Test results by question

Test:	**JAVA TEST BANK 2 (ID# 22c845aad6e450a3)**
Candidate:	**AZEEM MAREDIYA**
Score	**0%**

Question 1 (ID: # 47319)

Subject	Java Enterprise Edition 5
Subtopic	JavaServer Pages
Description	Page Initialization

Which of the following methods can be used to initialize a JSP page?

A. __jspInit()
B. initialize()
C. init()
D. initjsp()
E. jspInit()

Correct Answer	E
User Answer	
Elapsed Time (seconds)	1
Explanation	The jspInit() method can be used to initialize a JSP page.

Question 2 (ID: # 47322)

Subject	Java Enterprise Edition 5
Subtopic	JavaServer Pages
Description	Handling Errors

What will happen if you include the `page` directive below at the beginning of your JSP page?

```
<%@ page errorPage="FileName" %>
```

A. errorPage.jsp will serve as an error page.
B. You will access the error data object in an EL expression via the errorPage .
C. The web container will forward control to the error page if an exception happens.
D. Accessing the FileName will cause an errorPage to appear.
E. The WAR page's error page will take precedence.

Correct Answer	C
User Answer	
Elapsed Time (seconds)	0

Explanation Any number of exceptions can arise when a JSP page is executed. The page directive shown above is used to state that the web container should forward control to an error page if an exception takes place.

Question 3 (ID: # 47323)

Subject	Java Enterprise Edition 5
Subtopic	JavaServer Pages
Description	Conditional Tags

Which of the following are conditional tags?

A. if

B. foreach

C. choose

D. out

E. parse

Correct Answer A ;C ;

User Answer

Elapsed Time (seconds) 0

Explanation The choose tag and the if tag perform conditional block execution. In addition, the forEach tag permits you to iterate over a collection of objects.

Question 4 (ID: # 47324)

Subject	Java Enterprise Edition 5
Subtopic	JavaServer Pages
Description	Jsp:text Element

What will the output be when the following code is run?

```
<c:forEach var="counter" begin="1" end="${3}">
<jsp:text>${counter}</jsp:text>
</c:forEach>
```

A. 13

B. 321

C. 123

D. The code will not compile because the <jsp:text> expression should be used only for preserving all whitespace.

E. The code will not compile because the ${counter} expression is illegal in a JSP document.

Correct Answer C

User Answer

Elapsed Time (seconds) 0

Explanation The jsp:text element is used to output static data that is not well formed. The ${counter} expression in the example above would be illegal in a JSP document if it were not wrapped in a jsp:text tag. Therefore, the code is correct and the output will be 123. Also, if you use jsp:text , all whitespace is preserved.

Question 5 (ID: # 47327)	
Subject	Java Enterprise Edition 5
Subtopic	JavaServer Pages
Description	XML Syntax

A JSP document is an XML document, and therefore must comply with the XML standard. How is the `<%!  ..%>` standard JSP syntax converted to XML syntax?

A. <jsp:scriptlet> .. </jsp:scriptlet>

B. <jsp:expression> .. </jsp:expression>

C. <jsp:directive.page> .. </jsp:directive.page>

D. <jsp:declaration> .. </jsp:declaration>

E. <jsp:taglib> .. </jsp:taglib>

Correct Answer D

User Answer

Elapsed Time (seconds) 0

Explanation The standard JSP syntax corresponds to the .. XML syntax.

Question 6 (ID: # 47328)	
Subject	Java Enterprise Edition 5
Subtopic	JavaServer Pages
Description	Including an Applet

The `jsp:plugin` element is used when ______________________________.

A. a web component invokes another web resource

B. you want to include an applet in a JSP page

C. you want to access a servlet

D. you want to include HTML text in a tag body

E. you want to include a JavaBeans component in a JSP page

Correct Answer B ;E ;

User Answer

Elapsed Time (seconds) 0

Explanation The element plays or displays an object (usually an applet or a bean) in the client web browser. It uses a Java plug-in that is built into the browser or downloaded from a stated URL.

Question 7 (ID: # 47329)	
Subject	Java Enterprise Edition 5
Subtopic	JavaServer Pages
Description	Transferring Control

Which of the following mechanisms can a JSP page use to transfer control to another Web component?

A. the <jsp:redirect ..> element
B. the <%@ forward ..%> directive
C. the <jsp:forward ..> element
D. the <%@ redirect ..%> directive
E. There is no way of transferring the control. Client-side scripting must be used for that.

Correct Answer	C
User Answer	
Elapsed Time (seconds)	0
Explanation	The mechanism for transferring control to a different Web component from a JSP page uses the functionality offered by the Java Servlet API. You can access this functionality from a JSP page with the jsp:forward element (for instance).

Question 8 (ID: # 47332)

Subject	Java Enterprise Edition 5
Subtopic	JavaServer Pages
Description	IsThreadSafe

When the `isThreadSafe` attribute is set to "true," requests are dispatched one at a time, in the order they were received.

Correct Answer	FALSE
User Answer	
Elapsed Time (seconds)	0
Explanation	When the isThreadSafe attribute is set to "true," the web container can choose to dispatch multiple concurrent client requests to the JSP page.

Question 9 (ID: # 47333)

Subject	Java Enterprise Edition 5
Subtopic	JavaServer Pages
Description	ContentType Attribute

The `contentType` attribute can be used to ________________________________.

A. specify the encoding of the response
B. use a scripting language other than the default
C. specify the encoding of the sent message
D. specify the format of the static content in a JSP page
E. ensure that a method is always part of a transaction

Correct Answer	A ;D ;
User Answer	
Elapsed Time (seconds)	0

Explanation Options A and D are correct. For example, the date application states that the page has to be encoded using UTF -8, an encoding that supports almost all locales employing the following page directive:
. If you wanted a page to enclose data expressed in the WML, you would incorporate the following directive:
.

Question 10 (ID: # 47334)

Subject	Java Enterprise Edition 5
Subtopic	JavaServer Pages
Description	Static Content

What is the default format for the static content in a JSP page?

A. SVG
B. WML
C. XML
D. XHTML
E. HTML

Correct Answer C ;E ;

User Answer

Elapsed Time (seconds) 0

Explanation Static content may be expressed in any text-based format, for example HTML, Wireless Markup Language (WML), or XML. The default value for the MIMETYPE of the created response is the text/html for the JSP page in standard syntax, and the text/xml for JSP documents in XML syntax.

Question 11 (ID: # 47337)

Subject	Java Enterprise Edition 5
Subtopic	JavaServer Pages
Description	Life Cycle

When a request is mapped to a JSP page, the web container first checks whether the JSP page's servlet is older than the JSP page. What happens if the servlet is older?

A. The web container initializes the servlet instance by calling the jspInit method.
B. Scripting elements are inserted into the JSP page's servlet class.
C. The web container instantiates an instance of the servlet class.
D. The web container translates the JSP page into a servlet class and compiles the class.
E. The request cannot proceed, because having an older servlet is not allowed.

Correct Answer D

User Answer

Elapsed Time (seconds) 0

Explanation A JSP page services requests as a servlet. Thus, the life cycle and many of the capacities of JSP pages (especially the dynamic aspects) are determined by the Java Servlet technology. If the servlet is older, the web container translates the JSP page into a servlet class and compiles the class.

Question 12 (ID: # 47338)

Subject	Java Enterprise Edition 5
Subtopic	JavaServer Pages
Description	Tag Library Directives

What does the `<%@taglib ... %>` JSP construct do?

A. It sets the content type returned by the tag library.

B. It opens the custom tag libraries.

C. It imports the custom tag libraries.

D. It sets the custom tag attribute values.

E. It modifies the tag libraries.

Correct Answer	C
User Answer	
Elapsed Time (seconds)	0
Explanation	Tag library directives ( ) import custom tag libraries.

Question 13 (ID: # 47339)

Subject	Java Enterprise Edition 5
Subtopic	JavaServer Pages
Description	JSP Technology

The JSP technology allows you to easily create web content that has ________________________.

A. only static components

B. only dynamic components

C. only text-based components

D. both static and dynamic components

E. both static and text-based components

Correct Answer	D
User Answer	
Elapsed Time (seconds)	0
Explanation	The JSP technology makes all of the dynamic capacities of the Java Servlet technology available, but it offers a more natural method of creating static content.

Question 14 (ID: # 47347)

Subject	Java Enterprise Edition 5
Subtopic	Servlets
Description	Shutdown

To provide a clean shutdown, which of the following options should be inserted in the servlet code below?

```
public void destroy() {
if (numServices() > 0) {
/*insert code here*/
}
while(numServices() > 0) {
try {
Thread.sleep(interval);
} catch (InterruptedException e) {
} } }
```

A. leavingServiceMethod();

B. return shuttingDown;

C. setShuttingDown(true);

D. shuttingDownCommit;

E. isShuttingDown()

Correct Answer	C
User Answer	
Elapsed Time (seconds)	0
Explanation	The code shown offers a clean shutdown using the destroy method if setShuttingDown(true); is added. The destroy method will then wait for the service methods to complete, in order to provide a clean shutdown.

Question 15 (ID: # 47348)	
Subject	Java Enterprise Edition 5
Subtopic	Servlets
Description	Service Methods

All of a servlet's service methods should be complete when a servlet is removed. The server tries to ensure this by ______________________.

A. checking the service counter

B. calling the delete method only after all service requests have returned

C. accessing the session via service methods

D. counting the number of service methods that are still running

E. calling the destroy method only after all service requests have returned, or after a server-specific grace period

Correct Answer	E
User Answer	
Elapsed Time (seconds)	0
Explanation	The server tries to ensure this by calling the destroy method only after all service requests have returned, or after a server-specific grace period.

Question 16 (ID: # 47349)	
Subject	Java Enterprise Edition 5

Subtopic	Servlets
Description	Incomplete Initialization

If a servlet cannot complete its initialization process, which of the following exceptions should it throw?

A. ServletException

B. UnexpectedException

C. UnavailableException

D. ServerException

E. RuntimeException

Correct Answer C

User Answer

Elapsed Time (seconds) 0

Explanation A servlet that cannot complete its initialization process should throw an UnavailableException .

Question 17 (ID: # 47353)

Subject	Java Enterprise Edition 5
Subtopic	Servlets
Description	Monitoring Events

You can monitor and react to events in a servlet's life cycle by ______________________________.

A. periodically accessing the session via service methods

B. defining listener objects whose methods get invoked when life-cycle events occur

C. retrieving an output stream from the response

D. calling the getSession method of a request object

E. invoking the forward method of a RequestDispatcher

Correct Answer B

User Answer

Elapsed Time (seconds) 0

Explanation To react to events in a servlet's life cycle, you must define Listener objects. To employ these Listener objects, you must define and state the Listener class.

Question 18 (ID: # 47354)

Subject	Java Enterprise Edition 5
Subtopic	Servlets
Description	Removing a Servlet

If a container needs to remove a servlet, it finalizes the servlet by calling the servlet's `delete` method.

Correct Answer FALSE

User Answer

Elapsed Time (seconds) 0

Explanation When a servlet container decides that a servlet must be removed from service (for instance, when a container desires to retrieve memory resources or when it is being shut down), the container calls the destroy method of the Servlet interface.

Question 19 (ID: # 47357)	
Subject	Java Enterprise Edition 5
Subtopic	Servlets
Description	Life Cycle

The life cycle of a servlet is controlled by the ____________ in which the servlet has been deployed.

A. deployment descriptor
B. container
C. instance
D. listener
E. session

Correct Answer B

User Answer

Elapsed Time (seconds) 0

Explanation The life cycle of a servlet is controlled by the container in which the servlet has been deployed.

Question 20 (ID: # 47358)	
Subject	Java Enterprise Edition 5
Subtopic	Servlets
Description	Servlets

A servlet is ____________.

A. a class
B. a package
C. a web page
D. a method
E. a small server

Correct Answer A

User Answer

Elapsed Time (seconds) 0

Explanation A servlet is a Java programming language class that is used to extend the capabilities of servers that host application access via a request-response programming model.

Question 21 (ID: # 47352)

Subject	Java Enterprise Edition 5
Subtopic	Servlets
Description	Initializing a Servlet

A Web container initializes a servlet _________________________.

A.	using the [get|set]Attribute methods
B.	after the Web container loads and instantiates the Servlet class
C.	after it creates an instance of the Servlet class
D.	before it delivers requests from clients
E.	after extracting information from the request

Correct Answer B ;D ;

User Answer

Elapsed Time (seconds) 0

Explanation A Web container initializes the servlet after the Web container loads and instantiates the Servlet class, and before it delivers requests from clients.

Question 22 (ID: # 47343)

Subject	Java Enterprise Edition 5
Subtopic	Servlets
Description	Filters

Filters differ from Web components in that filters usually do not create a response themselves.

Correct Answer TRUE

User Answer

Elapsed Time (seconds) 0

Explanation The statement is true. Filters normally do not generate responses themselves. However, as a substitute, they offer universal functions that may be attached to any type of servlet or JSP page.

Question 23 (ID: # 47344)

Subject	Java Enterprise Edition 5
Subtopic	Servlets
Description	HTTP Session

Which of the following methods will allow you to obtain the current HTTP session, given a `HttpServletRequest` instance-named request?

A.	request.getSession(HttpServlet.HTTP_SESSION)
B.	request.getCurrentHTTPSession()
C.	request.getSession()
D.	request.getCurrentSession()
E.	request.getHTTPSession()

Correct Answer C

User Answer

Elapsed Time (seconds) 0

Explanation Option C is correct. The rest of the methods don't exist.

Question 24 (ID: # 47342)

Subject Java Enterprise Edition 5

Subtopic Servlets

Description Session Management

The timeout period associated to each session can be set by ________________________________.

A. accessing a session's time-to-live counter

B. setting an integer value in the session-timeout element in the deployment descriptor

C. using a session's [get|set]SetInterval methods

D. using a session's [get|set]MaxInactiveInterval methods

E. using a session's [get|set]MinInactiveInterval methods

Correct Answer B ;D ;

User Answer

Elapsed Time (seconds) 0

Explanation You can set the timeout period by setting an integer value in the session-timeout element, which is a child element of a session-config element. The integer value represents the number of minutes of inactivity that must pass before the session times out. It can also be set by using a session's [get|set]MaxInactiveInterval methods.

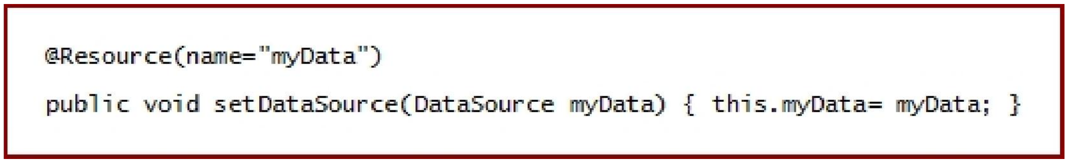
Question 25 (ID: # 162618)

Subject Java Frameworks: Spring 2.5

Subtopic Dependency Injection

Description @Resource

What is the role of the `@Resource` annotation in the example shown below?



```
@Resource(name="myData")
public void setDataSource(DataSource myData) { this.myData= myData; }
```

A. It creates a bean that will be used as a resource for the application.

B. It indicates which bean should be overridden.

C. It indicates an injection point.

D. It explicitly defines the target bean name.

E. It points to multiple beans with matching names.

Correct Answer D

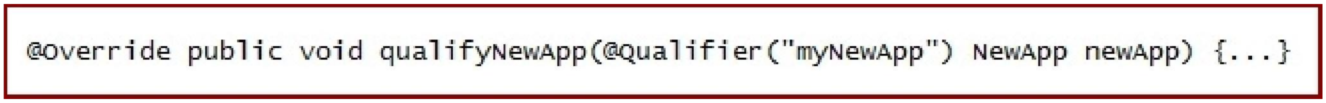
User Answer

Elapsed Time (seconds)	0
Explanation	Option D is correct. The @Resource annotation explicitly defines the target bean name.

Question 26 (ID: # 162619)

Subject	Java Frameworks: Spring 2.5
Subtopic	Dependency Injection
Description	@Qualifier Example

What does the `@Qualifier` annotation do in the example shown below?



```
@Override public void qualifyNewApp(@Qualifier("myNewApp") NewApp newApp) {...}
```

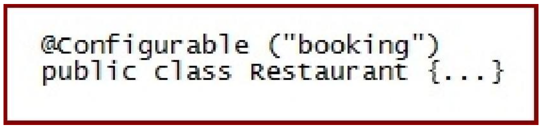
A. It indicates which object must be overridden.
B. It cancels the @Override annotation.
C. It points out a specific target bean.
D. It qualifies a simple object as a dependency object.
E. It does nothing, but it is required because of the @Override annotation.

Correct Answer	C
User Answer	
Elapsed Time (seconds)	0
Explanation	Option C is correct. The @Qualifier annotation from the example above points out a specific target bean.

Question 27 (ID: # 162622)

Subject	Java Frameworks: Spring 2.5
Subtopic	Dependency Injection
Description	@Configurable

The first line in the code snippet below __________.



```
@Configurable ("booking")
public class Restaurant {...}
```

A. specifies the name of the bean definition that serves as a configuration template
B. creates an object and makes it configurable for a database query
C. injects a configurable object
D. allows for a transactional configuration for the booking object
E. creates an object outside of the control

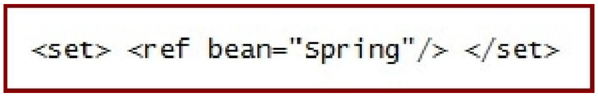
Correct Answer	A
User Answer	
Elapsed Time (seconds)	0

Explanation Option A is correct. @Configurable ("booking") specifies the name of the bean definition that serves as a configuration template.

Question 28 (ID: # 162623)

Subject	Java Frameworks: Spring 2.5
Subtopic	Dependency Injection
Description	Ref Tag

What does the `ref` tag do in the example shown below?



```
<set> <ref bean="Spring"/> </set>
```

A. It allows the container to validate the named bean.
B. It creates a reference to the bean in the container.
C. It refers to another bean that is configured in the same XML file.
D. It wraps the bean in a parent container.
E. It initializes the bean in the XML file.

Correct Answer C

User Answer

Elapsed Time (seconds) 0

Explanation Option C is correct. The ref tag refers to another bean that is configured in the same XML file.

Question 29 (ID: # 162624)

Subject	Java Frameworks: Spring 2.5
Subtopic	Dependency Injection
Description	Auto Wiring

Which of the following are `AUTO WIRING` types?

A. byName
B. byType
C. bySize
D. constructor
E. no (by default)

Correct Answer A ;B ;D ;E ;

User Answer

Elapsed Time (seconds) 0

Explanation Options A, B, D, and E are correct. The AUTO WIRING types are listed below.
- NO (the default)
- byName (checks the property name)
- byType (checks the property type)
- constructor

Question 30 (ID: # 162627)

Subject	Java Frameworks: Spring 2.5
Subtopic	Dependency Injection
Description	Configuration

You can configure the detection of Spring beans by the container, using __________.

A.	XML
B.	@Service
C.	@Pointcut
D.	@Advisor
E.	@Component

Correct Answer	A ;B ;E ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Options A, B, and E are correct. You can configure your Spring beans by the container via annotations (i.e. @ Service , @Component), or via XML.

Question 31 (ID: # 162628)

Subject	Java Frameworks: Spring 2.5
Subtopic	Dependency Injection
Description	Bean Definition

You can define a bean without using annotations.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	This statement is true. Beans can also be defined using XML configuration files.

Question 32 (ID: # 162629)

Subject	Java Frameworks: Spring 2.5
Subtopic	Dependency Injection
Description	meta-inf/manifest.mf

A web module (web deployment unit) can have a manifest. `META-INF/MANIFEST.MF` specifies the __________.

A.	Spring Bean dependencies
B.	transactions
C.	jar dependecies
D.	security

E.	caching

Correct Answer A ;C ;

User Answer

Elapsed Time (seconds) 0

Explanation Options A and C are correct. The meta-inf/manifest.mf specifies both Spring Bean dependencies and jar dependecies.

Question 33 (ID: # 162632)	
Subject	Java Frameworks: Spring 2.5
Subtopic	Dependency Injection
Description	Bean Tags: Class Attribute

In the space below, type in the attribute of the bean tag that specifies the type of bean in a Spring configuration using XML.

Correct Answer CLASS ;

User Answer

Elapsed Time (seconds) 0

Explanation The correct answer is "CLASS." The bean tag has two distinctive attributes: id and class . The id attribute is used to give the bean its default name, while the class attribute specifies the bean's type.

Question 34 (ID: # 162633)	
Subject	Java Frameworks: Spring 2.5
Subtopic	Dependency Injection
Description	@Qualifier

You can use the `@Qualifier` annotation __________.

A.	on constructors or classes
B.	on fields, parameters, or custom annotations
C.	on custom events, parameters, or controller annotations
D.	on fields or bean property getter methods
E.	only on fields or parameters

Correct Answer B

User Answer

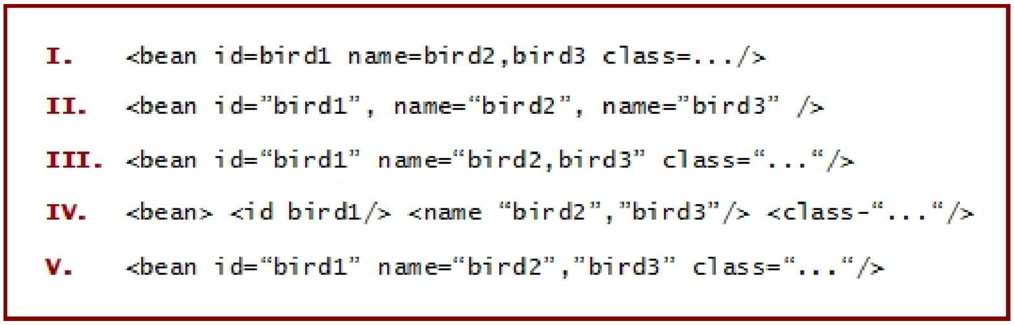
Elapsed Time (seconds) 0

Explanation Option B is correct. The @Qualifier annotation can be used on fields, parameters, or on custom annotations.

Question 35 (ID: # 162634)	
Subject	Java Frameworks: Spring 2.5

Subtopic	Dependency Injection
Description	Multiple Names for a Single Bean

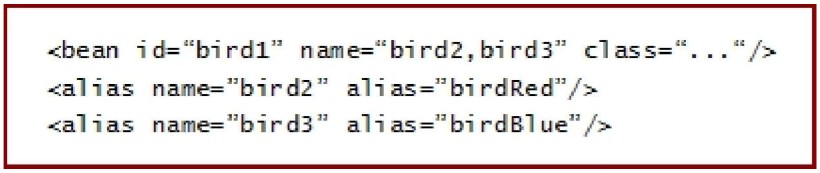
Suppose you want a simple bean configuration that defines multiple names for a single bean: one using the `id` attribute and two others as a comma-separated list in the `name` attribute. Which of the examples below would you use?



```
I.   <bean id=bird1 name=bird2,bird3 class=.../>
II.  <bean id="bird1", name=“bird2", name="bird3" />
III. <bean id=“bird1" name=“bird2,bird3" class=“...“/>
IV.  <bean> <id bird1/> <name “bird2","bird3"/> <class-“...“/>
V.   <bean id=“bird1" name=“bird2","bird3" class=“...“/>
```

A.	I
B.	II
C.	III
D.	IV
E.	V

Correct Answer	C
User Answer	
Elapsed Time (seconds)	0
Explanation	Option C is correct. The correct syntax for defining a simple bean configuration with multiple names for a single bean with the desired attributes is the following: .



```
<bean id=“bird1" name=“bird2,bird3" class=“...“/>
<alias name="bird2" alias="birdRed"/>
<alias name="bird3" alias="birdBlue"/>
```

Question 36 (ID: # 162637)

Subject	Java Frameworks: Spring 2.5
Subtopic	Framework Environment
Description	Transaction Management

Which type of transaction management is preferable when an application has many transactional operations?

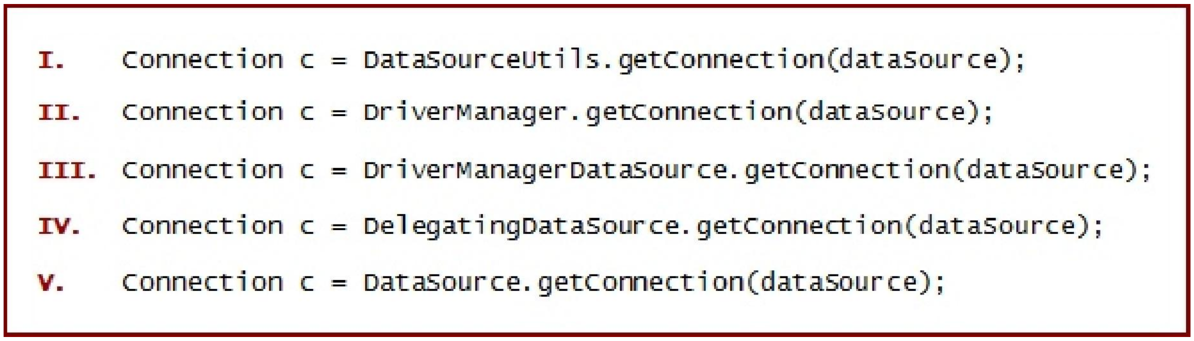
A.	declarative
B.	programmatic
C.	integrable
D.	methodive
E.	wiredible

Correct Answer	A
User Answer	
Elapsed Time (seconds)	0
Explanation	Option A is correct. There are only two transaction management types: declarative and programmatic . Declarative transaction management is preferred over programmatic transaction management when there are a

large number of transactions, so that the transaction code won't be littered in your business logic.

Question 37 (ID: # 162638)	
Subject	Java Frameworks: Spring 2.5
Subtopic	Framework Environment
Description	getConnection() Method

Which of the numbered syntax lines below should be used to call the `getConnection()` method in Spring 2.5?



```
I.    Connection c = DataSourceUtils.getConnection(dataSource);
II.   Connection c = DriverManager.getConnection(dataSource);
III.  Connection c = DriverManagerDataSource.getConnection(dataSource);
IV.   Connection c = DelegatingDataSource.getConnection(dataSource);
V.    Connection c = DataSource.getConnection(dataSource);
```

A. I
B. II
C. III
D. IV
E. V

Correct Answer	A
User Answer	
Elapsed Time (seconds)	0
Explanation	Option A is correct. You use the syntax below to call the getConnection() method. Connection c = DataSourceUtils.getConnection(dataSource)

Question 38 (ID: # 162639)	
Subject	Java Frameworks: Spring 2.5
Subtopic	Framework Environment
Description	TestContext / TestContextManager

What is the difference between `TestContext` and `TestContextManager` classes?

A. TestContext is used to create a new test, and TestContextManager encapsulates all of the tests executed up to a certain point.

B. TestContext encloses the context in which a test is executed, and TestContextManager manages a TestContext which holds the context of the current test.

C. TestContext is responsible for the current test, and TestContextManager manages all the tests of an application.

D. TestContext creates a new context, and TestContextManager encapsulates the first and last test contexts of an application.

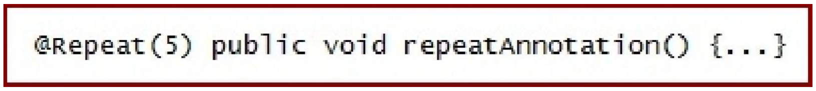
E. TestContext provides support for dependency injection, and TestContextManager encapsulates all the injection points of an application.

Correct Answer	B
User Answer	
Elapsed Time (seconds)	0
Explanation	Option B is correct. The TestContext class encloses the context in which a test is executed, while the TestContextManager class manages a TestContext which holds the context of the current test.

Question 39 (ID: # 162642)

Subject	Java Frameworks: Spring 2.5
Subtopic	Framework Environment
Description	@Repeat

What does the `@Repeat` annotation indicate in the example shown below?



```
@Repeat(5) public void repeatAnnotation() {...}
```

A. The method must be executed five times.

B. The method will repeat the execution of Line 5.

C. All of the methods after this one will be executed five times.

D. The line is incorrect; the @Repeat annotation requires no argument.

E. none of the above

Correct Answer	A
User Answer	
Elapsed Time (seconds)	0
Explanation	Option A is correct. The @Repeat annotation indicates that the method will execute repeatedly, five times.

Question 40 (ID: # 162643)

Subject	Java Frameworks: Spring 2.5
Subtopic	Framework Environment
Description	JDBC Common Annotations

Which of the following is not a Spring 2.5 annotation used for testing support?

A. @IfProfileValue

B. @Timed

C. @ExpectedException

D. @Repeat

E. @ShowException

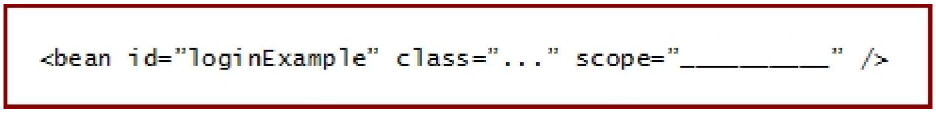
Correct Answer	E
User Answer	

Elapsed Time (seconds)	0
Explanation	Option E is correct. The @ShowException annotation is not a Spring 2.5 annotation.

Question 41 (ID: # 162644)	
Subject	Java Frameworks: Spring 2.5
Subtopic	Framework Environment
Description	Bean Scopes: Request Scope

The ________ bean scope is available only if you are using a web-aware Spring `ApplicationContext` implementation. It scopes a single bean definition to the lifecycle of each HTTP request, as shown below.

Type in the word that correctly completes the statement above.

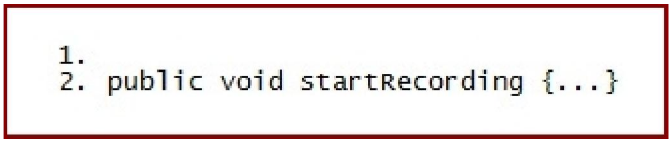


```
<bean id="loginExample" class="..." scope="_________" />
```

Correct Answer	REQUEST ;
User Answer	
Elapsed Time (seconds)	0
Explanation	The correct answer is "request." The request scope of a bean is available only if you are using a web-aware Spring ApplicationContext implementation. It scopes a single bean definition to the lifecycle of each HTTP request. The session scope deals with the HTTP session.

Question 42 (ID: # 162647)	
Subject	Java Frameworks: Spring 2.5
Subtopic	Framework Environment
Description	@PostConstruct

What annotation should be inserted at Line 1 below, so that the method gets called upon bean initialization, and no additional XML configuration is necessary?



```
1.
2. public void startRecording {...}
```

A.	@AfterInitialize
B.	@OnInitialize
C.	@PreDestroy
D.	@PostConstruct
E.	@Activate

Correct Answer	D
User Answer	
Elapsed Time (seconds)	0

Explanation Option D is correct. The @PostConstruct annotation is used when an object requires the invocation of a callback method upon initialization. The @PreDestroy annotation is used when the application context hosting that object is closed (i.e., stopRecording).

Question 43 (ID: # 162648)

Subject Java Frameworks: Spring 2.5

Subtopic Framework Environment

Description Dom4j and Log4j

What is the role of `Dom4j` and `Log4j`, respectively, in Spring 2.5?

A. Dom4j is used for creating domain objects, and Log4j is used for persistence logic.

B. Dom4j is used for domain object injection, and Log4j is used for logic data access.

C. Dom4j is used for the domain layer, and Log4j is used for the configuration file.

D. Dom4j is used for parsing XML, and Log4j is used for application logging.

E. Dom4j is used for controller integration, and Log4j is used for logic constructor implementation.

Correct Answer D

User Answer

Elapsed Time (seconds) 0

Explanation Option D is correct. Dom4j is used for parsing XML, while Log4j is used for logging in your applications.

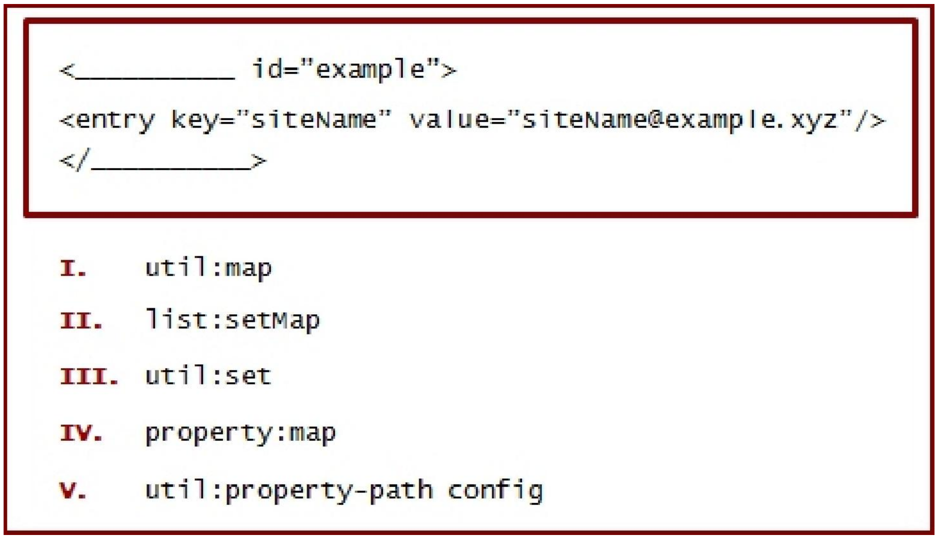
Question 44 (ID: # 162649)

Subject Java Frameworks: Spring 2.5

Subtopic Framework Environment

Description ApplicationContext Syntax

Which of the numbered elements below is the application context syntax used to create a `map` in Spring 2.5?



```
<__________ id="example">
<entry key="siteName" value="siteName@example.xyz"/>
</__________>
```

I. util:map

II. list:setMap

III. util:set

IV. property:map

V. util:property-path config

A. I

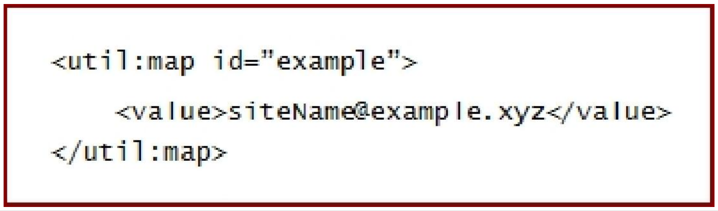
B. II

C. III

D. IV

E. V

Correct Answer	A
User Answer	
Elapsed Time (seconds)	0
Explanation	Option A is correct. The application context syntax to create a map is util:map , as shown in the code snippet below.



```
<util:map id="example">
    <value>siteName@example.xyz</value>
</util:map>
```

Question 45 (ID: # 162652)

Subject	Java Frameworks: Spring 2.5
Subtopic	Framework Environment
Description	Component Scan

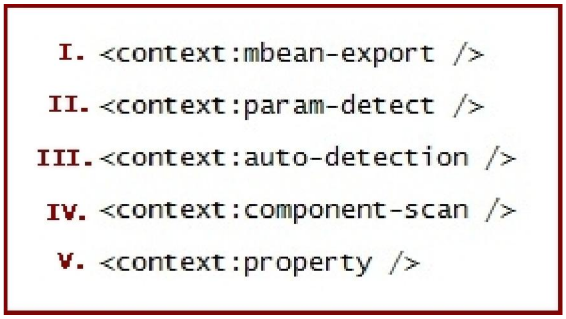
In Spring 2.5, you can only partially get rid of XML configurations.

Correct Answer	FALSE
User Answer	
Elapsed Time (seconds)	0
Explanation	This statement is false. Many people incorrectly think that is still needed for annotation-driven configuration. However, you can completely get rid of XML configurations if you use Spring annotations, and you can register the annotation-related processors with the Spring context.

Question 46 (ID: # 162653)

Subject	Java Frameworks: Spring 2.5
Subtopic	Framework Environment
Description	Auto-Detection

Which of the numbered examples below can be used to make auto-detection possible for an `@Controller` annotation?



```
  I. <context:mbean-export />
 II. <context:param-detect />
III. <context:auto-detection />
 IV. <context:component-scan />
  V. <context:property />
```

A. I

B. II

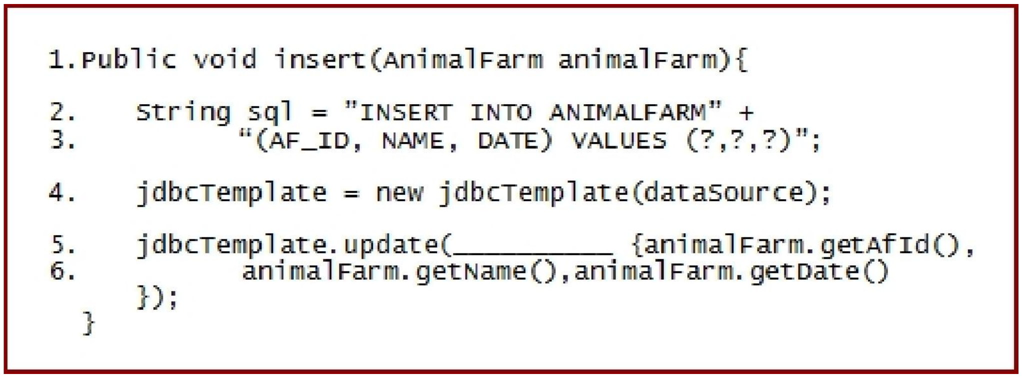
C. III
D. IV
E. V

Correct Answer	D
User Answer	
Elapsed Time (seconds)	0
Explanation	Option D is correct. You can use the element to make auto-detection possible for an @Controller annotation.

Question 47 (ID: # 162654)

Subject	Java Frameworks: Spring 2.5
Subtopic	Framework Environment
Description	JdbcTemplate

Suppose you are using Spring `jdbcTemplate`, and you want to update the `farmAnimal` table with the new values provided in the code below. Which of the following options should be used to fill in the blank at Line 5?



```
1.Public void insert(AnimalFarm animalFarm){

2.    String sql = "INSERT INTO ANIMALFARM" +
3.           “(AF_ID, NAME, DATE) VALUES (?,?,?)";

4.    jdbcTemplate = new jdbcTemplate(dataSource);

5.    jdbcTemplate.update(__________ {animalFarm.getAfId(),
6.           animalFarm.getName(),animalFarm.getDate()
      });
  }
```

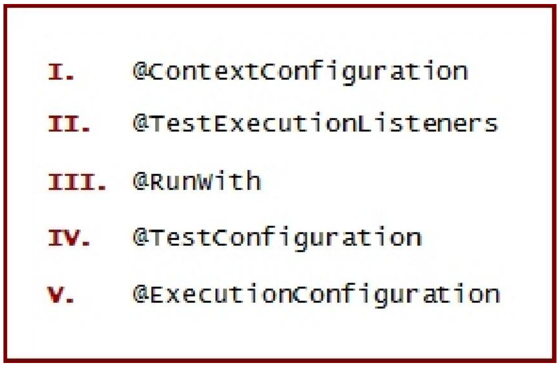
A. sql.Object[]
B. animalFarm, new Object[]
C. sql, new animalFarm[]
D. sql, new Object[]
E. new Object[].sql

Correct Answer	D
User Answer	
Elapsed Time (seconds)	0
Explanation	Option D is correct. jdbcTemplate helps you save time because it handles redundant code automatically. The missing element from the code is sql, new Object[] .

Question 48 (ID: # 162667)

Subject	Java Frameworks: Spring 2.5
Subtopic	Inversion of Control Container (IoC)
Description	Configure Your Test Classes

Which of the following numbered elements can be used to configure your own test classes in Spring 2.5?



```
I.    @ContextConfiguration

II.   @TestExecutionListeners

III.  @RunWith

IV.   @TestConfiguration

V.    @ExecutionConfiguration
```

A. I
B. II
C. III
D. IV
E. V

Correct Answer A ;B ;C ;

User Answer

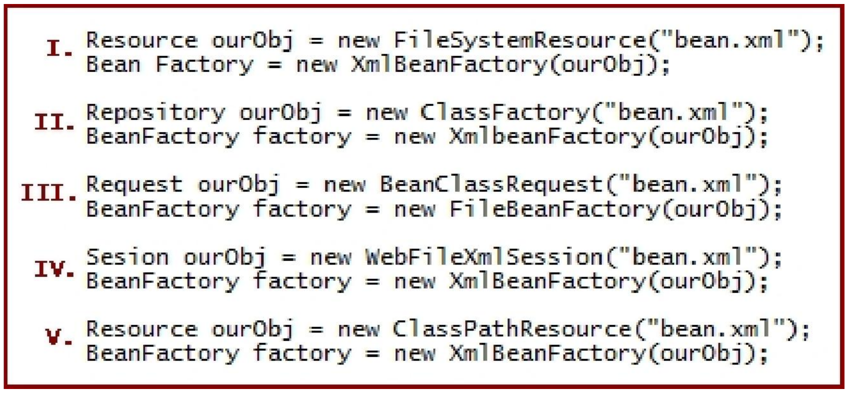
Elapsed Time (seconds) 0

Explanation Options A, B, and C are correct. From the options provided, only the first three are used to configure your own test class: @ContextConfiguration , @TestExecutionListeners , and @RunWith() .

Question 49 (ID: # 162668)

Subject	Java Frameworks: Spring 2.5
Subtopic	Inversion of Control Container (IoC)
Description	Bean Factory

Which of the numbered examples below can be used to instantiate a Bean Factory?



```
  I. Resource ourObj = new FileSystemResource("bean.xml");
     Bean Factory = new XmlBeanFactory(ourObj);

 II. Repository ourObj = new ClassFactory("bean.xml");
     BeanFactory factory = new XmlbeanFactory(ourObj);

III. Request ourObj = new BeanClassRequest("bean.xml");
     BeanFactory factory = new FileBeanFactory(ourObj);

 IV. Sesion ourObj = new WebFileXmlSession("bean.xml");
     BeanFactory factory = new XmlBeanFactory(ourObj);

  V. Resource ourObj = new ClassPathResource("bean.xml");
     BeanFactory factory = new XmlBeanFactory(ourObj);
```

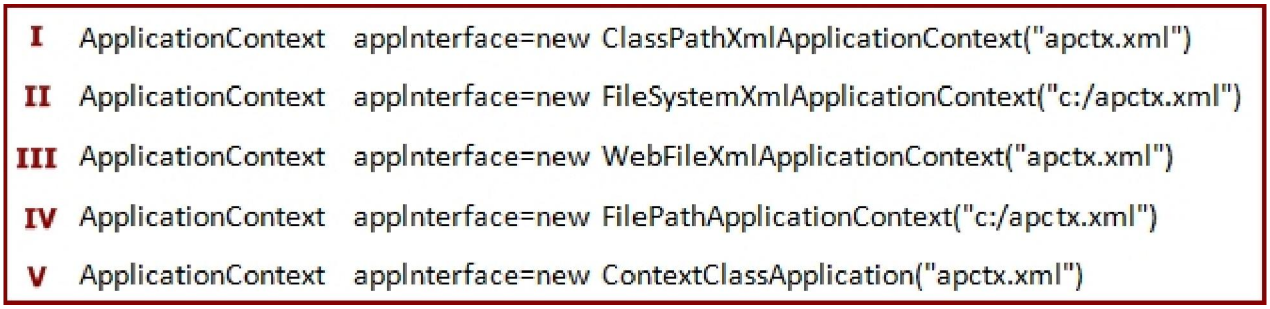
A. I
B. II
C. III
D. IV
E. V

Correct Answer E

User Answer	
Elapsed Time (seconds)	0
Explanation	Option E is correct. A Bean Factory is instantiated as follows. Resource ourObj = new ClassPathResource("bean.xml"); BeanFactory factory = new XmlBeanFactory(ourObj)

Question 50 (ID: # 162669)	
Subject	Java Frameworks: Spring 2.5
Subtopic	Inversion of Control Container (IoC)
Description	ApplicationContext

Which of the numbered examples below can be used to create the `ApplicationContext` interface?



```
I    ApplicationContext  applInterface=new ClassPathXmlApplicationContext("apctx.xml")
II   ApplicationContext  applInterface=new FileSystemXmlApplicationContext("c:/apctx.xml")
III  ApplicationContext  applInterface=new WebFileXmlApplicationContext("apctx.xml")
IV   ApplicationContext  applInterface=new FilePathApplicationContext("c:/apctx.xml")
V    ApplicationContext  applInterface=new ContextClassApplication("apctx.xml")
```

A. I

B. II

C. III

D. IV

E. V

Correct Answer	A ;B ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Options A and B are correct. There are three ways to create the ApplicationContext interface: ClassPathXmlApplicationContext , FileSystemXmlApplicationContext , and XmlWebApplicationContext .

Question 51 (ID: # 162672)	
Subject	Java Frameworks: Spring 2.5
Subtopic	Inversion of Control Container (IoC)
Description	Lifecycle Annotations

Which of the following options are Spring 2.5 annotations?

A. @PostConstruct

B. @Component

C. @Aspect

D. @PreDestroy

E. @Controller

Correct Answer	A ;B ;C ;D ;E ;
User Answer	
Elapsed Time (seconds)	0
Explanation	All of the options are correct; they are all lifecycle annotations in Spring 2.5.

Question 52 (ID: # 162673)

Subject	Java Frameworks: Spring 2.5
Subtopic	Inversion of Control Container (IoC)
Description	Auto-Detection of Spring Components

Spring 2.5 provides auto-detection of Spring Components. Which of the following annotations are used for this?

A.	@Component
B.	@Service
C.	@Repository
D.	@Controller
E.	@Autowired

Correct Answer	A ;B ;C ;D ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Options A, B, C, and D are correct. The annotations used for auto-detection of Spring Compenents are @ Component , @Service , @Repository , and @Controller .

Question 53 (ID: # 162674)

Subject	Java Frameworks: Spring 2.5
Subtopic	Inversion of Control Container (IoC)
Description	Meta-Annotation

What is the difference between an annotation and a meta-annotation?

A. An annotation is a meta-tag used to give life to your code. A meta-annotation is an annotation that annotates other annotations.

B. An annotation is the @ symbol, followed by the annotation name. A meta-annotation is the @ symbol, followed by the annotation name and data added within parentheses.

C. An annotation is the @ symbol, followed by a name. A meta-annotation is an annotation with multiple data elements.

D. An annotation is a name followed by the @ symbol. A meta-annotation is an annotation with two or more elements.

E. An annotation is a name followed by the @ symbol. A meta-annotation is a name preceded and followed by @ symbols.

Correct Answer	A
User Answer	

Elapsed Time (seconds) 0

Explanation Option A is correct. An annotation is a meta-tag used to give life to your code. A meta-annotation is an annotation that annotates other annotations.

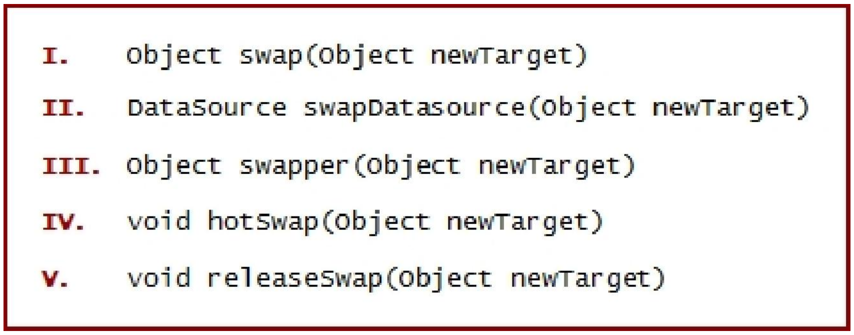
Question 54 (ID: # 162599)

Subject Java Frameworks: Spring 2.5

Subtopic Spring AOP

Description Target Sources

If you want to change the datasource while the application is running, which of the numbered examples below is the `HotSwappableTargetSource` method to call?



```
I.   Object swap(Object newTarget)
II.  DataSource swapDatasource(Object newTarget)
III. Object swapper(Object newTarget)
IV.  void hotSwap(Object newTarget)
V.   void releaseSwap(Object newTarget)
```

A. I

B. II

C. III

D. IV

E. V

Correct Answer A

User Answer

Elapsed Time (seconds) 0

Explanation Option A is correct. The datasource can be swapped while the application is running by calling the Object swap (Object newTarget) method.

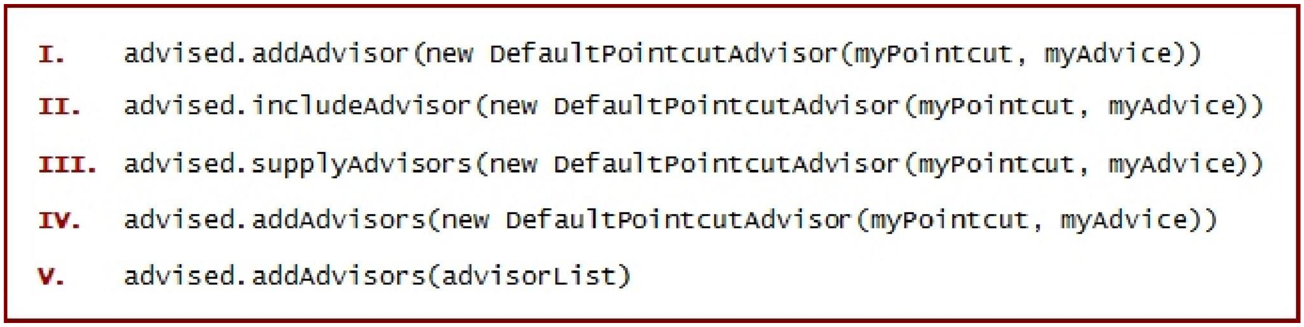
Question 55 (ID: # 162602)

Subject Java Frameworks: Spring 2.5

Subtopic Spring AOP

Description Add Selective Advisors

Which of the numbered examples below can be used to add selective advice using a `pointcut` to an Advised object defined as `Advised advised = getAdvised();`?



```
I.    advised.addAdvisor(new DefaultPointcutAdvisor(myPointcut, myAdvice))
II.   advised.includeAdvisor(new DefaultPointcutAdvisor(myPointcut, myAdvice))
III.  advised.supplyAdvisors(new DefaultPointcutAdvisor(myPointcut, myAdvice))
IV.   advised.addAdvisors(new DefaultPointcutAdvisor(myPointcut, myAdvice))
V.    advised.addAdvisors(advisorList)
```

A. I
B. II
C. III
D. IV
E. V

Correct Answer A

User Answer

Elapsed Time (seconds) 0

Explanation Option A is correct. You must use advised.addAdvisor() in order to add selective advice using a pointcut .

Question 56 (ID: # 162603)

Subject Java Frameworks: Spring 2.5
Subtopic Spring AOP
Description getAdvisors()

Which of the following methods can be used to get all the advisors for the `aspectj` instance?

A. getAdvisor()
B. returnAdvisor()
C. getAdvisors()
D. obtainAdvisors()
E. moveAdvisor()

Correct Answer C

User Answer

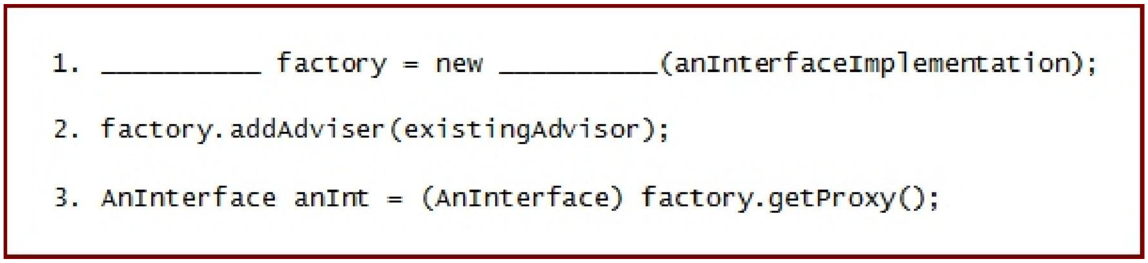
Elapsed Time (seconds) 0

Explanation Option C is correct. The method that will return an Advisor for every advisor is the getAdvisors() method.

Question 57 (ID: # 162604)

Subject Java Frameworks: Spring 2.5
Subtopic Spring AOP
Description ProxyFactory

If you want to create a proxy for a target object having one advisor, which of the following elements should be added to fill in the blanks in Line 1 of the code shown below?



```
1. __________ factory = new __________(anInterfaceImplementation);

2. factory.addAdviser(existingAdvisor);

3. AnInterface anInt = (AnInterface) factory.getProxy();
```

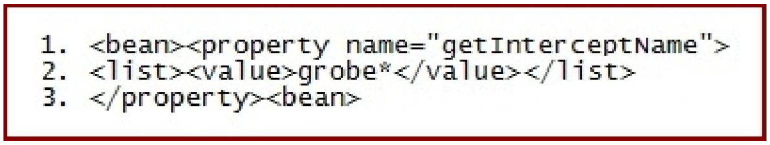
A. BeanFactory
B. AdvisorFactory
C. ProxyFactory
D. SessionFactory
E. SpringFactory

Correct Answer	C
User Answer	
Elapsed Time (seconds)	0
Explanation	Option C is correct. ProxyFactory will proxy the interfaces implemented by the object.

Question 58 (ID: # 162607)

Subject	Java Frameworks: Spring 2.5
Subtopic	Spring AOP
Description	AOP Proxies

When you are using AOP proxies, what does the asterisk in Line 2 below indicate, if it is preceded by an interceptor name?



```
1. <bean><property name="getInterceptName">
2. <list><value>grobe*</value></list>
3. </property><bean>
```

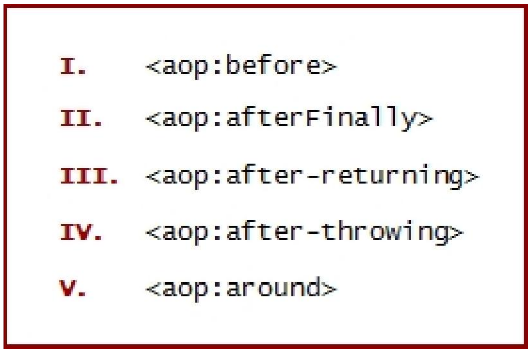
A. All advisors with a bean name that starts with the element preceding the asterisk will join the advisor series.
B. All advisors with a bean name containing the element preceding the asterisk anywhere in the bean name will join the advisor series.
C. All advisors containing the element preceded by the asterisk will be counted.
D. grobe is a key word, and an exception will be thrown.
E. All advisors whose bean names contain the word grobe will be eliminated.

Correct Answer	A
User Answer	
Elapsed Time (seconds)	0
Explanation	Option A is correct. When an asterisk is appended to an interceptor name, all advisors with a bean name that matches the element preceding the asterisk will join the advisor series.

Question 59 (ID: # 162608)

Subject	Java Frameworks: Spring 2.5
Subtopic	Spring AOP
Description	Declaring an Advice

Which of the numbered elements below are supported by the `AspectJ` style for an `advice` declaration?



```
I.    <aop:before>
II.   <aop:afterFinally>
III.  <aop:after-returning>
IV.   <aop:after-throwing>
V.    <aop:around>
```

A. I
B. II
C. III
D. IV
E. V

Correct Answer A ;C ;D ;E ;

User Answer

Elapsed Time (seconds) 0

Explanation Options A, C, D, and E are correct. There are five kinds of advice supported by the AspectJ style: before , after , after throwing , after returning and around . The afterFinally advice does not exist.

Question 60 (ID: # 162609)

Subject	Java Frameworks: Spring 2.5
Subtopic	Spring AOP
Description	Introduction

Which of the following should you use to make an introduction in `AspectJ`?

A. @DeclareIntroduction
B. @BeanIntroduction
C. @DeclareParents
D. @IntroductionAspect
E. @DeclareChild

Correct Answer C

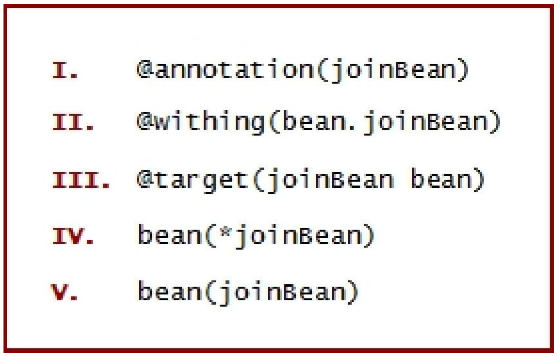
User Answer

Elapsed Time (seconds) 0

Explanation Option C is correct. You can make an introduction in AspectJ using the @DeclareParents annotation.

Question 61 (ID: # 162612)	
Subject	Java Frameworks: Spring 2.5
Subtopic	Spring AOP
Description	Join Point

Which of the numbered examples below is a `join point` (method execution only in Spring AOP) on a Spring bean named `joinBean`?



```
I.    @annotation(joinBean)
II.   @withing(bean.joinBean)
III.  @target(joinBean bean)
IV.   bean(*joinBean)
V.    bean(joinBean)
```

A. I
B. II
C. III
D. IV
E. V

Correct Answer D ;E ;

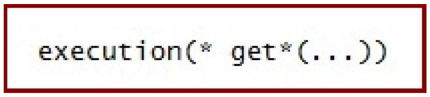
User Answer

Elapsed Time (seconds) 0

Explanation Options D and E are correct. Both lines of code will allow for a join point on all methods of a bean named joinBean , since the wildcard matching of Option D will also succeed.

Question 62 (ID: # 162613)	
Subject	Java Frameworks: Spring 2.5
Subtopic	Spring AOP
Description	Pointcut Execution

In Spring AOP, the execution `pointcut` expression shown below __________.



```
execution(* get*(...))
```

A. executes a private method
B. executes any public method
C. executes any method whose name begins with get
D. executes any method whose name contains get

E. executes a method defined by the get method

Correct Answer C

User Answer

Elapsed Time (seconds) 0

Explanation Option C is correct. The expression above executes any method whose name begins with get .

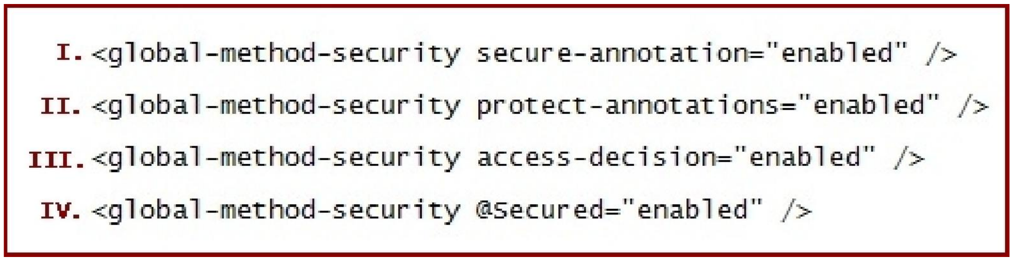
Question 63 (ID: # 162614)

Subject Java Frameworks: Spring 2.5

Subtopic Spring AOP

Description Security with AOP

Which of the numbered examples below enables support for a security method with AOP using the `@Secured` annotation?



```
  I. <global-method-security secure-annotation="enabled" />
 II. <global-method-security protect-annotations="enabled" />
III. <global-method-security access-decision="enabled" />
 IV. <global-method-security @Secured="enabled" />
```

A. I

B. II

C. III

D. IV

E. none of the above

Correct Answer E

User Answer

Elapsed Time (seconds) 0

Explanation Option E is correct. None of the declarations is correct. The declaration that enables support for a security method with AOP using the @Secured annotation is .

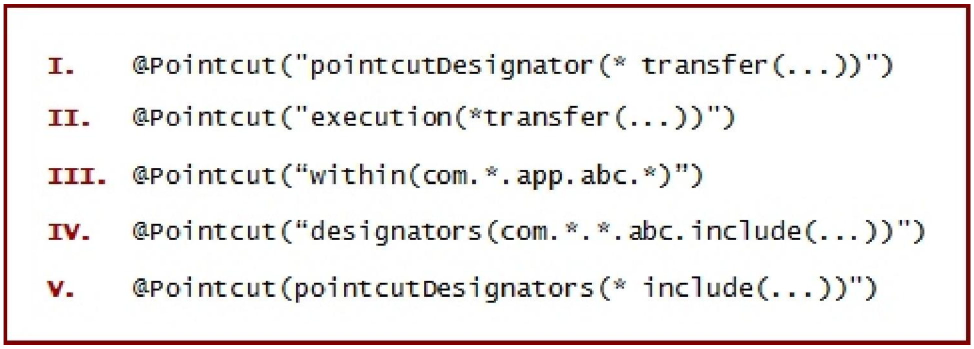
Question 64 (ID: # 162617)

Subject Java Frameworks: Spring 2.5

Subtopic Spring AOP

Description Pointcut Declaration

Which of the numbered examples below is the correct way of declaring a `pointcut` in Spring AOP?



```
I.    @Pointcut("pointcutDesignator(* transfer(...))")
II.   @Pointcut("execution(*transfer(...))")
III.  @Pointcut(“within(com.*.app.abc.*)")
IV.   @Pointcut(“designators(com.*.*.abc.include(...))")
V.    @Pointcut(pointcutDesignators(* include(...))")
```

A. I

B. II

C. III

D. IV

E. V

Correct Answer A

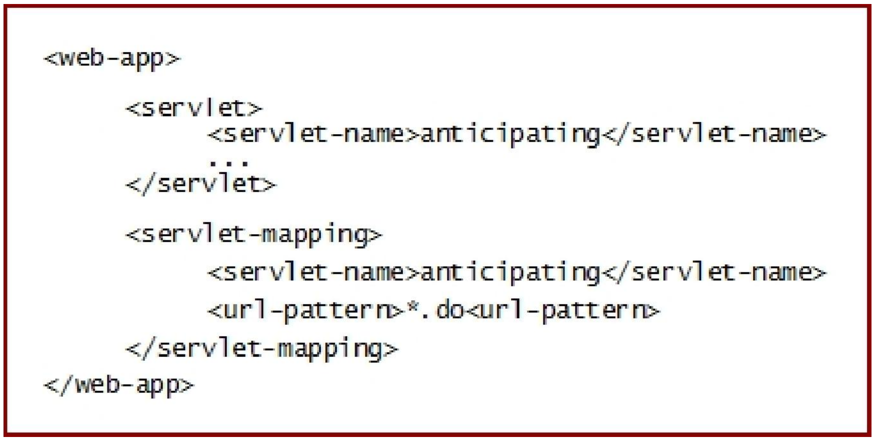
User Answer

Elapsed Time (seconds) 0

Explanation Option A is correct. A pointcut is declared in Spring AOP using the following expression: @Pointcut(" pointcutDesignator(* transfer(...))") .

Question 65 (ID: # 162657)	
Subject	Java Frameworks: Spring 2.5
Subtopic	Spring MVC (Model-View-Controller)
Description	DispatcherServlet

For the code fragment shown below, from the `DispatcherServlet` servlet configuration (in the `web.xml` file), you will need to have a file called `/WEB-INF/anticipating-servlet.xml` in your application, to contain all of your beans.



```
<web-app>

    <servlet>
        <servlet-name>anticipating</servlet-name>
        ...
    </servlet>

    <servlet-mapping>
        <servlet-name>anticipating</servlet-name>
        <url-pattern>*.do<url-pattern>
    </servlet-mapping>
</web-app>
```

Correct Answer TRUE

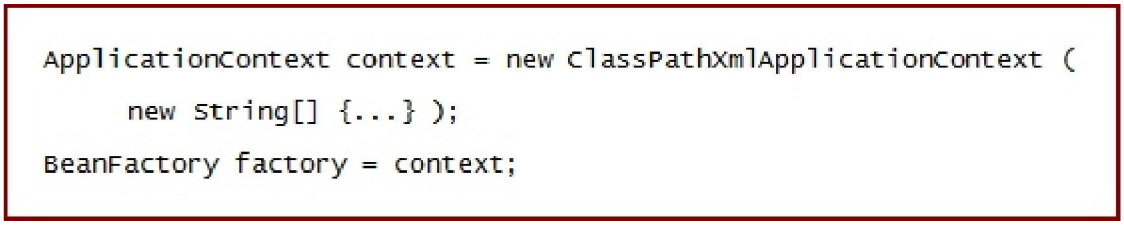
User Answer

Elapsed Time (seconds) 0

Explanation The statement is true. Given the code fragment of the DispatcherServlet servlet configuration, you will need to have a file called /WEB-INF/anticipating-servlet.xml in your application, to contain all of your beans.

Question 66 (ID: # 162658)	
Subject	Java Frameworks: Spring 2.5
Subtopic	Spring MVC (Model-View-Controller)
Description	BeanFactory

The code snippet shown below instantiates a Spring IoC (Inversion of Control) Container.



```
ApplicationContext context = new ClassPathXmlApplicationContext (
     new String[] {...} );
BeanFactory factory = context;
```

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	The statement is true. The code snippet instantiates a Spring IoC. This is because, via inheritance, an ApplicationContext is also a BeanFactory .

Question 67 (ID: # 162659)	
Subject	Java Frameworks: Spring 2.5
Subtopic	Spring MVC (Model-View-Controller)
Description	Servlet & Portlet

Which of the following are controllers that can quickly access `Servlet MVC` or `Portlet MVC` features, if desired?

A.	@PreDestroy
B.	@RequestMapping
C.	@PostConstruct
D.	@RequestParam
E.	@ModelAttribute

Correct Answer	B ;D ;E ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Options B, D, and E are correct. @RequestMapping , @RequestParam , and @ModelAttribute are annotation-based controllers. @PreDestroy and @PostConstruct are configuration annotations.

Question 68 (ID: # 162662)	
Subject	Java Frameworks: Spring 2.5

Subtopic	Spring MVC (Model-View-Controller)
Description	@RequestParam

Which of the following annotations should you use to connect request parameters to a method parameter in your controller?

A.	@RequestMethod
B.	@MethodParam
C.	@Controller
D.	@RequestParam
E.	@RequestMapping

Correct Answer	D
User Answer	
Elapsed Time (seconds)	0
Explanation	Option D is correct. Use the @RequestParam annotation to connect request parameters to a method parameter in your controller.

Question 69 (ID: # 162663)

Subject	Java Frameworks: Spring 2.5
Subtopic	Spring MVC (Model-View-Controller)
Description	@MVC

Which of the following are Spring MVC annotations?

A.	@Controller
B.	@RequestMethod
C.	@RequestParam
D.	@InitBinder
E.	@SessionAttributes

Correct Answer	A ;B ;C ;D ;E ;
User Answer	
Elapsed Time (seconds)	0
Explanation	All of the options are correct; they are all Spring MVC annotations.

Question 70 (ID: # 162664)

Subject	Java Frameworks: Spring 2.5
Subtopic	Spring MVC (Model-View-Controller)
Description	MVC Controllers

Controllers understand user input and change it into a model which is represented by the view.

Correct Answer	TRUE
User Answer	

Elapsed Time (seconds)	0
Explanation	This statement is true. Controllers understand user input and change it into a model which is represented by the view. This is the basic definition of the "C" part of MVC.

Question 71 (ID: # 56408)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	ORM (Object Relational Mapping)

Which of the following statements about Hibernate are true?

A.	Hibernate is an Object/Relational framework.
B.	Hibernate is a Query framework.
C.	Hibernate follows object-oriented principles.
D.	Hibernate is an RDBMS.
E.	Hibernate allows SQL queries.

Correct Answer	A ;B ;C ;E ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Hibernate allows you to develop persistent classes following object-oriented principles such as association, inheritance, polymorphism, composition, and collections. Object/Relational persistence and Query Service implementation offer potent APIs that allow you to develop persistent classes following the object-oriented idiom. Both implementations support the utilization of association, inheritance, polymorphism, composition, and collections with persistent classes. To completely employ persistent class implementations, you must learn how to generate and execute queries using EJB-QL, the portable Hibernate SQL extension (HQL), native SQL, and the object-oriented Criteria, and Example API.

Question 72 (ID: # 56409)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Hierarchies in Hibernate

Hibernate allows you to mix Table-per-Class hierarchy and Table-per-Subclass strategies.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	It's true. Hibernate does not support mixing , and mappings under the same root element. You can mix the table per hierarchy and table per subclass strategies under the same element, by merging the and elements.

Question 73 (ID: # 56422)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Immutability

To make a property immutable, so that it can be read from the database but not modified in any way, you should use the ________________ attribute.

A.	insertable=false
B.	updatable=false
C.	update=true
D.	insert=true
E.	set=true

Correct Answer	A ;B ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Options A and B are correct. By using both insertable=false and updatable=false , you can create a property that can be read from the database but not altered (immutable).

Question 74 (ID: # 56423)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Mapping

You can declare mappings for multiple classes in one mapping file.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	The statement is true; you can do this by using multiple elements. However, the recommended practice is to use one mapping file for each persistent class.

Question 75 (ID: # 56424)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Hibernate Mapping File

Hibernate-mapping is the root level element in a Hibernate mapping file.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0

Explanation	The statement is true; this is the main element in a Hibernate mapping file.

Question 76 (ID: # 56427)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Configuration

Hibernate is configured to access an instance variable directly by mapping the property with `access="field"`.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	The statement is true. This forces Hibernate to bypass the Setter method and access the instance variable directly, while initializing a newly-loaded object.

Question 77 (ID: # 56428)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Persistent Classes in Hibernate

What methods must the `Persistent` classes implement in Hibernate?

A.	a constructor with no arguments for every Persistent class
B.	Getter and Setter methods for all of the Instance variables
C.	a constructor with an argument for every Persistent class
D.	only Getter methods for all Instance variables
E.	only Setter methods for all Instance variables

Correct Answer	A ;B ;
User Answer	
Elapsed Time (seconds)	0
Explanation	A and B are correct. Since Hibernate instantiates Persistent classes using Constructor.newInstance() , it requires a constructor with no arguments for every Persistent class, and Getter and Setter methods for all of the Instance variables.

Question 78 (ID: # 56429)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Serializable in Hibernate

Hibernate requires `Persistent` classes to implement `Serializable`.

Correct Answer	FALSE

User Answer	
Elapsed Time (seconds)	0
Explanation	Hibernate doesn't require Persistent classes to implement Serializable . However, when objects are stored in an HttpSession , or passed by value using RMI, serialization is necessary.

Question 79 (ID: # 56432)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Configuration

What happens if both the `hibernate.properties` and the `hibernate.cfg.xml` files are found in the application `classpath`?

A. Both files are used; the settings declared in the hibernate.cfg.xml file will override the settings declared in the hibernate.properties file.

B. Both files are used; the settings declared in the hibernate.properties file will override the settings declared in the hibernate.cfg.xml file

C. Only the hibernate.properties file is used.

D. Only the hibernate.cfg.xml file is used.

E. Placing these two files in the application classpath will result in a runtime fatal error.

Correct Answer	A
User Answer	
Elapsed Time (seconds)	0
Explanation	Both the hibernate.properties and the hibernate.cfg.xml files offer the same function: to configure Hibernate. Which file you decide to employ depends on your syntax preference. If you are using both hibernate.properties and hibernate.cfg.xml , the settings of the XML configuration file will override the settings employed in the Properties file.

Question 80 (ID: # 56433)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Configuring Hibernate

Which of the following are common methods of configuring Hibernate?

A. Place the hibernate.properties file in the class path.

B. Include hibernate.cfg.xml in the class path.

C. Place the Hibernate Mapping file in the class path.

D. Call the configure() method in the main class.

E. None of the options above are correct. Hibernate cannot be configured.

Correct Answer	A ;B ;
User Answer	
Elapsed Time (seconds)	0

Explanation	You can configure Hibernate by placing the hibernate.properties file in the classpath and including the elements in the hibernate.cfg.xml file in the class path.

Question 81 (ID: # 56434)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Hibernate Query Language (HQL)

Which of the following is not true about Hibernate Query Language (HQL)?

A.	It provides full support for relational operations.
B.	It is database-independent.
C.	It allows you to represent SQL queries in the form of objects.
D.	It can return results as Objects .
E.	It does not support aggregate functions.

Correct Answer	E
User Answer	
Elapsed Time (seconds)	0
Explanation	Options A, B, C, and D make HQL a powerful query language. It offers full support for relational operations; it permits you to return results as Objects ; it entirely supports polymorphic queries; and it is database-independent.

Question 82 (ID: # 56437)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Object-Relational Paradigm

Which of the following statements are true about the object-relational paradigm?

A.	The object-oriented paradigm is founded on software engineering principles, while the relational paradigm is founded on mathematical principles.
B.	The object-oriented paradigm is founded on mathematical principles, while the relational paradigm is founded on software engineering principles.
C.	The object-oriented technology is used to build applications out of networks of objects that have both data and behavior. The relational technology is used to store data in tables and to manipulate that data , using data manipulation language (DML).
D.	The object-oriented paradigm is based on math.
E.	The object-oriented paradigm supports the storage of data in tables.

Correct Answer	A ;C ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Options A and C are correct. The object-oriented paradigm is founded on software engineering principles, while the relational paradigm is founded on mathematical principles. Also, the object-oriented technology is used to build applications out of networks of objects that have both data and behavior. The relational technology is used

to store data in tables and to manipulate that data using data manipulation language (DML).

Question 83 (ID: # 56438)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	OptimisticLock Property

The automatic version increment for certain properties and collections may be disabled by setting the optimistic-lock mapping attribute to `False`.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	You can avoid automatic version incrementing by annotating the property (or collection) with @ OptimisticLock (excluded=true).

Question 84 (ID: # 56439)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Annotation Extensions

Which of the following packages include annotation extensions?

A.	org.apache.hibernate.annotations
B.	org.hibernate.annotations
C.	com.hibernate.annotations
D.	com.hibernate.extensions
E.	com.apache.hibernate.annotation

Correct Answer	B
User Answer	
Elapsed Time (seconds)	0
Explanation	To empower the EJB3 capabilities, Hibernate provides specific annotations that match Hibernate features. The org.hibernate.annotations package contains all of these annotation extensions.

Question 85 (ID: # 56442)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Entity Bean

What annotation is used for the declaration of every bound persistence POJO class?

A.	@Entity Bean

B. @Entity

C. @POJO Entity

D. @Entity POJO

E. @POJO

Correct Answer B

User Answer

Elapsed Time (seconds) 0

Explanation Every bound persistent POJO class is an entity bean and is declared using the @Entity annotation (at the class level).

```
@Entity
public class Flight implements Serializable {
 Long id;

 @Id
 public Long getId() { return id; }

 public void setId(Long id) { this.id = id; }
}
```

Question 86 (ID: # 56443)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Association

Associations that are related to container management persistence are called managed associations. From the options below, choose the statement about managed associations and Hibernate associations that is correct.

A. Both of the associations are unidirectional.

B. Both of the associations are bidirectional.

C. Managed associations are bidirectional and Hibernate associations are unidirectional.

D. Managed associations are unidirectional and Hibernate associations are bidirectional.

E. none of the above

Correct Answer C

User Answer

Elapsed Time (seconds) 0

Explanation Associations that are related to container management persistence are called managed associations. These are bidirectional associations, which means that if a change is made to one end of the association, it will be reflected at the other end. Hibernate associations, in contrast, are unidirectional.

Question 87 (ID: # 56444)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Interceptors

Which of the following statements about Hibernate interceptors is true?

A. They track statistic information.
B. They provide callbacks from the session.
C. They provide callbacks from the transaction.
D. They inspect a persistent object's properties before it is evicted or flushed.
E. They inspect a non-persistent object's properties before it is saved, updated, deleted, or loaded.

Correct Answer B

User Answer

Elapsed Time (seconds) 0

Explanation An interceptor may be used to intercept the existing business functionality, in order to offer extensible or add-on features. Hibernate interceptors offer pluggable architecture and are usually callback methods that will be called by the framework in response to a specific set of events or actions, if they are correctly registered and configured.

Question 88 (ID: # 56447)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Object Identification

____________ is a method by which identical objects are identified.

A. Object Identity
B. Object Equality
C. Database Identity
D. Object Similarity
E. Object Alias

Correct Answer A ;B ;C ;

User Answer

Elapsed Time (seconds) 0

Explanation Options A, B, and C are correct.
- Object Identity identifies objects that are identical because they reside in the same memory location in the JVM. This can be checked by using the = = operator.
- Object Equality identifies objects that are equal because they have the same value, as defined by the equals() method. Classes that don't explicitly override this method inherit the implementation defined by the java.lang. Object , which compares object identity.
- Database Identity identifies objects stored in a relational database that are identical because they represent the same row or share the same table and primary key value.

Question 89 (ID: # 56448)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	XDoclet

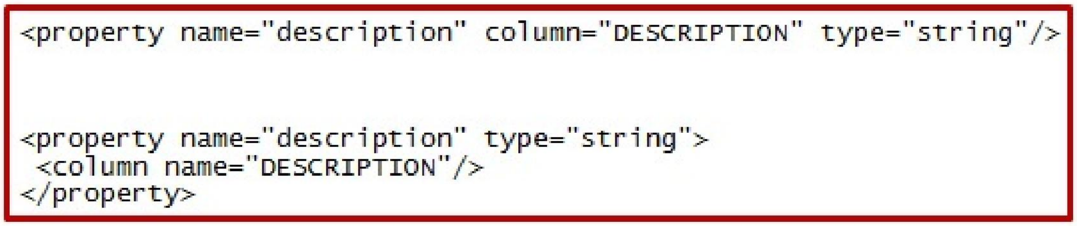
`XDoclet` uses the Javadoc tag format to specify the class-, field-, or method-level metadata.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	XDoclet has brought the notion of attribute-oriented programming to Java. Until JDK 1.5, the Java language had no support for annotations. XDoclet uses the Javadoc tag format (@attribute) to identify class-, field-, or method-level metadata attributes. These attributes are employed to produce a Hibernate mapping file automatically when the application is built.

Question 90 (ID: # 56449)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Property Mapping

The following property mapping alternatives are correct.



```
<property name="description" column="DESCRIPTION" type="string"/>


<property name="description" type="string">
 <column name="DESCRIPTION"/>
</property>
```

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	It's true, and this is the typical and most common property mapping.

Question 91 (ID: # 56452)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Transformation

Mappings between classes and tables, properties and columns, associations and foreign keys, and Java types and SQL types are defined as __________.

A. transformed relationships

B. persistence

C. Object Relational Mapping metadata

D. Object Relational Mapping relationships

E. transformations

Correct Answer	C
User Answer	

Elapsed Time (seconds)	0
Explanation	ORM tools require a metadata format to indicate the mapping between classes and tables, properties and columns, associations and foreign keys, and Java types and SQL types. This information is called Object Relational Mapping metadata. It defines the modification between the diverse data type systems and the relationship representations.

Question 92 (ID: # 56453)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	POJO

POJO is a simple JAVA __________.

A.	interface
B.	servlet
C.	object
D.	bean
E.	method

Correct Answer	C
User Answer	
Elapsed Time (seconds)	0
Explanation	POJOs are basic JAVA objects.

Question 93 (ID: # 56454)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	POJO

What does POJO stand for?

A.	Part Of Java Objects
B.	Plain Old Java Objects
C.	Package Of Java Objects
D.	Package Old Java Objects
E.	Property Of Java Objects

Correct Answer	B
User Answer	
Elapsed Time (seconds)	0
Explanation	POJO represents Plain Old Java Objects. These are just basic JavaBeans that have defined Setter and Getter methods for every property they have.

Question 94 (ID: # 56457)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	SessionFactory

Considering the J2EE environment, where should `SessionFactory` be placed so that it can be easily accessed?

A. in JNDI
B. in a well-defined class
C. before the declaration of a class
D. in memory
E. in a separate file

Correct Answer A

User Answer

Elapsed Time (seconds) 0

Explanation In the J2EE environment, if the SessionFactory is placed in JNDI, it can be easily accessed and shared among diverse threads and various elements that are Hibernate-aware. You can set the SessionFactory to a JNDI by configuring a property hibernate.session_factory_name in the hibernate.properties file.

Question 95 (ID: # 56458)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	SessionFactory

Which of the following methods is used for creating Session Factory?

A. createSessionFactory()
B. buildSessionFactory()
C. addSessionFactory()
D. create()
E. makeSessionFactory()

Correct Answer B

User Answer

Elapsed Time (seconds) 0

Explanation To generate Session Factory, you initially have to generate an instance of Configuration, then use that instance to refer to the location of the Configuration file. After configuring, this instance is used to generate the Session Factory by calling the buildSessionFactory() method.

Question 96 (ID: # 56459)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate

Description Event Listener

From the following options, select the ways you can generate and register your own load event listener programmatically, without disabling Hibernate's entity loading.

A. Implement LoadEventListener , then register it programmatically.

B. Implement LoadEventListener , then register it programmatically, along with the DefaultLoadEventListener .

C. Extend the DefaultLoadEventListener class, then register it programmatically.

D. Disable Hibernate's entity loading, in order to register your own load event listener programmatically.

E. Implement LockEventListener , then register it programmatically.

Correct Answer B ;C ;

User Answer

Elapsed Time (seconds) 0

Explanation Core loading functionality is offered through the DefaultLoadEventListener class. You can implement LoadEventListener and register it programmatically, together with the DefaultLoadEventListener . Alternatively, you can extend the DefaultLoadEventListener class and register it programmatically.

Question 97 (ID: # 56462)

Subject Java Technologies: Hibernate, Spring, and Struts

Subtopic Hibernate

Description Interceptor

A `SessionFactory`-scoped interceptor is registered with the configuration object before the `SessionFactory` is built.

```
new Configuration().setInterceptor( new AuditInterceptor() );
```

Correct Answer TRUE

User Answer

Elapsed Time (seconds) 0

Explanation A SessionFactory -scoped interceptor is registered with the Configuration object before the SessionFactory is built. In this situation, the offered interceptor will be applied to all sessions opened from that SessionFactory . This is true if a session that clearly states which interceptor to employ is not opened . SessionFactory -scoped interceptors must be thread-safe. Session-specific states should not be stored, since several sessions may use this interceptor concomitantly.

new Configuration().setInterceptor(new AuditInterceptor());

Question 98 (ID: # 56463)

Subject Java Technologies: Hibernate, Spring, and Struts

Subtopic Hibernate

Description Environments

JBoss, Weblogic, and Websphere are examples of _____________.

A. managed environments

B.	non-managed environments
C.	serializable environments
D.	management
E.	replicate environments

Correct Answer	A
User Answer	
Elapsed Time (seconds)	0
Explanation	In managed environments, everything from database connections to transaction boundaries and security levels is defined.

Question 99 (ID: # 56464)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Environments

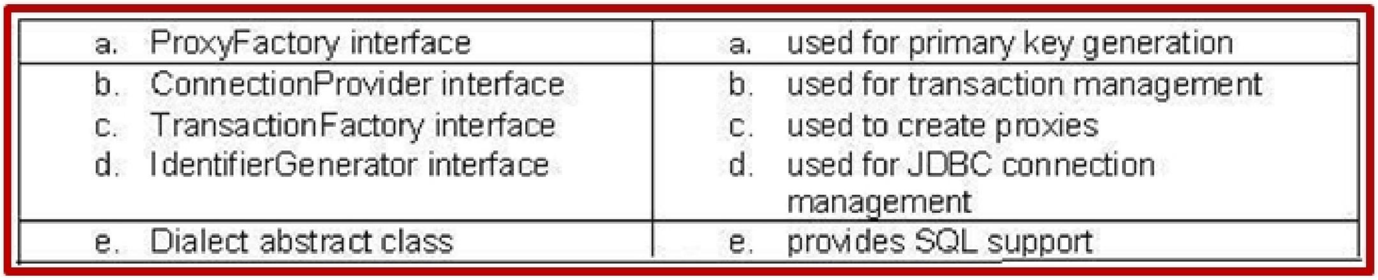
Hibernate may be configured to run in ___________ environments.

A.	isolated
B.	managed
C.	concurrent
D.	non-managed
E.	parallel

Correct Answer	B ;D ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Hibernate may be configured to run in both managed and non-managed environments. Managed environments pool resources such as database connections and allow transaction boundaries and security to be specified declaratively (that is, in metadata). A J2EE application server such as JBoss, BEA WebLogic, or IBM WebSphere implements the standard (J2EE-specific) managed environment for Java. Non-managed environments provide basic concurrency management via thread pooling. A servlet container like Jetty or Tomcat provides a non-managed server environment for Java Web applications. A stand-alone desktop or command-line application is also considered to be non-managed. Non-managed environments don't provide automatic transaction or resource management or security infrastructure. The application itself manages database connections and demarcates transaction boundaries.

Question 100 (ID: # 56467)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Hibernate Interfaces

Which of the following options provides a correct match for the interfaces in Hibernate?



a. ProxyFactory interface	a. used for primary key generation
b. ConnectionProvider interface c. TransactionFactory interface d. IdentifierGenerator interface	b. used for transaction management c. used to create proxies d. used for JDBC connection management
e. Dialect abstract class	e. provides SQL support

A. a - c; b - d; c - b; d - a; e - e

B. a - a; b - b; c - c; d - d; e - e

C. a - b; b - c; c - d; d - e; e - a

D. a - d; b - c; c - a; d - b; e - d

E. a - b; c - d; b - d; c - e; e - a

Correct Answer A

User Answer

Elapsed Time (seconds) 0

Explanation Option A is a correct match for the interfaces in Hibernate.

Question 101 (ID: # 56468)

Subject Java Technologies: Hibernate, Spring, and Struts

Subtopic Hibernate

Description Interfaces for ORM Strategies

Which of the following interfaces provides ORM strategies?

A. Extension

B. Callback

C. ClassPersister

D. Base

E. IdentifierGenerator

Correct Answer C

User Answer

Elapsed Time (seconds) 0

Explanation The ClassPersister interface provides ORM strategies.

Question 102 (ID: # 56469)

Subject Java Technologies: Hibernate, Spring, and Struts

Subtopic Hibernate

Description Callback Interface

Which of the following interfaces is used to receive a notification when an object event occurs?

A.	Extension
B.	Callback
C.	Acknowledgement
D.	Base
E.	Serializable

Correct Answer	B
User Answer	
Elapsed Time (seconds)	0
Explanation	The Callback interface is used to receive a notification of object events, such as when an object is loaded, saved , or deleted.

Question 103 (ID: # 56472)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Core Interfaces

Which of the following options are core interfaces of the Hibernate framework?

A.	Session
B.	SessionFactory
C.	Management
D.	Transaction
E.	Quality

Correct Answer	A ;B ;D ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Options A, B, and D are correct. The following are also core interfaces: Configuration , Query , and Criteria . These core interfaces are used in almost every Hibernate application. Using these interfaces, you can store and recover persistent objects and control transactions.

Question 104 (ID: # 56473)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Business and Persistence Layers

In a general Hibernate overview, the _________ and ________ layers are connected together by the interceptor.

A.	Business
B.	Persistence
C.	Logical
D.	Service
E.	Presentation

Correct Answer	A ;B ;
User Answer	
Elapsed Time (seconds)	0
Explanation	The Business layer and the Persistence layer are connected together by the interceptor in a general overview of Hibernate.

Question 105 (ID: # 56474)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Interceptor Interface

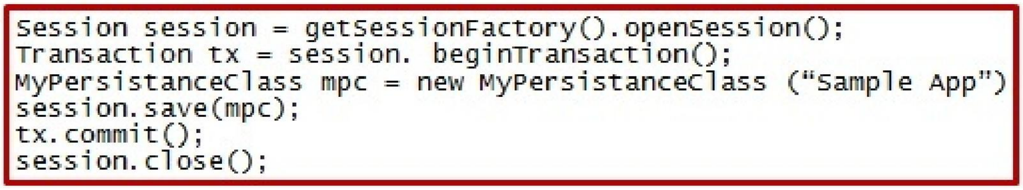
Which of the following statements is true?

A. Implementing the Interceptor interface is better than extending the EmptyInterceptor class.
B. Extending the EmptyInterceptor class is better than implementing the Interceptor interface.
C. A session should be invoked from a callback.
D. The onSave method returns an int value representing the error code.
E. The setValue method returns an int value representing the error code.

Correct Answer	B
User Answer	
Elapsed Time (seconds)	0
Explanation	To generate an interceptor, you need to implement the Interceptor interface. You could do this directly, but it is better to extend the EmptyInterceptor , so that you can just implement the methods you need.

Question 106 (ID: # 56477)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Session and Transaction Interfaces

In the Hibernate code below, what are `Session` and `Transaction`?



```
Session session = getSessionFactory().openSession();
Transaction tx = session. beginTransaction();
MyPersistanceClass mpc = new MyPersistanceClass (“Sample App")
session.save(mpc);
tx.commit();
session.close();
```

A. They are both classes of Hibernate.
B. They are interfaces provided by Hibernate.
C. Session is a class and Transaction is an interface.
D. They are both objects.
E. They are both relations.

Correct Answer	B

User Answer	
Elapsed Time (seconds)	0
Explanation	Session and Transaction are interfaces provided by Hibernate. There are many other interfaces, in addition to these.

Question 107 (ID: # 56478)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Listeners

Listeners registered declaratively cannot share instances, so if the same class name is employed in several `<listener/>` elements, every reference will result in a separate instance of that class.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	Listeners registered declaratively cannot share instances. If the same class name is employed in several elements, every reference will result in a separate instance of that class. If you need to share listener instances between listener types, use the programmatic registration approach.

Question 108 (ID: # 56479)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Full Object Mapping

In ______________________ efficient fetching and caching strategies are implemented transparently.

A. Medium Object Mapping

B. Pure Object Mapping

C. Light Object Mapping

D. Non-pure Object Mapping

E. Full Object Mapping.

Correct Answer	E
User Answer	
Elapsed Time (seconds)	0
Explanation	Competent fetching and caching strategies are implemented transparently in Full Object Mapping.

Question 109 (ID: # 56482)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate

Description Object Mapping

In an ORM framework, ________________________________.

A. SQL code is generated at compile time

B. SQL code is generated at run time by a framework

C. SQL code is never generated

D. source code is generated

E. stored procedures are generated

Correct Answer B

User Answer

Elapsed Time (seconds) 0

Explanation The application is designed around an Object model. SQL is created at build time using a code generation tool or at runtime by framework code. Associations between objects are handled by the Persistence mechanism, and queries can be stated using an object-oriented Expression language. Objects are cached by the Persistence layer. A great many ORM products and homegrown Persistence layers handle at least this level of functionality. It's appropriate for medium-sized applications with some difficult transactions, mainly when portability between different database products is significant. These applications typically don't use stored procedures.

Question 110 (ID: # 56483)

Subject Java Technologies: Hibernate, Spring, and Struts

Subtopic Hibernate

Description Data Models

In ____________________, the entities are represented as classes that are mapped manually to the relational table.

A. Light Object Mapping

B. Heavy Object Mapping

C. Pure Relational Mapping

D. Full Object Mapping

E. Light and Full Object Mapping

Correct Answer A

User Answer

Elapsed Time (seconds) 0

Explanation This strategy is successful for applications that have a small number of entities, or for applications with common, metadata-driven data models. The code is concealed from the business logic, using particular design patterns.

Question 111 (ID: # 56484)

Subject Java Technologies: Hibernate, Spring, and Struts

Subtopic Hibernate

Description Packaging in Hibernate

Which of the following packages implements backward compatibility with Hibernate 2.1 APIs, now deprecated in Hibernate 3?

A. org.hibernate.cfg

B. org.hibernate.criterion

C. org.hibernate.classic

D. org.hibernate.metadata

E. com.hibernate

Correct Answer C

User Answer

Elapsed Time (seconds) 0

Explanation Option C is correct. This is a core API of Hibernate. There are 3 interfaces: Lifecycle , Session , and Validatable .

Question 112 (ID: # 56487)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	ORM Quality Levels

Which of the following are levels of ORM quality?

A. Light Object Mapping

B. Heavy Object Mapping

C. Medium Object Mapping

D. Pure Relational Mapping

E. Full Object Mapping

Correct Answer A ;C ;D ;E ;

User Answer

Elapsed Time (seconds) 0

Explanation Options A, C, D, and E are the four levels of ORM quality.

Question 113 (ID: # 56488)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	ORM Solution

Which of the following statements about ORM Solution are true?

A. It should have an API for executing basic CRUD (Create, Read, Update, Delete) operations on objects of Persistent classes.

B. It should have a language or an API for specifying queries that denote the classes and the properties of classes.

C. It has the ability to specify Mapping metadata.

D. It should have a technique for ORM implementation to relate with transactional objects to carry out dirty checking, lazy association fetching, and other optimization functions.

E. It is a programming technique for converting data between web services.

Correct Answer A ;B ;C ;D ;

User Answer

Elapsed Time (seconds) 0

Explanation Options A through D are correct. Object-Relational Mapping (ORM) represents the notion of mapping an application's business objects to relational database tables, so that data may be accessed and updated completely through the object model of an application.

Question 114 (ID: # 56489)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	ORM

ORM stands for _________________________.

A. Object Relational Modeling

B. Object Relational Mapping

C. Object Related Model

D. Object Rate Mapping

E. Object Relational Mining

Correct Answer B

User Answer

Elapsed Time (seconds) 0

Explanation ORM stands for Object Relational Mapping. It is the programmed and transparent perseverance of objects in a Java application into the tables of a relational database employing the metadata that illustrates the mapping between the objects and the database. It transforms the data from one representation to another.

Question 115 (ID: # 56492)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Hibernate Configuration File

Which of the following are configuration parameters for the `hibernate.cfg.xml` file?

A. Database URL

B. Username/Password

C. Ant Version

D. Destination Path

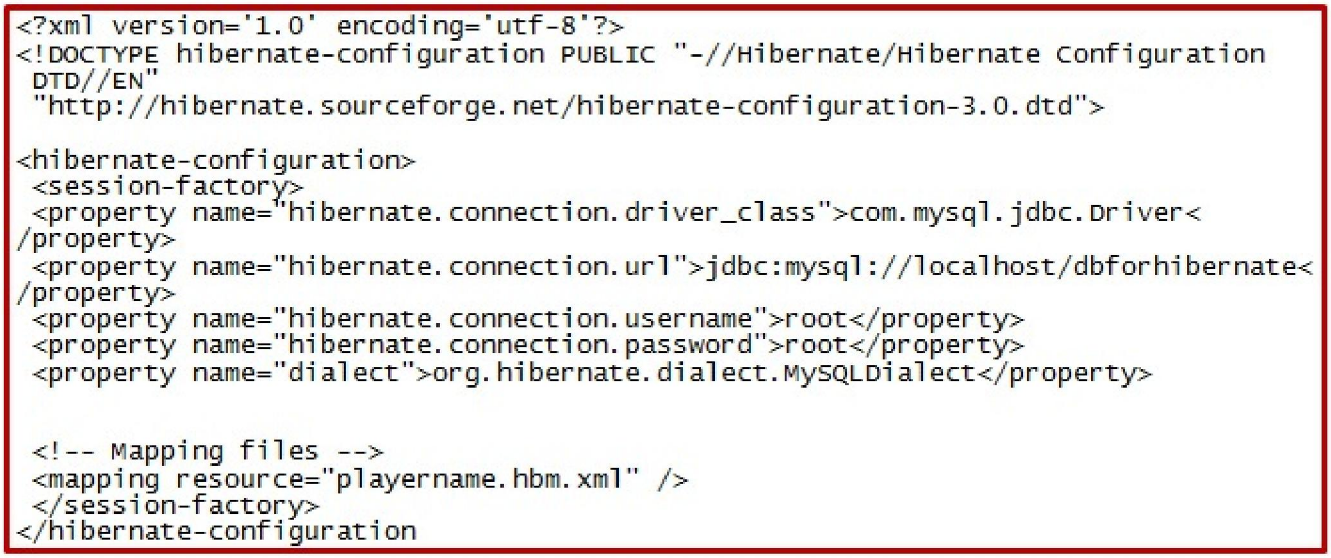
E. Classpath

Correct Answer A ;B ;

User Answer

Elapsed Time (seconds) 0

Explanation The configuration file for the sample application is shown below. It takes the Database URL and the Username/ Password.



```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration
 DTD//EN"
 "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
 <session-factory>
 <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver<
/property>
 <property name="hibernate.connection.url">jdbc:mysql://localhost/dbforhibernate<
/property>
 <property name="hibernate.connection.username">root</property>
 <property name="hibernate.connection.password">root</property>
 <property name="dialect">org.hibernate.dialect.MySQLDialect</property>


 <!-- Mapping files -->
 <mapping resource="playername.hbm.xml" />
 </session-factory>
</hibernate-configuration
```

Question 116 (ID: # 56493)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Validatable Interface

In which of the following situations is the `validate()` method of the `Validatable` interface called?

A. Session.save()

B. Session.update()

C. Session.saveOrUpdate()

D. Session.flush()

E. Session.change()

Correct Answer A ;B ;C ;D ;

User Answer

Elapsed Time (seconds) 0

Explanation Options A, B, C, and D are correct. The Validatable.validate() method will be called by the framework during a Save operation. A Save operation can happen whenever the Session.save() , Session.update() , Session. saveOrUpdate() , or Session.flush() methods are invoked.

Question 117 (ID: # 56494)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Interceptors

Which of the following are interceptors in Hibernate?

A. Request-scoped

B. Bean-Scoped

C. Session-scoped

D.	Application-scoped
E.	Parameter-scoped

Correct Answer	C ;D ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Interceptors in Hibernate can fall under two categories: Application-scoped Interceptors and Session-scoped Interceptors .

Question 118 (ID: # 56497)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Inheritance Hierarchy

Which of the following are the different approaches to Inheritance Hierarchy?

A.	table per concrete class
B.	table per class hierarchy
C.	table per subclass
D.	table per inheritance
E.	table per object

Correct Answer	A ;B ;C ;
User Answer	
Elapsed Time (seconds)	0
Explanation	To deal with the problems found in hierarchy persistence, Hibernate supports three special inheritance persistence strategies: table per class hierarchy, table per subclass, and table per concrete class. Each mapping strategy is incrementally more complex.

Question 119 (ID: # 147822)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	HQL queries

Which of the following aggregate functions are supported by HQL queries?

A.	count(*)
B.	total(...)
C.	count(distinct ...)
D.	min(...)
E.	avg(...)

Correct Answer	A ;C ;D ;E ;
User Answer	
Elapsed Time (seconds)	0

Explanation	Options A, C, D, and E are correct. HQL queries may even return the results of aggregate functions on properties. The following aggregate functions are supported. - avg(...), sum(...), min(...), max(...) - count(*) - count(...), count(distinct ...), count(all...)

Question 120 (ID: # 147823)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Session.isDefaultReadOnly()

If `Session.isDefaultReadOnly()` returns `true`, entities loaded by _____________ are automatically made read-only.

A.	Session.load()
B.	Session.persist()
C.	Session.refresh()
D.	Session.get()
E.	Session.merge()

Correct Answer	A ;D ;E ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Options A, D, and E are correct. You can establish the current setting by calling Session.isDefaultReadOnly() . If Session.isDefaultReadOnly() returns true , entities loaded by the following are automatically made read-only. - Session.load() - Session.get() - Session.merge() - executing, scrolling, or iterating HQL queries and criteria

Question 121 (ID: # 147824)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Hibernate
Description	Persistent Entities

Only __________ entities can be made read-only.

A.	deprecated
B.	deleted
C.	persistent
D.	transient
E.	detached

Correct Answer	C
User Answer	
Elapsed Time (seconds)	0

Explanation	Only persistent entities can be made read-only. Transient and detached entities must be put into a persistent state before they can be made read-only.

Question 122 (ID: # 147827)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	tx:method

Which of the following attributes of the `<tx:method/>` tags that are nested inside `<tx:advice/>` and `<tx:attributes/>` tags are not required?

A. isolation

B. timeout

C. rollback-for

D. name

E. propagation

Correct Answer	A ;B ;C ;E ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Options A, B, C, and E are correct. The name attribute is the only one that is required. It represents the method name(s) the transaction attributes are to be linked to. The wildcard (*) character may be used to associate the same transaction attribute settings with a number of methods, such as get* , handle* , on*Event , etc.

Question 123 (ID: # 147828)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	beanNamesForTypeIncludingAncestors

In Spring 2.0, the `beanNamesForTypeIncludingAncestors` method automatically includes prototypes and `FactoryBeans`.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	The statement is true. This method retrieves all bean names for the supplied type, as well as those defined in ancestor factories. It will return unique names in case of overridden bean definitions.

Question 124 (ID: # 147829)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	WebServiceMessageReceiver

Which of the following are all known implementing classes of the `WebServiceMessageReceiver` interface?

A. MailReceiverConnection

B. MessageDispatcher

C. JmsReceiverConnection

D. HttpUrlConnection

E. SoapMessageDispatcher

Correct Answer A ;E ;

User Answer

Elapsed Time (seconds) 0

Explanation Options A and E are correct. WebServiceMessageReceiver is the essential dispatcher used within Spring-WS, dispatching Web service messages to registered endpoints. And SoapMessageDispatcher is a SOAP-specific subclass of the MessageDispatcher .

Question 125 (ID: # 147832)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	NestedTransactionNotSupportedException

The `NestedTransactionNotSupportedException` class extends ______________________.

A. SimpleTransactionStatus

B. AbstractPlatformTransactionManager

C. CannotCreateTransactionException

D. TransactionUsageException

E. TransactionException

Correct Answer C

User Answer

Elapsed Time (seconds) 0

Explanation This exception can be thrown when nested transactions are not supported by the underlying backend.

Question 126 (ID: # 56498)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	TransactionDefinition Interface

Which of the following does the `TransactionDefinition interface` specify?

A. isolation

B. global or local transactions

C. timeout

D. read-only status

E. write-only status

Correct Answer A ;C ;D ;

User Answer	
Elapsed Time (seconds)	0
Explanation	The TransactionDefinition specifies information about the following. - Isolation specifies the degree of isolation a given transaction has from the work of other transactions. For instance, this indicates whether a transaction may see uncommitted writes from other transactions. - Propagation specifies whether all code executed inside a transaction scope will run in that transaction. There are numerous choices stating behavior if a transactional method is executed when a transaction context already exists. For instance, continuing to run in the existing transaction (the common case), or suspending the existing transaction and generating a new transaction. Spring gives the entire transaction propagation choices familiar from EJB CMT. - Timeout specifies how long this transaction might run before timing out (and automatically being rolled back by the underlying transaction infrastructure). - Read-only status is provided, since a read-only transaction does not change any data. Read-only transactions can be a helpful optimization in some cases (for instance when you are using Hibernate).

Question 127 (ID: # 56499)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Optimistic Locking Strategy

In an Optimistic Locking Strategy, the application is responsible for checking whether the data to be updated has been changed in another transaction, since it was read from the database.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	The statement is true. This is the concept of optimistic locking.

Question 128 (ID: # 56502)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	MessageDriven Beans (MDBs)

Which abstract class is used to develop MDBs (Message Driven Beans) to accept sources other than JMS?

A.	AbstractMessageDrivenBean
B.	AbstractJmsMessageDrivenBean
C.	AbstractStatefulSessionBean
D.	AbstractBean
E.	AbstractSessionBean

Correct Answer	A
User Answer	
Elapsed Time (seconds)	0

Explanation	The AbstractMessageDrivenBean is used to develop MDBs that accept sources other than JMS

Question 129 (ID: # 56503)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Transaction Manager Attribute

You can omit the `transaction-manager` attribute in the transactional advice (`<tx:advice/>`) if the bean name of the `PlatformTransactionManager` you want to wire in has the name `txManager`.

Correct Answer	FALSE
User Answer	
Elapsed Time (seconds)	0
Explanation	The statement is false. You can omit the transaction-manager attribute if the bean name is transactionManager .

Question 130 (ID: # 56504)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	TransactionTemplate

You cannot use a `TransactionTemplate` without defining a `PlatformTransactionManager` in your Spring context.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	A TransactionTemplate needs a PlatformTransactionManager . If you don't pass it as an argument to its constructor, you have to call its setter method. The TransactionTemplate 's afterPropertiesSet() will check whether the PlatformTransactionManager is not set. If it is not, it will throw an IllegalArgumentException .

Question 131 (ID: # 56507)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	DataSource

When a `DataSource` is not present, you need a connection pooling bean that implements `dataSource`.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0

Explanation	The statement is true. You can use dbcp.BasicDataSource for this. By using this, you can have a dataSource with connection pooling independent of the application server.

Question 132 (ID: # 56508)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Application Context

Which of the following statements about Application Context are true?

A.	It provides a means for message resource handling, including support for internationalization.
B.	It is a generic way to load file resources.
C.	It provides events to beans that are registered as listeners.
D.	It is an html file.
E.	It is represented by the ContextAware interface.

Correct Answer	A ;B ;C ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Application Context is Spring's more advanced container. Like BeanFactory , it can be used to load bean definitions, wire beans together, and dispense beans upon request. Because of the additional functionality, in most cases Application Context is preferred over BeanFactory . In some environments, such as mobile environments where the application's footprint must be minimal, BeanFactory should be used.

Question 133 (ID: # 56509)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Spring Testing

For testing the Spring application, Static block or Spring classes are used in Spring Mock from __________________.

A.	org.springframework.test
B.	org.springframework.testing
C.	org.springframework.tests
D.	com.springFramework
E.	org.springframework.transaction

Correct Answer	A
User Answer	
Elapsed Time (seconds)	0
Explanation	org.springframework.test provides Spring classes for testing the Spring application.

Question 134 (ID: # 56512)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	DAO and DataSource

Which of the following statements about DAO and `DataSource` are true?

A. DAO has a Setter or a Constructor argument for the DataSource .
B. DataSource has a Setter or a Constructor argument for the DAO.
C. DataSource is configured as a dependency of the DAO.
D. DAO is configured as a dependency of the DataSource .
E. DataSource cannot be configured.

Correct Answer A ;C ;

User Answer

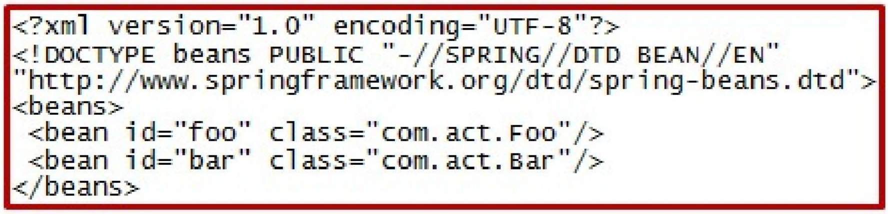
Elapsed Time (seconds) 0

Explanation DAO and DataSource make it easy to swap out implementations. Tests can verify whether operations succeed on the interface.

Question 135 (ID: # 56513)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Adding a Bean to the Spring Framework

A bean can be added to the Spring framework, using the following code for Spring 1.



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">
<beans>
 <bean id="foo" class="com.act.Foo"/>
 <bean id="bar" class="com.act.Bar"/>
</beans>
```

Correct Answer TRUE

User Answer

Elapsed Time (seconds) 0

Explanation In the Bean tag, the ID attribute specifies the Bean name, and the Class attribute specifies the fully qualified Class name.

Question 136 (ID: # 56514)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Bean Wiring

Combining beans within the Spring container is known as __________.

A. Bean Wiring

B.	Bean Concatenation
C.	Bean Combination
D.	Bean Container
E.	Bean Coupling

Correct Answer	A
User Answer	
Elapsed Time (seconds)	0
Explanation	When wiring beans, you should tell the container what beans are needed and how the container should use Dependency Injection to tie them together.

Question 137 (ID: # 56517)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Beans

If any `BeanPostProcessors` are associated with a bean, their `postProcessBeforeInitialization()` and `postProcessAfterInitialization()` methods will be called.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	BeanPostProcessors are part of the typical bean life cycle. -If there are any BeanPostProcessors associated with the bean, their postProcessBeforeInitialization() methods will be called. -If an init -method is specified for the bean, it will be called. -Finally, if there are any BeanPostProcessors associated with the bean, their postProcessAfterInitialization() methods will be called.

Question 138 (ID: # 56518)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	BeanFactory

If a bean implements the `BeanFactoryAware` interface, and the factory calls `setBeanFactory()`, what is passed to this method?

A.	the bean ID
B.	an instance of the factory itself
C.	the bean name
D.	a number of beans
E.	the Bean location path

Correct Answer	B
User Answer	
Elapsed Time	0

(seconds)

Explanation If the bean implements the BeanFactoryAware interface, the factory calls setBeanFactory() , passing an instance of itself.
The bean ID is passed if the bean implements the BeanNameAware interface and the factory calls setBeanName () .

Question 139 (ID: # 56519)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Spring Framework

Which of the following are important `ApplicationContext` implementations in the Spring framework?

A. ClassPathXmlApplicationContext
B. FileSystemXmlApplicationContext
C. XmlWebApplicationContext
D. Class
E. Object

Correct Answer A ;B ;C ;

User Answer

Elapsed Time (seconds) 0

Explanation Options A, B, and C are correct.
- ClassPathXmlApplicationContext loads a context definition from an XML file located in the class path, treating context definition files as class path resources.
- FileSystemXmlApplicationContext loads a context definition from an XML file in the file system.
- XmlWebApplicationContext loads the context definitions from an XML file contained within a Web application.

Question 140 (ID: # 56522)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Bean Factory

To create `XmlBeanFactory`, what is passed to the constructor?

A. the OutputStream
B. the InputStream
C. an object of BeanFactory
D. an object of the class
E. the name of the bean

Correct Answer B

User Answer

Elapsed Time (seconds) 0

Explanation	The InputStream will provide the XML to the factory.

Question 141 (ID: # 56523)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Transaction Management

For a small number of transactional operations, which transaction management option is best?

A.	Declarative Transaction Management
B.	Programmatic Transaction Management
C.	Less Operation Programmatic Transaction Management
D.	Serializable Transaction Management
E.	Context Transaction Management

Correct Answer	B
User Answer	
Elapsed Time (seconds)	0
Explanation	Programmatic Transaction Management is used, for example, if you have a Web application that requires transactions for only certain update operations.

Question 142 (ID: # 56524)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	<code>@Transactional</code> Annotation

The `@Transactional` annotation can only be successfully applied to methods with public visibility.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	If you do annotate protected, private, or package-visible methods with the @Transactional annotation, no error is raised, but the annotated method will not display the configured transactional settings.

Question 143 (ID: # 56527)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	<code>TransactionDefinition</code> Interface

Which of the following Spring-compliant transaction properties does the `TransactionDefinition` interface define?

A.	Isolation
B.	Savepoints

C.	Propagation
D.	Timeout
E.	Write-only Status

Correct Answer A ;C ;D ;

User Answer

Elapsed Time (seconds) 0

Explanation Options A, C, and D are correct.
- Isolation represents the degree of isolation this transaction has from the work of other transactions.
- Propagation choices given by Spring are the same as those transaction propagation choices in EJB CMT.
- Timeout states how long this transaction might run before it times out.
- Read-only Status does not change any data.

Question 144 (ID: # 56528)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	GetTransaction Method

Which of the following options specifies the parameter and return type for the `PlatformTransactionManager.getTransaction()` method?

A.	Parameter: NumberOfTransactions and Return: transactionName
B.	Parameter: TransactionDefinition and Return: TransactionStatus
C.	Parameter: TransactionDefinition and Return: numberOfTransactions
D.	The getTransaction() method doesn't take any parameters, but it returns the number of transactions.
E.	The getTransaction() method doesn't take any parameters, but it returns void .

Correct Answer B

User Answer

Elapsed Time (seconds) 0

Explanation The getTransaction() method returns the currently active transaction, or it generates a new one in relation to the stated propagation behavior. Parameters such as isolation level or timeout will just be applied to new transactions, not the active ones.

Question 145 (ID: # 56529)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Transaction Management

Which of the following are valid choices for Transaction Management?

A.	Local Transactions
B.	Isolation Transactions
C.	Atomic Transactions
D.	Global Transactions
E.	Managed Transactions

Correct Answer A ;D ;

User Answer

Elapsed Time (seconds) 0

Explanation There are two choices for Transaction Management: Local Transactions and Global Transactions . A global transaction coordinates numerous types of services. These services might be placed in different programs on the server. Global transactions are also called distributed transactions. A local transaction is one in which each resource manager coordinates its own modifications.

Question 146 (ID: # 56532)

Subject Java Technologies: Hibernate, Spring, and Struts

Subtopic Spring

Description IoC

Which packages provide the basis for Spring framework's IoC container?

A. org.springframework.beans

B. org.springframework.context

C. org.springframework.util

D. org.springframework.scripting

E. com.springframework

Correct Answer A ;B ;

User Answer

Elapsed Time (seconds) 0

Explanation The BeanFactory interface offers an advanced configuration mechanism capable of managing objects of any nature.
The ApplicationContext interface builds on top of Bean Factory.

Question 147 (ID: # 56533)

Subject Java Technologies: Hibernate, Spring, and Struts

Subtopic Spring

Description Core Package

Which jar files make the core package that is the central part of Spring's Web services functionality?

A. spring-ws-core.jar

B. spring-ws-core-tiger.jar

C. spring.webservice

D. spring-xml.jar

E. spring-ws-security.jar

Correct Answer A ;B ;

User Answer

Elapsed Time (seconds) 0

Explanation	Both spring-ws-core.jar and spring-ws-core-tiger.jar generate the core package that is the central part of Spring's Web services functionality.

Question 148 (ID: # 56534)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Context File

You can have more than one `aop:config` tag in a context file.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	The aop:config tag nicely groups all AOP-related functionality together in your context file.

Question 149 (ID: # 56537)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	BeanFactory

Which of the following is used to implement `BeanFactory` in Spring?

A. org.springframework.beans.factory.xml.XmlBeanFactory

B. org.springframework.beans.factory.xml.BeanFactory

C. org.springframework.factory.xml.XmlBeanFactory

D. org.apache.spring framework.beans.factory

E. com.apache.framework.beans

Correct Answer	A
User Answer	
Elapsed Time (seconds)	0
Explanation	One of the most useful implementations of BeanFactory is org.springframework.beans.factory.xml. XmlBeanFactory , which loads its beans based on the definitions contained in an XML file.

Question 150 (ID: # 56538)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Spring Configuration File

The Spring configuration file ______________________.

A. is an XML file

B. contains bean information

C. contains interface information

D. contains all information about the Spring framework

E. contains methods information

Correct Answer A ;B ;

User Answer

Elapsed Time (seconds) 0

Explanation The Spring configuration file is an XML file that contains information about how the classes are configured and introduced to one another.

Question 151 (ID: # 56539)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Web Module

Which of the following modules contains integration support with Jakarta Struts?

A. O/R Module

B. AOP Module

C. Web Module

D. Application Context Module

E. Client Module

Correct Answer C

User Answer

Elapsed Time (seconds) 0

Explanation The Web module also includes support for some Web-oriented tasks, for instance transparently controlling multipart requests for file uploads as well as programmatic binding request parameters to your business objects.

Question 152 (ID: # 56542)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	JDBC Module

JDBC abstraction and the DAO module employ Spring's AOP module to offer Transaction Management services for objects in a Spring application.

Correct Answer TRUE

User Answer

Elapsed Time (seconds) 0

Explanation Using this module, you can keep the database code clean and simple, and avoid problems that might result from a malfunction to close database resources.

Question 153 (ID: # 56543)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	AOP Module

The AOP module ___________________________________.

A.	provides rich support for aspect-oriented programming
B.	introduces Object-Oriented programming
C.	introduces Entity-Relationship programming
D.	introduces types of applications
E.	introduces procedural programming

Correct Answer	A
User Answer	
Elapsed Time (seconds)	0
Explanation	This module serves as the foundation for developing your own features for your Spring-enabled application. To guarantee interoperability between Spring and other AOP frameworks, much of Spring's AOP support is founded on the API defined by the AOP Alliance. The AOP Alliance is an open-source project whose objective is to encourage adoption of AOP and interoperability among diverse AOP implementations, by defining a common set of interfaces and components.

Question 154 (ID: # 56544)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Modules

Which of the following modules supplies many Enterprise services such as JNDI access, EJB integration, remoting, and scheduling?

A.	Object/Relational Module
B.	Web Module
C.	JDBC and DAO Modules
D.	Aspect Oriented Programming Module
E.	Application Context Module

Correct Answer	E
User Answer	
Elapsed Time (seconds)	0
Explanation	The Application Context Module extends the concept of BeanFactory . It also provides support to other frameworks.

Question 155 (ID: # 56547)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Spring Container

Which of the following is the center of any Spring-based application that makes the Spring Container?

A.	SessionFactory
B.	BeanFactory
C.	JNDI
D.	Request
E.	SQLQuery

Correct Answer	B
User Answer	
Elapsed Time (seconds)	0
Explanation	In the core container module, BeanFactory is the center of any Spring-based application. The entire framework was built on top of this module.

Question 156 (ID: # 56548)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Spring Framework

The MVC Framework module is a part of the Spring Framework.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	The Spring Framework modules are listed below. - Core Container Module - Application Context Module - AOP Module (Aspect Oriented Programming) - JDBC Abstraction and DAO Module - O/R Mapping Integration Module (Object/Relational) - Web Module - MVC Framework Module

Question 157 (ID: # 56549)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	About Spring

Which of the following statements are not true about the Spring Framework?

A.	It is lightweight.
B.	It offers Inversion of Control.
C.	It is aspect-oriented.
D.	It includes a container and a framework.
E.	It has an object-relational mapping library.

Correct Answer	E
User Answer	
Elapsed Time (seconds)	0
Explanation	The Spring Framework has the following attributes. - Lightweight - Spring is lightweight when it comes to size and transparency. The basic version of the Spring framework is around 1MB. The processing overhead is also very small. - Inversion of Control (IoC) - Loose coupling is accomplished in Spring using the Inversion of Control technique. The objects offer their dependencies instead of generating or searching for dependent objects. - Aspect-oriented (AOP) - Spring handles Aspect-oriented programming and enables cohesive development by dividing application business logic from system services. - Container - Spring includes and administers the life cycle and configuration of application objects. - Framework - Spring offers most of the intra-functionality, leaving the rest of the coding to the developer.

Question 158 (ID: # 56552)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	RowCallbackHandler

Which of the following methods needs to be implemented in the `RowCallbackHandler` interface, so that it is applicable for each and every row?

A.	processRow()
B.	getRow()
C.	callbackRow()
D.	processCallbackRow()
E.	addRow()

Correct Answer	A
User Answer	
Elapsed Time (seconds)	0
Explanation	processRow() needs to be implemented so that it is applicable for each and every row. The signature for the method is void processRow(java.sql.ResultSet rs);

Question 159 (ID: # 56553)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	BatchPreparedStatementSetter

`getBatchSize()` and `setValues()` are the methods provided by the ____________________ interface.

A.	PreparedStatementCreator
B.	SqlProvider
C.	RowCallbackHandler
D.	BatchPreparedStatementSetter
E.	BatchProcessor

Correct Answer	D
User Answer	
Elapsed Time (seconds)	0
Explanation	getBatchSize() tells the JdbcTemplate class how many statements to create. This also determines how many times setValues() will be called.

Question 160 (ID: # 56554)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	PreparedStatementCreator Interface

When the `PreparedStatementCreator` interface is implemented, which other interface (that has a method called `getSql()`) is also implemented?

A. BatchPreparedStatementSetter

B. RowCallbackHandler

C. SqlProvider

D. getSqlInterface

E. SqlReturnType

Correct Answer	C
User Answer	
Elapsed Time (seconds)	0
Explanation	SqlProvider is the only other interface that is implemented when the PreparedStatementCreator interface is implemented. It has a method called getSql() , which is used to offer sql strings to JdbcTemplate .

Question 161 (ID: # 56557)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Transaction Managers

Which of the following Transaction Managers were available in Spring 2?

A. JTATransactionManager

B. DataSourceTransactionManager

C. HibernateTransactionManager

D. JdoTransactionManager

E. DSTransactionManager

Correct Answer	A ;B ;C ;D ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Options A through D are correct: org.springframework.transaction.jta.JtaTransactionManager , org. springframework.jdbc.datasource.DataSourceTransactionManager , org.springframework.orm.hibernate.

HibernateTransactionManager , and
org.springframework.orm.jdo.JdoTransactionManager .

Question 162 (ID: # 56558)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Exception

Which of the following is the Exception class related to all of the exceptions that are thrown in Spring applications?

A.	SQLException
B.	ArithmeticException
C.	DataAccessException
D.	FileNotFoundException
E.	Exception

Correct Answer	C
User Answer	
Elapsed Time (seconds)	0
Explanation	DataAccessException is a RuntimeException . This is an Unchecked Exception . The user is not forced to handle these kinds of exceptions. The package structure is org.springframework.dao.DataAccessException .

Question 163 (ID: # 56559)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Around Advice Type

The package for the Around advice type is `org.appliance.intercept.MethodInterceptor`.

Correct Answer	TRUE
User Answer	
Elapsed Time (seconds)	0
Explanation	The statement is true. The different advice types have the following packages. - Around: org.appliance.intercept.MethodInterceptor - Before: org.springframework.aop.BeforeAdvice - After: org.springframework.aop.AfterReturningAdvice - Throws: org.springframework.aop.ThrowsAdvice

Question 164 (ID: # 56562)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Advice Types

What are the different advice types in Spring?

A. Around

B. Before

C. After

D. Throws

E. Finally

Correct Answer A ;B ;C ;D ;

User Answer

Elapsed Time (seconds) 0

Explanation Options A through D are correct. The different advice types for Spring are listed below.
- The Around advice type intercepts the calls to the Target method.
- The Before advice type is called before the Target method is invoked.
- The After advice type is called after the Target method is returned.
- The Throws advice type is called when the Target method throws an exception.

Question 165 (ID: # 56563)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Weaving

Which of the following are times when `Weaving` can be applied?

A. Compile time

B. Runtime

C. ClassLoad time

D. Development time

E. Construction time

Correct Answer A ;B ;C ;

User Answer

Elapsed Time (seconds) 0

Explanation Weaving can be applied at Compile time, Runtime, or Classload time.

Question 166 (ID: # 56564)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Weaving

The process of applying aspects to a `Target` object to create a new `Proxy` object is called `Weaving`.

Correct Answer TRUE

User Answer

Elapsed Time (seconds) 0

Explanation The statement is true. The aspects are woven into the Target object at the specified joinpoints.

Question 167 (ID: # 56567)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Advice

Advice is ______________________________________.

A. a cross-cutting functionality that is being implemented

B. the implementation of an aspect

C. the class that is being advised

D. the process of applying aspects to a Target object

E. the process of forcing a class to perform certain operations

Correct Answer B

User Answer

Elapsed Time (seconds) 0

Explanation Advice is the implementation of an aspect, which explains the new behavior to the application. Normally, an advice is placed into an application at joinpoints.

Question 168 (ID: # 56568)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Joinpoint

A joinpoint is a point in the execution of the application where a(n) __________ can be plugged in.

A. advice

B. aspect

C. pointcut

D. class

E. interface

Correct Answer B

User Answer

Elapsed Time (seconds) 0

Explanation A joinpoint is a point in the implementation of the application where an aspect can be plugged in. This might be a method being called, an exception being thrown, or even a field being altered. These are the points at which your aspect's code can be inserted into the ordinary flow of your application to add a new behavior.

Question 169 (ID: # 56569)

Subject	Java Technologies: Hibernate, Spring, and Struts

Subtopic	Spring
Description	Autowire Types

Which of the following are `Autowire` types?

A.	byName
B.	byType
C.	constructor
D.	autodetect
E.	byValue

Correct Answer	A ;B ;C ;D ;
User Answer	
Elapsed Time (seconds)	0
Explanation	Options A through D are correct. byValue is not an Autowire type.

Question 170 (ID: # 56572)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Auto Wiring

A bean can be auto-wired by setting the `Autowire` attribute of the Bean tag to "True."

Correct Answer	FALSE
User Answer	
Elapsed Time (seconds)	0
Explanation	The statement is false. You can wire beans manually, but Spring Framework also achieves this for you. It can also auto-wire related beans together. All you need to do is set the Autowire attribute of the bean tag to an Autowire type.

Question 171 (ID: # 56573)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Bean Injection

Which of the following are the two types of bean injections?

A.	By Getter
B.	By Setter
C.	By Constructor
D.	By Method
E.	By Interface

Correct Answer	B ;C ;
User Answer	

Elapsed Time (seconds)	0
Explanation	Options B and C are correct. The two types of bean injections are By Setter and By Constructor.

Question 172 (ID: # 56574)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Inner Beans

An Inner Bean is ____.

A.	a bean that is declared within another in a nesting of beans when there is nesting of beans
B.	a bean element that is embedded in a property tag directly
C.	a bean that is declared by Inner
D.	a bean element that can be used within many other beans
E.	There is no such thing as an Inner Bean.

Correct Answer	B
User Answer	
Elapsed Time (seconds)	0
Explanation	The drawback of an Inner Bean is that it cannot be reused anywhere else.

Question 173 (ID: # 56577)

Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	Bean Lifecycle Methods

When a bean is unloaded from a container, the __________ method is called.

A.	unload
B.	unloadBean
C.	teardown
D.	freeBean
E.	stop

Correct Answer	C
User Answer	
Elapsed Time (seconds)	0
Explanation	There are two significant bean lifecycle methods. The initial one is setup , which is called when the bean is loaded into the container. The next method is teardown , which is called when the bean is unloaded from the container.

Question 174 (ID: # 56578)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	WebServiceMessageSender Interface

Which of the following are implementations for the `WebServiceMessageSender` interface in version 1.5.9 of Spring Web Services?

A.	AbstractHttpWebServiceMessageSender
B.	HttpMessageSender
C.	ConnectionMessageSender
D.	HttpUrlConnectionMessageSender
E.	JmsMessageSender

Correct Answer	A ;D ;E ;
User Answer	
Elapsed Time (seconds)	0
Explanation	There are six implementations for the WebServiceMessageSender interface that is used for sending messages via HTTP. Here are all of the known implementing classes for WebServiceMessageSender interface: AbstractHttpWebServiceMessageSender , CommonsHttpMessageSender , HttpsUrlConnectionMessageSender , HttpUrlConnectionMessageSender , JmsMessageSender , and MailMessageSender .

Question 175 (ID: # 56579)	
Subject	Java Technologies: Hibernate, Spring, and Struts
Subtopic	Spring
Description	WebServiceTemplate

`WebServiceTemplate` is the core class for server-side Web service access in Spring-WS.

Correct Answer	FALSE
User Answer	
Elapsed Time (seconds)	0
Explanation	The statement is false. Spring-WS's server-side support is designed around a MessageDispatcher that ships incoming messages to endpoints. The plainest endpoint is PayloadEndpoint .