


Core Spring 3.0 Certification Mock Test 2

Following on from the positive response to my previous mock I have created this 2nd one. As before all answers are fully referenced back to official Spring 3.0 documentation. All syllabus topics covered from the real exam: Container, Test, AOP, SpEL, Database, JMS, JMX, Web, MVC, Remoting, Dynamic Lang support etc.

Category	Certification / Mock test		
Keywords	Java, core spring 3.0, certification mock test		
Tags	spring		
Questions	10- Free questions 40 - Paid questions		
Test takers	834		
Average score	48.39		
Rating	 (4/5)		
Guru	ikoko		
Price	\$ 10.0 \$ 1.99	Add to Cart	
Validity	Forever 		
View questions of this test			
Start the Test		Go Back	

Question :1

Database

To use an embedded HSQL database in Spring, that runs a script called skillguru.sql, which steps could you use from the below?

Select Your Answer :

- ☐ Use the "embedded-database" tag in the spring-jdbc namespace: `<jdbc:embedded-database id="dataSource"> <jdbc:script location="classpath:skillguru.sql"/> </jdbc:embedded-database>`
- ☐ Use a programmatic approach as below, i.e. `EmbeddedDatabaseBuilder builder = new EmbeddedDatabaseBuilder(); EmbeddedDatabase db = builder.setType(H2).addScript("skillguru.sql").build(); // do stuff with the db (EmbeddedDatabase extends javax.sql.DataSource) db.shutdown()`
- ☐ Use the "embedded" tag in the spring-jdbc namespace: `<jdbc:embedded id="dataSource"> <jdbc:script location="classpath:skillguru.sql"/> </jdbc:embedded>`
- ☐ Use a programmatic approach as below, i.e. `EmbeddedDatabaseBuilder builder = new EmbeddedDatabaseBuilder(); EmbeddedDatabase db = builder.addScript("skillguru.sql").build(); // do stuff with the db (EmbeddedDatabase extends javax.sql.DataSource) db.shutdown()`

Question :2

Web MVC

The `@RequestMapping` annotation may be used to map URIs to Java code in Spring MVC.

Which of the following 2 statements regarding this annotation are CORRECT?

Select Your Answer :

- ☐ It may be applied only to a method and not a class.
- ☐ It may be applied to a method and a class
- ☐ You may define the http request method (GET or POST) as well as a cookie name to access this part of the http request.
- ☐ You may define the http request method (GET or POST) but NOT any cookie details.

Question :3

Type Conversion

Spring 3.0 provides the `PropertyEditor` for performing thread-safe type conversion. This allows any Type to be converted to any other Type.

Select Your Answer :

- ☐ TRUE
- ☐ FALSE

Question :4

Remoting

Spring 3.0 provides support for RESTful services.

Which of the following statements regarding these features of the framework are TRUE?

1. A `RestTemplate` is provided for accessing a RESTful service, which is analogous to a `JdbcTemplate`, `JmsTemplate` etc.
2. A `RestServiceExporter` is provided for exporting a RESTful service, which is analogous to a `HessianServiceExporter`, `RmiServiceExporter` etc.
3. `RestTemplate` provides support for ALL of the http methods: i.e. GET and POST as well as DELETE, HEAD, OPTIONS and PUT.
4. `RestTemplate` provides support for 2 of the 6 http methods: GET and POST only. You need to use a more tailored API, such as Jakarta Commons `HttpClient`, to gain full REST features.

Select Your Answer :

☐ 1, 2, 3

☐ 1, 2, 4

☐ 1, 3

☐ 2, 3

Question :5

AOP

Load-time weaving (LTW) is the process of weaving AspectJ aspects into an application's class files as they are being loaded into the Java virtual machine (JVM).

Which of the following are *MANDATORY* steps in integrating AspectJ LTW in to your application?

1. Add the jars: `spring-aop.jar`, `aspectjrt.jar` and `aspectjweaver.jar` in to your classpath.
2. Create you Aspect class/es.
3. Create a `META-INF/aop.xml` that declares your aspect class/es.
4. Add `<context:load-time-weaver/>` in to your application's XML.

Select Your Answer :

- ☐ 1, 2, 3
- ☐ 1, 3, 4
- ☐ 2, 3, 4
- ☐ 1, 2, 3, 4

Question :6

SpEL

The expression language supports the Ant S syntax for expressing values in your application context XML file/s or annotations?

Select Your Answer :

- ☐ TRUE
- ☐ FALSE

Question :7

Transactions

The `TransactionDefinition` interface specifies Isolation, Propagation, Timeout, and Read-only status?

Select Your Answer :

☐ TRUE

☐ FALSE

Question :8

AOP

If some target code (that is matched to a pointcut expression) throws an exception - which of the following advices would NOT execute?

Select Your Answer :

☐ An advice using the `@After` annotation (with a matching pointcut expression)

☐ An advice using the `@Before` annotation (with a matching pointcut expression)

☐ An advice using the `@AfterReturning` annotation (with a matching pointcut expression)

☐ An advice using the `@AfterThrowing` annotation (with a matching pointcut expression)

Question :9

Database

The following methods ...

- `queryForInt (args)`
- `queryForList (args)`
- `getJdbcOperations ()`
- `getNamedParameterJdbcOperations ()`

are ALL present on which of the following classes?

Select Your Answer :

-
- ☐ SimpleJdbcTemplate
-
- ☐ JdbcTemplate
-
- ☐ NamedParameterJdbcTemplate
-
- ☐ All of the above
-

Question :10

Resources - Which of the following statements regarding accessing low level resources using Spring are TRUE?

Select Your Answer :

☐ This API provides support for accessing files held on an FTP server.

☐ Spring builds on the `java.net.URL` API and its ability to access files held on a classpath, and to verify whether a resource exists or not.

☐ Spring's `Resource` interface contains the following methods also on `java.io.File`, i.e.

- `boolean exists()`
- `boolean delete()`
- `File getAbsoluteFile()`
- `boolean createNewFile()`

☐ The `Resource` API is used extensively in Spring and wraps existing functionality where possible. For example, a `UrlResource` wraps a `URL` which is where the underlying work is done.

Question :11

Container & Test

Lifecycle callbacks enable developers to execute code at certain points in a bean's lifecycle. However, there is a small downside in that you can only implement callbacks by coupling your application code to Spring.

Is this statement TRUE or FALSE?

Select Your Answer :

☐ TRUE

☐ FALSE

Question :12

Database -

Spring's `LocalSessionFactoryBean` supports the use of both traditional Hibernate mapping (hbm.xml) files AND `javax.persistence` (i.e. `@Entity`) annotated POJOs?

Select Your Answer :

☐ TRUE

☐ FALSE

Question :13

Container & Test

Spring 3.0 introduces more support for annotation based configuration. Consider the following configuration.

```
<bean id="springExamService"
class="com.skillguru.services.SpringExamServiceImpl"/>
```

What would be the equivalent annotated code?

Select Your Answer :

☐ `@BeanFactory`

```
public class AppConfig {
```

```
    @Bean public SpringExamService springExamService() {
```

```
        return new SpringExamServiceImpl();
```

```
    }
```

```
}
```


☐ @Configuration

```
public class AppConfig {  
  
    @Bean public SpringExamService springExamService() {  
  
        return new SpringExamServiceImpl();  
  
    }  
  
}
```

☐ @Configuration

```
public class AppConfig {  
  
    @ConfigurationBean public SpringExamService springExamService() {  
  
        return new SpringExamServiceImpl();  
  
    }  
  
}
```

☐ There is no annotated equivalent for that configuration.

Question :14

Container & Test

If a class has the following methods...

```
@PreDestroy public shutDown() {  
  
    System.out.println("shutDown ");  
  
}  
  
public closeMe() {  
  
    System.out.println("closeMe");  
  
}
```

...and the bean is configured with... `destroy-method="closeMe"`

Assuming the container is configured to process annotations correctly - what would be the expected order of the log output?

Select Your Answer :

-
- ☐ None - a RuntimeException would be thrown as it is not legal to define more than one lifecycle destroy mechanism on a single bean
-
- ☐ "closeMe" only. This is because if you define a destroy-method this takes precedence over any other destroying lifecycle mechanism.
-
- ☐ "shutDown" then "closeMe" are printed.
-
- ☐ "closeMe" then "shutDown" are printed.
-

Question :15

Remoting

Spring 3.0 provides support for exposing a bean as a JAX-WS service, via exporting it using the `SimpleJaxWsServiceExporter`.

This is a similar concept to using an `RmiServiceExporter` or a `HessianServiceExporter`.

Select Your Answer :

-
- ☐ TRUE
-
- ☐ FALSE
-

Question :16

SpEL

Which of the following 2 statements regarding SpEL are *true*?

Select Your Answer :

-
- ☐ The use of @Value("#{ systemProperties['user.region'] }") will correctly locate the user.region from the system properties.
-
- ☐ The use of @Value("#{ system['user.region'] }") will correctly locate the user.region from the system properties.
-
- ☐ The @Value annotation (with an embedded SpEL expression) may only be set on fields.
-
- ☐ The @Value annotation (with an embedded SpEL expression) may be set on to a field, setter method and on a parameter in a method that is annotated with @Autowired
-

Question :17

Transactions

Identify the correct statement below regarding annotation driven transactions.

Select Your Answer :

-
- ☐ The presence of the @Transactional annotation is enough to activate the transactional behavior.
-
- ☐ Spring recommends that you annotate interfaces with the @Transactional annotation, as opposed to annotating concrete classes (and methods of concrete classes).
-
- ☐ In the config tag ... <tx:annotation-driven/> ... the optional "mode" attribute defaults to "aspectj". This weaves the classes with Spring's AspectJ transaction aspect, modifying the target class byte code to apply to any kind of method call.
-
- ☐ In the config tag ... <tx:annotation-driven/> ... the "proxy-target-class" attribute allows two types of proxy mode and only applies when the "mode" is in "proxy" mode and not "aspectj".
-

Question :18

Database

Which of the following statements is TRUE regarding batch processing?

Select Your Answer :

-
- ☐ Both JdbcTemplate and SimpleJdbcTemplate offer this feature via their batchUpdate() methods.
-
- ☐ Spring advice to break out in to regular JDBC when batch processing as this feature is not supported in Spring.
-
- ☐ Only JdbcTemplate offers this feature via its batchUpdate() method.
-
- ☐ You can achieve JdbcTemplate batch processing by implementing two methods of the interface, BatchPreparedStatementCreator, and passing this in as a second parameter in the batchUpdate method call.
-

Question :19

Container & Test

Which of the following annotations is *NOT* component scanned by default?

Select Your Answer :

-
- ☐ @Repository
-
- ☐ @Service
-
- ☐ @Bean
-
- ☐ Your custom annotation that is itself annotated with @Component
-
- ☐ @Component
-

Question :20

Dynamic Language Support

The dynamic languages currently supported are:

Select Your Answer :

-
- ☐ Groovy
-
- ☐ JRuby
-
- ☐ BeanShell
-
- ☐ All of the above
-

Question :21

Web MVC

Which of the following statements is FALSE regarding multipart handling / file uploading in forms in Spring?

Select Your Answer :

-
- ☐ By default Spring does multi-part handling
-
- ☐ You enable Spring multipart handling by adding a multipart resolver to the web application's context.
-
- ☐ If you configure CommonsMultipartResolver, you need to add commons-fileupload.jar to your classpath.
-
- ☐ When the Spring DispatcherServlet detects a multi-part request, it activates the resolver that has been declared in your context and hands over the request.
-

Question :22

AOP -

What is wrong with the following code excerpt that uses PCDs (pointcut definitions)?

```
@Pointcut("execution(public * *(..))")

private void anyPublicOperation() {}

@Pointcut("within(com.ikoko..*)")

private void inIkoko() {}

@Pointcut("anyPublicOperation() && inIkoko() ")

private void examOperation() {}
```

Select Your Answer :

-
- ☐ In the final pointcut it is not legal to use the && operator within an expression.
-
- ☐ The 2nd pointcut has an erroneous two dots and should be a single dot
-
- ☐ In the final pointcut the expression refers to other pointcuts by name. This is not legal.
-
- ☐ There is nothing wrong.
-

Question :23

XML

Spring's OXM (Object XML mapping) APIs provide support for which of the following technologies?

1.

JAXB

2.

Castor

3.

XMLBeans

4.

JiBX

Select Your Answer :

☐ None of the above

☐ 1 only

☐ 1 & 2 only

☐ 1, 2 & 3 only

☐ All of the above

Question :24

Transactions

If a method is given a propagation type of NOT_SUPPORTED. What will happen in the event that the method is run within a transaction from the calling method?

Select Your Answer :

- ☐ It throws an exception.
- ☐ It supports the current transaction.
- ☐ It executes within a nested transaction
- ☐ It suspend the current transaction.

Question :25

Container & Test

In the following code excerpt...

```
// create and configure beans
```

```
ApplicationContext context = new ClassPathXmlApplicationContext(new  
String[] { "services.xml", "daos.xml" });
```

```
// retrieve configured instance
```

```
CustomerService service = (CustomerService )context.getBean  
("customerService");
```

How has this approach been improved in Spring 3.0?

Select 2 answers.

Select Your Answer :

- ☐ You no longer programmatically need to load an ApplicationContext - it is done purely through configuration.
- ☐ A new overloaded constructor on ClassPathXmlApplicationContext that uses varargs reduces code and enables you to construct an instance like so... `ApplicationContext context = new ClassPathXmlApplicationContext("services.xml", "daos.xml");`
- ☐ A new overloaded `getBean()` method that takes a type reduces code and enables you to retrieve your bean like so... `// retrieve configured instance CustomerService service = context.getBean("customerService", CustomerService.class);`
- ☐ `ClassPathXmlApplicationContext` and `FileSystemXmlApplicationContext` have both been deprecated and replaced with a single simplified class that can utilise both approaches to load a container.

Question :26

Container & Test

The `@Configuration` annotation is a new Java-centric way of configuring your Spring application. Which of the following statements regarding this annotation is *FALSE*?

Select Your Answer :

- ☐ Spring's `@Configuration` class support is intended to be a 100% complete replacement for Spring XML.
- ☐ You can "switch on" `@Configuration` annotations in your XML configuration with `<context:annotation-config/>`
- ☐ Because `@Configuration` is meta-annotated with `@Component`, annotated classes are automatically candidates for component scanning.
- ☐ `@Configuration` must be applied to a class only and not methods or fields.

Question :27

EJB

Study the four statements below...

1. Spring provides support for implementing EJBs
2. Spring provides support for accessing EJBs
3. Spring provides support for EJB 2.x and EJB 3.x
4. Spring simplifies JNDI lookups.

Which of these are true ?

Select Your Answer :

- ☐ 1, 2 & 3
- ☐ 1, 2 & 4
- ☐ 2, 3 & 4
- ☐ 1, 2, 3 & 4

Question :28

AOP

The `ProxyFactoryBean` chooses to create one of either a JDK or CGLIB-based proxy for a particular target object (that is to be proxied).

Which of the following is NOT a consideration when choosing between them?

Select Your Answer :

- ☐ Whether performance is an issue
- ☐ Whether the target classes, or their methods, are declared as final
- ☐ Whether the target classes implement interfaces
- ☐ Whether you can modify the source code of the target classes

Question :29

JMX

Assemblers configure your application on the rules needed to decide which components are exported as managed resources. Which of the following is NOT a valid assembler in Spring 3.0?

Select Your Answer :

- ☐ MetadataMBeanInfoAssembler
- ☐ MethodNameBasedMBeanInfoAssembler
- ☐ AttributeBasedMBeanInfoAssembler
- ☐ InterfaceBasedMBeanInfoAssembler

Question :30

Email

Which of the following steps is INVALID regarding the integration of Spring's email functionality in to your application?

Select Your Answer :

- ☐ Add mail.jar (JavaMail) and activation.jar (JAF) in to your classpath.
- ☐ Define a `JavaMailSenderImpl` bean that takes the SMTP settings (host, authentication etc.)
- ☐ Define a `MailTemplate` bean which will be the API that simplifies usage to the underlying JavaMail API.
- ☐ Use the `SimpleMailMessage` API to send a basic email.

Question :31

JMX

Which core class in Spring's JMX framework is responsible for taking Spring beans and registering them with a JMX MBeanServer?

Select Your Answer :

- ☐ JMXExporter
- ☐ MBeanExporter
- ☐ ManagedResource
- ☐ MBeanServerFactoryBean

Question :32

Web MVC

Which of the following two options would be best practises if you require your controller to issue a redirect to the browser?

Select Your Answer :

- ☐ Just simply utilise the HttpServletResponse.sendRedirect() call.
- ☐ Create and return an instance of Spring's RedirectView.
- ☐ Return a view name that is prefixed with "redirect:"
- ☐ Simply call super.redirect() on the Spring BaseController that you extend

Question :33

AOP

With the following pointcut expression in mind...

```
execution(* set*(..))
```

Which of the following methods would be matched?

Select Your Answer :

- ☐ private void setFoo(String bar)
- ☐ public void setBar()
- ☐ public String setFoo(int bar)
- ☐ public void set(String foo, int bar)
- ☐ All of the above

Question :34

Container & Test

`ReflectionTestUtils` is a useful Spring API for usage in unit and integration test methods.

What feature does this API provide?

Select Your Answer :

- ☐ To be able to set a non-public field or invoke a non-public setter method when testing code.
- ☐ A collection of assertions intended to simplify testing scenarios dealing with Spring Web MVC `ModelAndView` objects.
- ☐ Provides support for dependency injection and initialization of test instances.
- ☐ Provides functionality of the Spring `TestContext` Framework to standard JUnit 4.5+ tests by means of the `TestContextManager` and associated support classes and annotations.

Question :35

General

Which of the following is NOT a new feature introduced in Spring 3.0?

Select Your Answer :

- ☐ Spring Expression Language
- ☐ `SimpleJdbcTemplate` updated to support Java 5.0 features
- ☐ Early support for Java EE6
- ☐ REST supported enhanced
- ☐ Embedded database support

Question :36

Container & Test

MockTest has an initialize() method, but no init() method.

SpringMockTest does *not* have an init() method.

...and they are configured like so...

```
<beans default-init-method="init">

    <bean id="mockTest" class="com.foo.MockTest" init-
method="initialize" />

    <bean id="springMockTest" class="com.foo.SpringMockTest" />

</beans>
```

What would you expect to happen at runtime?

Select Your Answer :

- ☐ An exception is thrown because the beans definition requires an init() method which is absent from both MockTest and SpringMockTest.
- ☐ An exception is thrown because the MockTest bean defines its own init-method which is not legal when also defining a top-level "default-init-method"
- ☐ An exception is thrown because the "springMockTest" does not have an init() method and this is required by the top-level "default-init-method" beans definition. "mockTest" is okay because it provides the container with an alternative lifecycle mechanism via its "init-method" attribute
- ☐ None of the above.

Question :37

Database

Spring's exception translation mechanism provides a consistent and runtime exception hierarchy to the client, abstracted away from the underlying data-access technology.

Select Your Answer :

- ☐ TRUE
- ☐ FALSE

Question :38

JMX

What is wrong with the following annotated code?

```
public class Customer {

    @ManagedResource

    private String name;

    @ManagedAttribute

    public void setName(String name) {

        this.name = name;

    }

    public String getName() {

        return name;

    }

    @ManagedOperation

    public String getNameFormatted() {

        return name.toUpperCase();

    }

}
```

Select Your Answer :

-
- ☐ @ManagedResource should be applied to a class only
-
- ☐ @ManagedAttribute should be on a property / field only not on a setter.
-
- ☐ @ManagedOperation should not be applied to this method as it is not a JavaBean style getter or setter.
-
- ☐ None of these - the annotated bean is correct
-

Question :39

Web MVC -

In Spring's form tag library which of the following tags are valid?

1. input and inputs
2. checkbox and checkboxes
3. radiobutton and radiobuttons
4. redirect

Select Your Answer :

-
- ☐ 1, 2 & 3 only
-
- ☐ 2 & 3 only
-
- ☐ 1 & 4 only
-
- ☐ All of the above
-

Question :40

AOP

Which of the following 2 statements are true regarding the decisions you might need to make regarding the use of aspect oriented programming?

Select Your Answer :

-
- ☐ Business logic that contains transactional code should be considered as a candidate for using aspects.
-
- ☐ AspectJ is simpler than using Spring AOP.
-
- ☐ If you have chosen to use Spring AOP, then you have a choice of @AspectJ or XML style.
-
- ☐ @AspectJ has two disadvantages: it does not encapsulate the implementation of the requirement in a single place. Secondly, @AspectJ is slightly more limited in what it can express than the XML style.
-

Question :41

Container & Test

The default bean scope in Spring is 'singleton' ?

Select Your Answer :

-
- ☐ TRUE
-
- ☐ FALSE
-

Question :42

Container & Test

Which of the 2 following are suitable approaches for unit testing using Spring.

Select Your Answer :

-
- ☐ Use ModelAndViewAssert to unit test your Spring MVC controllers.
-
- ☐ Use the @ContextConfiguration annotation to define your application context files.
-
- ☐ Make your test class extend AbstractDependencyInjectionSpringContextTests
-
- ☐ You can use numerous classes from the org.springframework.mock.web package or even dynamic mock objects such as EasyMock or MockObjects, to provide mocked collaborators to the classes you wish to test.
-

Question :43

JMS

Which of the following statement is TRUE regarding the JmsTemplate?

Select Your Answer :

-
- ☐ It should be configured using "prototype" scope to ensure one instance is created every time it is used to avoid thread issues.
-
- ☐ The JmsTemplate requires a reference to a ConnectionFactory.
-
- ☐ JmsTemplate supports the receiving of asynchronous message.
-
- ☐ JmsTemplate fully supports conversion between Java objects and JMS messages.
-

Question :44

Container & Test

Which of the following statements regarding container related annotations are *TRUE*?

Select Your Answer :

-
- ☐ @Inject has been largely superseded by the Java language @Autowired annotation
-
- ☐ @Autowired marks a method (typically a JavaBean setter method) as being 'required': that is, the setter method must be configured to be dependency-injected with a value.
-
- ☐ As of Spring 3.0 it is no longer necessary to define a BeanPostProcessor in your XML configuration to "switch on" annotations such as @Required and @Autowired.
-
- ☐ None of the above
-

Question :45

Container & Test

JSR 330's @Inject annotation can be used in place of Spring's @Autowired annotation in which scenarios below?

Select Your Answer :

-
- ☐ When annotating a field
-
- ☐ When annotating a JavaBean style setter method.
-
- ☐ When annotating a regular public method with multiple args.
-
- ☐ All of the above.
-

Question :46

Remoting

You have been tasked with exposing your hibernate application as a remote service. You will also be creating the client. Just as when you thought your lucky streak might run out - you have been blessed with a co-operative networks administrator who is willing to open up any ports that you need.

From the following technologies below which would you consider as a viable option?

1. Spring's HTTP invoker
2. RMI
3. Hessian
4. Burlap

Select Your Answer :

-
- ☐ 1 & 2
-
- ☐ 3 & 4
-
- ☐ All four are viable technologies to use
-
- ☐ None of the above are suitable.
-

Question :47

Container & Test

Consider the following two configuration fragments whereby the property 'answer' is of type `java.lang.String`:

```
<bean class="skillGuruBean1">

    <property name="answer" value=""/>

</bean>

<bean class="skillGuruBean2">

    <property name="answer"><null/></property>

</bean>
```

Consider the following statements regarding the possible values of 'answer' in each bean...

1. `skillGuruBean1.answer` is a blank String or ""
2. `skillGuruBean1.answer` is null
3. `skillGuruBean2.answer` is a blank String or ""
4. `skillGuruBean2.answer` is null
5. `skillGuruBean2` throws a `RuntimeException` upon initialization as there is no such tag as `<null />`

Which of the above statements are CORRECT?

Select Your Answer :

☐ 1 & 3

☐ 1 & 4

☐ 1 & 5

☐ 2 & 4

☐ 2 & 5

Question :48

Database

The framework reference documentation recommends that `JdbcTemplate` is scoped as a "prototype" bean. This prevents thread concurrency issues when reading / writing to the database.

Select Your Answer :

-
- ☐ TRUE
-
- ☐ FALSE
-

Question :49

JCA

A CCI, or Common Client Interface, is an interface that an application can use to interact with an EIS (Enterprise Information System).

Consider the following statements regarding Spring's support for JCA...

1. The base resource to use JCA CCI is the `ConnectionFactory` interface.
2. You may directly instantiate a `ConnectionFactory`.
3. The `CciTemplate` simplifies the use of CCI since it handles the creation and release of resources.
4. It is not possible to use global transactions when a connector is used in non-managed mode.

Which of these statements are true?

Select Your Answer :

-
- ☐ 1, 2, 3
-
- ☐ 1, 2, 4
-
- ☐ 1, 3, 4
-
- ☐ All four statements are true
-

Question :50

Web MVC

Consider the following web.xml fragment...

```
<servlet>

    <servlet-name>ikoko</servlet-name>

    <servlet-class> <THE CONTROLLER SERVLET> </servlet-class>

    <load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

    <servlet-name>ikoko</servlet-name>

    <url-pattern>/skillguru/*</url-pattern>

</servlet-mapping>
```

In order to complete configuration of this Spring MVC application - what should be used in place of <THE CONTROLLER SERVLET> ?

And what is the name of the Spring XML file that will, by default, be searched for?

Select Your Answer :

- ☐ org.springframework.web.servlet.DispatcherServlet and applicationContext.xml
- ☐ org.springframework.web.servlet.DispatcherServlet and ikoko-servlet.xml
- ☐ Your custom servlet and applicationContext.xml
- ☐ Your custom servlet and ikoko-servlet.xml