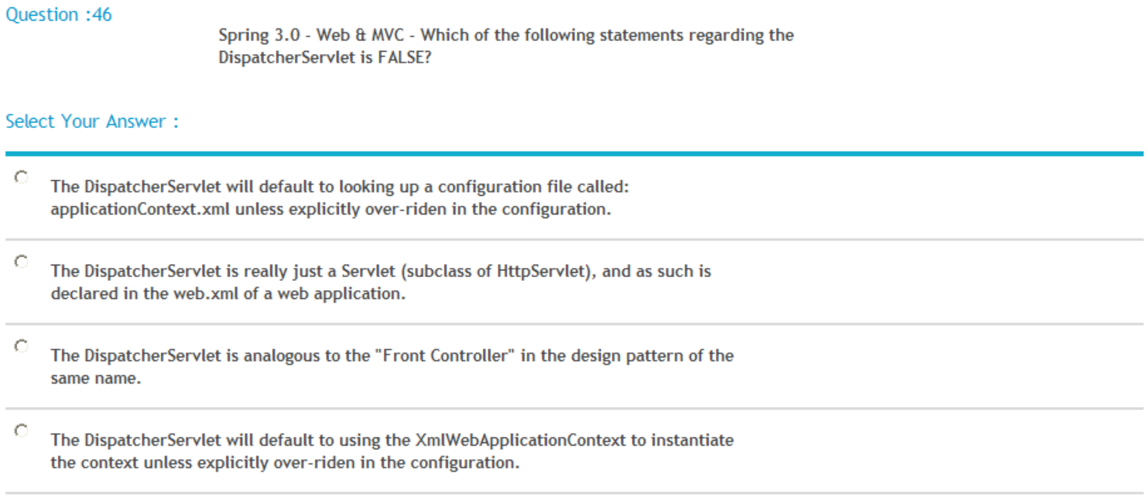
Select Your Answer :

□ Controllers are the components that form the 'C' part of the MVC.

□ Handler mappings are components capable of resolving view names to views.

□ AbstractWizardFormController is deprecated as of Spring 3.x in favour of annotated controllers

□ UrlBasedViewResolver is a HandlerMapping that maps symbolic view names to URLs, without an explicit mapping definition.



Question :46

Spring 3.0 - Web & MVC - Which of the following statements regarding the DispatcherServlet is FALSE?

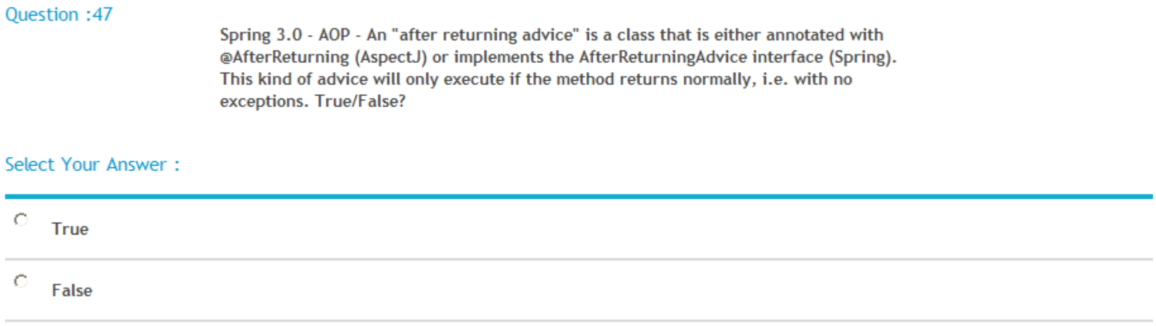
Select Your Answer :

○ The DispatcherServlet will default to looking up a configuration file called: applicationContext.xml unless explicitly over-riden in the configuration.

○ The DispatcherServlet is really just a Servlet (subclass of HttpServlet), and as such is declared in the web.xml of a web application.

○ The DispatcherServlet is analogous to the "Front Controller" in the design pattern of the same name.

○ The DispatcherServlet will default to using the XmlWebApplicationContext to instantiate the context unless explicitly over-riden in the configuration.



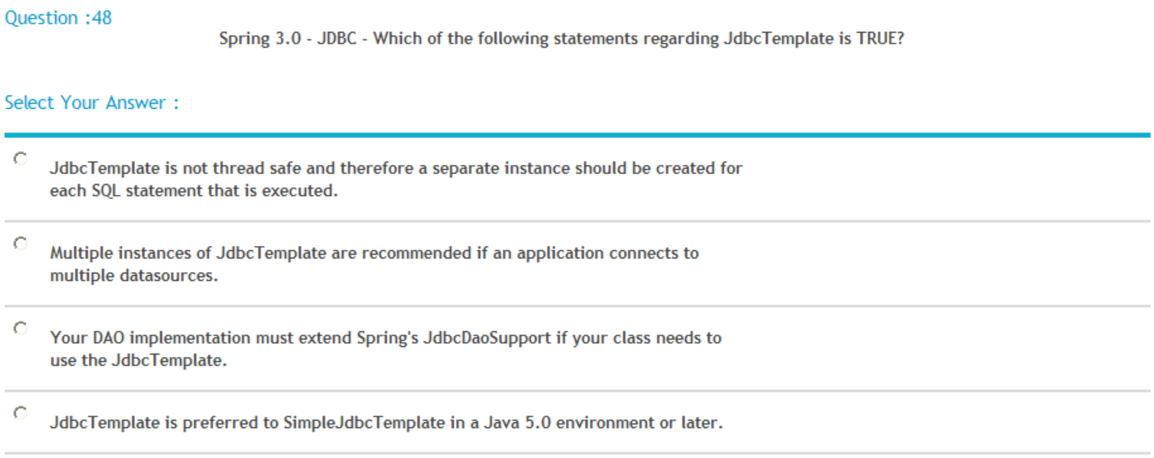
Question :47

Spring 3.0 - AOP - An "after returning advice" is a class that is either annotated with @AfterReturning (AspectJ) or implements the AfterReturningAdvice interface (Spring). This kind of advice will only execute if the method returns normally, i.e. with no exceptions. True/False?

Select Your Answer :

○ True

○ False



Question :48

Spring 3.0 - JDBC - Which of the following statements regarding JdbcTemplate is TRUE?

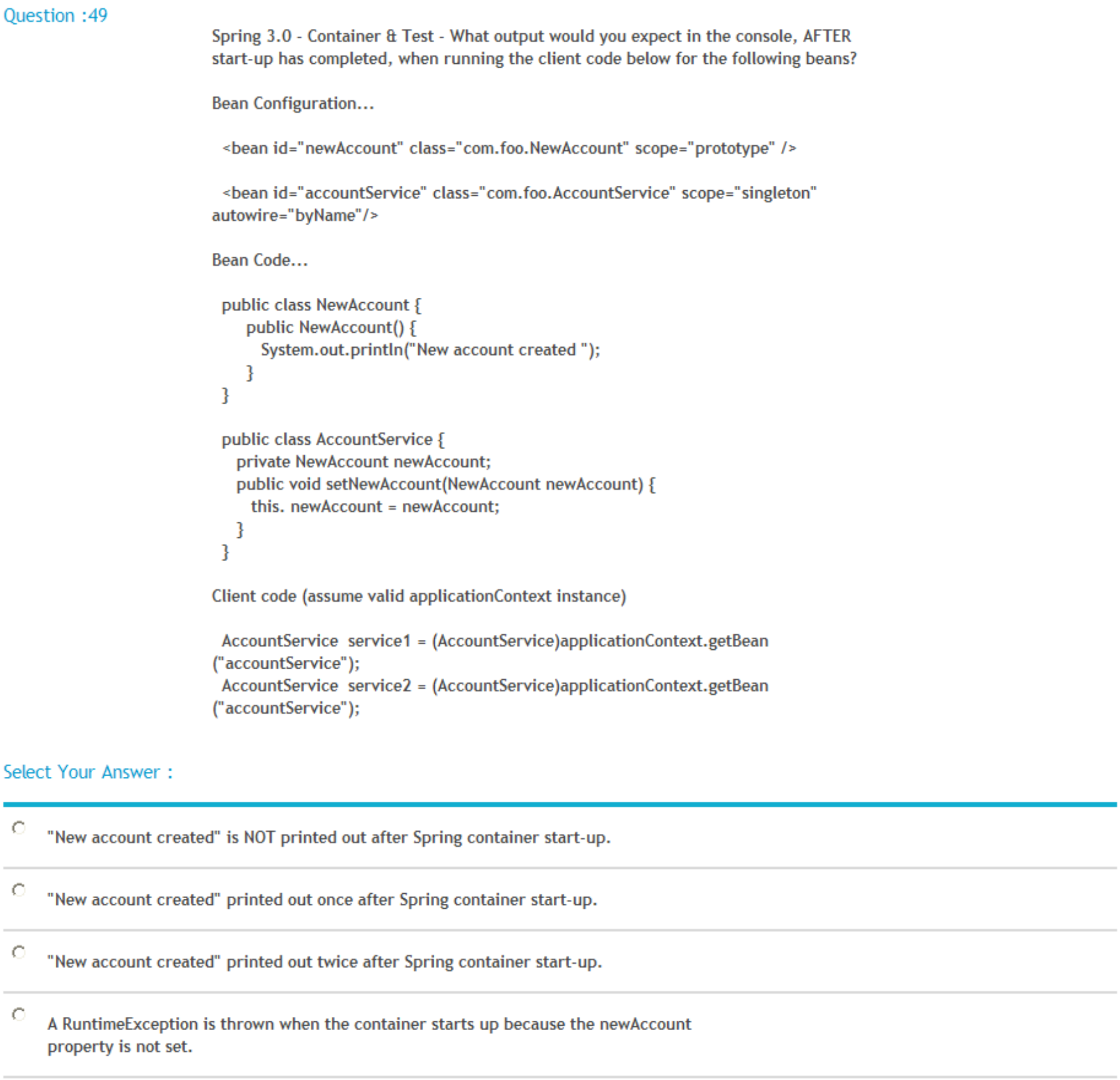
Select Your Answer :

○ JdbcTemplate is not thread safe and therefore a separate instance should be created for each SQL statement that is executed.

○ Multiple instances of JdbcTemplate are recommended if an application connects to multiple datasources.

○ Your DAO implementation must extend Spring's JdbcDaoSupport if your class needs to use the JdbcTemplate.

○ JdbcTemplate is preferred to SimpleJdbcTemplate in a Java 5.0 environment or later.



Question :49

Spring 3.0 - Container & Test - What output would you expect in the console, AFTER start-up has completed, when running the client code below for the following beans?

Bean Configuration...

```
 <bean id="newAccount" class="com.foo.NewAccount" scope="prototype" />

 <bean id="accountService" class="com.foo.AccountService" scope="singleton"
autowire="byName"/>
```

Bean Code...

```
 public class NewAccount {
     public NewAccount() {
       System.out.println("New account created ");
     }
 }

 public class AccountService {
   private NewAccount newAccount;
   public void setNewAccount(NewAccount newAccount) {
     this. newAccount = newAccount;
   }
 }
```

Client code (assume valid applicationContext instance)

```
 AccountService  service1 = (AccountService)applicationContext.getBean
("accountService");
 AccountService  service2 = (AccountService)applicationContext.getBean
("accountService");
```

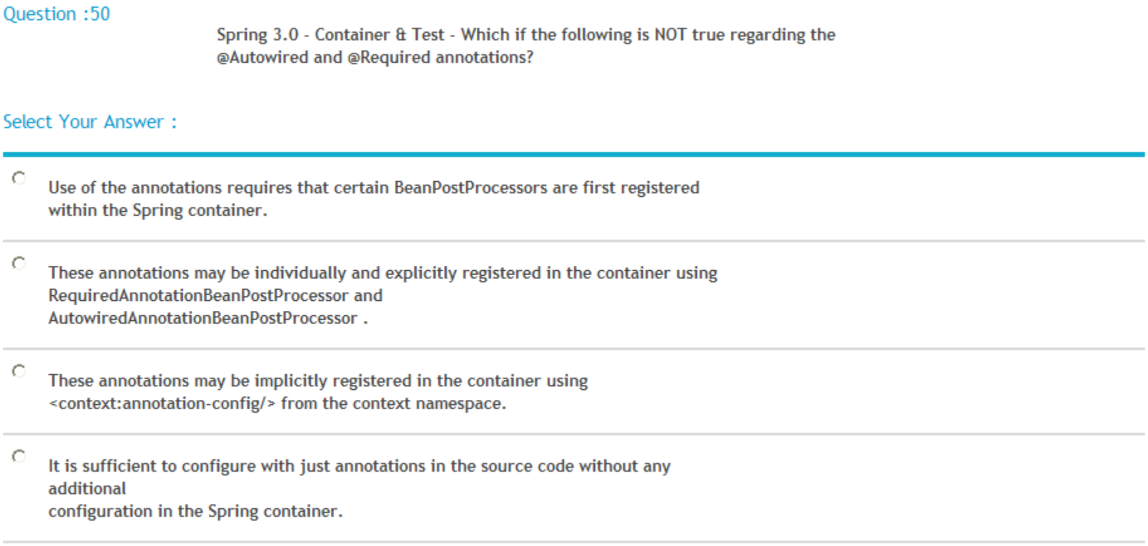
Select Your Answer :

○ "New account created" is NOT printed out after Spring container start-up.

○ "New account created" printed out once after Spring container start-up.

○ "New account created" printed out twice after Spring container start-up.

○ A RuntimeException is thrown when the container starts up because the newAccount property is not set.



Question :50

Spring 3.0 - Container & Test - Which if the following is NOT true regarding the @Autowired and @Required annotations?

Select Your Answer :

○ Use of the annotations requires that certain BeanPostProcessors are first registered within the Spring container.

○ These annotations may be individually and explicitly registered in the container using RequiredAnnotationBeanPostProcessor and AutowiredAnnotationBeanPostProcessor .

○ These annotations may be implicitly registered in the container using <context:annotation-config/> from the context namespace.

○ It is sufficient to configure with just annotations in the source code without any additional configuration in the Spring container.