OCEEJBD 6 Practice Test

Started on	Monday, 22 October 2018, 9:20 PM
State	Finished
Completed on	Monday, 22 October 2018, 9:21 PM
Time taken	50 secs
Grade	**0.00** out of 60.00 (**0**%)
Result	

QUESTION 1 — NOT ANSWERED — MARK FOR REVIEW — Ask our Experts

Given two interfaces and a session bean in the com.whizlabs package:

```
public interface LocalA {
   public void methodA();
}
public interface LocalB {
   public LocalA getA();
}
@Stateless(name = "Whizlabs")
Local({LocalA.class, LocalB.class})
public class WhizlabsBean implements LocalA, LocalB {
   @Resource
   private SessionContext sessionContext;
   public void methodA() { ... }
   public LocalA getA() {
      // insert-here
      return localA;
   }
   // other declarations
}
```

Which TWO of the following are valid statements, independently, to be put at // insert-here to make the WhizlabsBean.getA method return an instance of LocalA, provided there is no associated deployment descriptor?

Please select :

- A. LocalA localA = sessionContext.getBusinessObject(LocalA.class);
- B. Context initialContext = new InitialContext();
  LocalA localA = (LocalA) initialContext.lookup("Whizlabs!com.whizlabs.LocalA");
- C. Context initialContext = new InitialContext();
  LocalA localA = (LocalA) initialContext.lookup("com.whizlabs.WhizlabsBean!LocalA");
- D. LocalA localA = (LocalA) sessionContext.lookup("java:module/Whizlabs!com.whizlabs.LocalA");
- E. LocalA localA = (LocalA) sessionContext.lookup("Whizlabs!com.whizlabs.LocalA");

Your answer is incorrect.

Answer: A and D

Explanation:

When searching for a resource before it is injected into the component JNDI namespace, the resource must be looked up using full name instead of a relative one. As a result, options B, C and E are all incorrect.

The correct answers are: LocalA localA = sessionContext.getBusinessObject(LocalA.class);, LocalA localA = (LocalA) sessionContext.lookup("java:module/Whizlabs!com.whizlabs.LocalA");

Submit your Feedback/Queries to our Experts

QUESTION 2 — NOT ANSWERED — MARK FOR REVIEW — Ask our Experts

Which of the following is NOT an element of the @EJB annotation?

Please select :

- A. type
- B. beanName
- C. name
- D. lookup
- E. mappedName

Your answer is incorrect.

Answer: A

Explanation:

The @EJB annotation defines the following elements:

- beanInterface: Holds one of the types of the target EJB's client views
- beanName: The ejb-name of the enterprise bean to which this reference is mapped
- description: A string describing the bean
- lookup: A portable lookup string containing the JNDI name for the target EJB component
- mappedName: The product specific name of the EJB component to which this ejb reference should be mapped
- name: The logical name of the ejb reference within the declaring component's environment

References:

http://docs.oracle.com/javaee/6/api/javax/ejb/EJB.html

The correct answer is: type

Submit your Feedback/Queries to our Experts

QUESTION 3 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a session bean in the com.whizlabs package:

```
@Stateless
@LocalBean
@Local(Whizlabs.class)
public WhizlabsBean implements Whizlabs {
   // bean class body
}
```

The WhizlabsBean bean is injected into other beans in the same module:

```
@Stateless
public class BeanA {
   @EJB(name = "whizlabs")
   private WhizlabsBean whizlabsA;
   // other declarations
}
@Stateless
public class BeanB {
   @EJB(beanName = "WhizlabsBean", lookup = "java:module/WhizlabsBean!com.whizlabs.Whizlabs")
   private Whizlabs whizlabsB;
   // other declarations
}
```

Which of the following statements is correct about the whizlabsA and whizlabsB fields, provided there is no deployment descriptor?

Please select :

- A. The same WhizlabsBean instance can be injected into whizlabsA and whizlabsB
- B. Instances of WhizlabsBean injected into whizlabsA and whizlabsB are always different
- C. The declaration of BeanA is invalid
- D. The declaration of BeanB is invalid

Your answer is incorrect.

Answer: D

Explanation:

As per the Java EE 6 API documentation, either the beanName or the lookup element of the @EJB annotation can be used to resolve the EJB dependency to its target session bean component. It is an error to specify values for both beanName and lookup.

References:

http://docs.oracle.com/javaee/6/api/javax/ejb/EJB.html

The correct answer is: The declaration of BeanB is invalid

Submit your Feedback/Queries to our Experts

QUESTION 4 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given two session beans:

```
@Stateless
public class MyBean {
   @EJB(beanName = "beanName", mappedName = "mappedName")
   private Whizlabs whizlabs;
   // other declarations
}
// insert annotation here
public class WhizlabsBean implements Whizlabs {
   // bean class body
}
```

How the WhizlabsBean class should be annotated to make it injectable in MyBean, provided no deployment descriptor is existent?

Please select :

- A. @Stateless(name = "beanName")
- B. @Stateless(name = "mappedName")
- C. @Stateless(mappedName = "mappedName")
- D. @Stateless(mappedName = "beanName")
- E. @Stateless

Your answer is incorrect.

Answer: A

Explanation:

The beanName element of the @EJB annotation is the ejb-name of the enterprise bean to which this reference is mapped. Only the annotation in option A specifies a matching name for WhizlabsBean.

References:

http://docs.oracle.com/javaee/6/api/javax/ejb/EJB.html

http://docs.oracle.com/javaee/6/api/javax/ejb/Stateless.html

The correct answer is: @Stateless(name = "beanName")

Submit your Feedback/Queries to our Experts

QUESTION 5 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following statements is correct about the invocation of an asynchronous business method of an enterprise session bean?

Please select :

- A. The transaction context of the client, if any, propagates with the method invocation
- B. The security principal of the client propagates with the method invocation
- C. If the execution of the asynchronous method results in an exception, this exception is hidden from the client
- D. If the asynchronous method has the return type of Future, and an exception is thrown during the execution of this method, the client can retrieved the exception by calling the getException method on the returned Future instance

Your answer is incorrect.

Answer: A

Explanation:

Subsection 4.5.3 of the EJB Specification declares that client transaction context does not propagate with an asynchronous method invocation. From the bean developer's view, there is never a transaction context flowing in from the client. Meanwhile, subsection 4.5.4 mandates that caller security principal propagate with an asynchronous method invocation. Therefore, option A is correct while option B is not.

Option C is incorrect as the exception is not visible to the client only if the return type of the asynchronous method is void. Option D is incorrect since the exception must be retrieved via the getCause method of an ExecutionException thrown by the Future.get method.

The correct answer is: The transaction context of the client, if any, propagates with the method invocation

Submit your Feedback/Queries to our Experts

QUESTION 6 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a session bean:

```
@Stateless
public class Whizlabs {
   @Resource(name = "limit1", lookup = "limit2")
   private int minAmount = 1;
   public void doSomething() {
      System.out.println(minAmount);
      // do something
   }
   // other declarations
}
```

And part of the deployment descriptor that is associated with the Whizlabs bean:

```
<env-entry>
   <env-entry-name>limit1</env-entry-name>
   <env-entry-type>java.lang.Integer</env-entry-type>
   <env-entry-value>2</env-entry-value>
</env-entry>
<env-entry>
   <env-entry-name>limit2</env-entry-name>
   <env-entry-type>java.lang.Integer</env-entry-type>
   <env-entry-value>3</env-entry-value>
</env-entry>
```

What happens when the doSomething method of Whizlabs is executed?

Please select :

- A. Number 1 is printed on the console

B. Number 2 is printed on the console

C. Number 3 is printed on the console

D. An exception is thrown

Your answer is incorrect.

Answer: B

Explanation:

Subsection 16.2.3 of the EJB 3.1 Specification declares that the same environment entry may be declared using both an annotation and a deployment descriptor entry. In this case, the information in the deployment descriptor entry may be used to override some of the information provided in the annotation.

Notes that the relevant deployment descriptor entry is located based on the JNDI name used with the @Resource annotation. Applying to the given scenario, the env-entry element with the name of limit1 is associated with the minAmount field. As such, the value of this field is taken from the limit1 element, ignoring the lookup that retrieves value from the limit2 env-entry element.

The correct answer is: Number 2 is printed on the console

Submit your Feedback/Queries to our Experts

QUESTION 7 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a session bean within the default package:

```
@Stateless
@Resource(/* elements */)
public class Whizlabs {
   // bean class body
}
```

Which of the following statements is correct about the @Resource annotation shown above?

Please select :

A. The name element must be specified, while the type element is optional which defaults to Whizlabs.class

B. Both the name and type elements must be specified

C. Without specifying any element, this annotation is valid but does not have any effect

D. Without specifying any element, this annotation binds a resource of type Whizlabs to name "Whizlabs"

Your answer is incorrect.

Answer: B

Explanation:

As per the Java EE 6 API documentation, if the @Resource annotation applies to a class, its name and type elements have no default value and must be explicitly specified.

References:

http://docs.oracle.com/javaee/6/api/javax/annotation/Resource.html

The correct answer is: Both the name and type elements must be specified

Submit your Feedback/Queries to our Experts

QUESTION 8 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following types can NOT be declared on a non-transient field of a stateful session bean, provided the object referenced by the field must survive bean passivation?

Please select :

A. javax.transaction.UserTransaction

B. javax.ejb.Timer

C. java.sql.Connection

D. javax.naming.Context

E. javax.ejb.SessionContext

Your answer is incorrect.

Answer: C

Explanation:

As per the EJB 3.1 Specification (subsection 4.2.1), the bean provider is required to ensure that the PrePassivate method leaves the instance fields and the fields of its associated interceptors ready to be serialized by the container. The objects that are assigned to the instance's non-transient fields and the non-transient fields of its interceptors after the PrePassivate method completes must be one of the following:

- A serializable object
- A null
- A reference to an enterprise bean's business interface or no-interface view
- A reference to the SessionContext object
- A reference to the environment naming context
- A reference to the UserTransaction interface
- A reference to a resource manager connection factory
- A reference to a container-managed EntityManager object

- A reference to an EntityManagerFactory object obtained via injection or JNDI lookup
- A reference to a javax.ejb.Timer object
- A collection of those types mentioned above

The correct answer is: java.sql.Connection

Submit your Feedback/Queries to our Experts

QUESTION 9 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a business interface and a bean class:

```
public interface Whizlabs {
   @Asynchronous
   public Future<String> asynchronusMethod();
}
@Stateful
@Local(Whizlabs.class)
public class WhizlabsBean implements Whizlabs {
   @Resource
   private SessionContext context;
   public Future<Boolean> asyncMethod() {
      // do something
      wasCancelCalled = context.wasCancelCalled(); // Line 1
      return new AsyncResult<Boolean>(wasCancelCalled);
   }
   // other declaration
}
```

The asynchronous method shown above is invoked by a local client:

```
@Stateless
public class MyBean {
   @EJB
   private Whizlabs whizlabs;
   public void invokeAsync() {
      Future<Boolean> result = whizlabs.asyncMethod();
      // do something
   }
   // other declarations
}
```

When the invokeAsync method is executed, the body of asyncMethod runs and the value of variable wasCancelCalled after Line 1 is executed is true. Which TWO of the following statements are correct?

Please select :

- A. The statement result.cancel(true) has been called from the invokeAsync method
- B. The statement result.cancel(false) has been called from the invokeAsync method
- C. The cancelation of the asynchronous invocation is successful
- D. The cancelation of the asynchronous invocation is unsuccessful

Your answer is incorrect.

Answer: A and D

Explanation:

As per the EJB 3.1 Specification (subsection 3.4.8.1.1), if a client calls cancel on its Future object, the container will attempt to cancel the associated asynchronous invocation only if that invocation has not already been dispatched. Since the asyncMethod is executed, the invocation has been dispatched and the cancelation fails.

The flag control passed to the Future.cancel method controls whether the target enterprise bean should have visibility to the client's cancel attempt. Since the wasCancelCalled variable is true, the argument to result.cancel is true as well.

The correct answers are: The statement result.cancel(true) has been called from the invokeAsync method, The cancelation of the asynchronous invocation is unsuccessful

Submit your Feedback/Queries to our Experts

QUESTION 10 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which method can a session bean client invoke on the Future object, which is returned from an asynchronous business method, to retrieve the result of that method's computation?

Please select :

- A. retrieve
- B. get
- C. getResult
- D. take

Your answer is incorrect.

Answer: B

Explanation:

The Future interface defines the following methods to work with asynchronous method invocations:

- cancel(boolean mayInterruptIfRunning): Attempts to cancel execution of this task.

- get(): Waits if necessary for the computation to complete, and then retrieves its result.

- get(long timeout, TimeUnit unit): Waits if necessary for at most the given time for the computation to complete, and then retrieves its result, if available.

- isCancelled(): Returns true if this task was cancelled before it completed normally.

- isDone(): Returns true if this task completed.

This interface is part of the Java SE API, thus it works with all Java programs, not just enterprise session beans. Here is the detailed description of the Future.get method in the EJB 3.1 Specification (subsection 3.4.8.1.2):

The client calls one of the two Future.get methods in order to retrieve the result value or resulting exception from the associated asynchronous invocation. The specification recommends that unless the client successfully cancels the asynchronous invocation it should call get on every Future object it receives. If a call to get successfully returns a result value or throws an ExecutionException, all subsequent calls to get on the same Future object must result in that same behavior.

An EJB Container Provider is permitted to define a timeout value that governs the maximum amount of time the container maintains result values for completed asynchronous invocations. The configuration of such a timeout is beyond the scope of this specification.

References:

https://docs.oracle.com/javase/6/docs/api/java/util/concurrent/Future.html

The correct answer is: get

Submit your Feedback/Queries to our Experts

QUESTION 11 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following cannot be designated as an asynchronous business method of an enterprise session bean?

Please select :

- A. public String doSomething() { ... }
- B. public void doSomething() throws Exception{ ... }
- C. public Future<String> doSomething() throws Exception { ... }
- D. None of the above

Your answer is incorrect.

Answer: C

Explanation:

As per the EJB 3.1 Specification (subsection 4.5.2), the client return type of an asynchronous method is either void or java.util.concurrent.Future<V>, where V is the result value type. An asynchronous method with return type void must not declare any application exceptions. An asynchronous method with return type Future<V> is permitted to declare application exceptions.

The correct answer is: public Future<String> doSomething() throws Exception { ... }

Submit your Feedback/Queries to our Experts

QUESTION 12 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given two valid stateles session beans:

```
@Stateless
public class WhizlabsBean implements Whizlabs {
   public void businessMethod() { ... }
   // other declarations
}
@Stateless
public class MyBean {
   @EJB
   Whizlabs whizlabs;
   public void doSomething() {
      whizlabs.businessMethod();
      // do something
   }
   // other declarations
}
```

What can you tell about the doSomething method of MyBean?

Please select :

- A. It must be a business method
- B. It may be a timeout callback method
- C. It may be a @PostConstruct callback method
- D. It may be a @PreDestroy callback method

Your answer is incorrect.

Answer: B

Explanation:

As per Table 2 in the @EJB 3.1 Specification (subsection 4.7.2), lifecycle callback methods do not have access to enterprise beans, while business methods and timeout callback method do.

The correct answer is: It may be a timeout callback method

Submit your Feedback/Queries to our Experts

QUESTION 13 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a JMS Connection instance referenced by variable connection and a Queue object referenced by variable queue. These variables are used in the following code fragment:
Session session = connection.createSession(false, AUTO_ACKNOWLEDGE);
MessageConsumer consumer = session.createConsumer(queue);
connection.start();
Message message = consumer.receive();
What needs to be done to achieve once-and-only-once message delivery on the consumer, given the message is persistent?

Please select :

- A. Nothing, since a message delivery mode of PERSISTENT is enough to guarantee such a level of reliability
- B. The first argument to Connection.createSession must be changed to true, and session.commit must be called right after the message is received
- C. The first argument to Connection.createSession must be changed to true, and session.commit must be called after the message is processed
- D. It is impossible to achieve once-and-only-once message delivery from a queue

Your answer is incorrect.

Answer: C

Explanation:

A message delivery mode of PERSISTENT is not enough to guarantee once-and-only-once delivery. With automatic acknowledgement, the JMS message is deleted from the queue as soon as it is received. If the program crashes when the message is being processed, it (the message) will be lost forever.

The same happens if you implement the changes mentioned in option B. Once again, the message will be lost if the program crashes while the client is processing the message.

The solution in option C is the correct way to meet the requirement. A message is deleted from the queue only if it has been successfully processed, ensuring that no message is lost or resent.

The correct answer is: The first argument to Connection.createSession must be changed to true, and session.commit must be called after the message is processed

Submit your Feedback/Queries to our Experts

QUESTION 14 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following does not represent a JMS message type?

Please select :

- A. BytesMessage
- B. ListMessage
- C. MapMessage
- D. ObjectMessage
- E. StreamMessage

Your answer is incorrect.

Answer: B

Explanation:

The JMS API defines five message body formats, also called message types, which allow you to send and receive data in many different forms and which provide compatibility with existing messaging formats:

- TextMessage: A java.lang.String object (for example, the contents of an XML file).

- MapMessage: A set of name-value pairs, with names as String objects and values as primitive types in the Java programming language. The entries can be accessed sequentially by enumerator or randomly by name. The order of the entries is undefined.

- BytesMessage: A stream of uninterpreted bytes. This message type is for literally encoding a body to match an existing message format.

- StreamMessage: A stream of primitive values in the Java programming language, filled and read sequentially.

- ObjectMessage: A Serializable object in the Java programming language.

References:

http://docs.oracle.com/javaee/6/tutorial/doc/bnceh.html#bncew

The correct answer is: ListMessage

Submit your Feedback/Queries to our Experts

QUESTION 15 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

What happens if the receive method is invoked on a JMS MessageConsumer instance when its associated destination does not have any message?

Please select :

- A. A JMSException is thrown immediately
- B. If a message is not available within 1000 ms, a JMSException is thrown
- C. The invocation blocks indefinitely until a message is produced; if the consumer is closed while waiting, a JMSException is thrown
- D. The invocation blocks indefinitely until a message is produced or until this message consumer is closed

Your answer is incorrect.

Answer: D

Explanation:

As per the Java EE 6 API documentation, the MessageConsumer.receive method receives the next message produced for this message consumer. This call blocks indefinitely until a message is produced or until this message consumer is closed.

References:

http://docs.oracle.com/javaee/6/api/javax/jms/MessageConsumer.html#receive()

The correct answer is: The invocation blocks indefinitely until a message is produced or until this message consumer is closed

Submit your Feedback/Queries to our Experts

QUESTION 16 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which TWO of the following statements are NOT correct about message-driven beans?

Please select :

- A. A message-driven bean cannot be associated with more than one destination
- B. A queue should not be associated with more than one message-driven bean
- C. A message-driven bean associated with a topic using a non-durable subscription may lose messages
- D. A message-driven bean instance is never executed by more than one thread at the same time
- E. Messages are delivered to message-driven beans in the same order they arrive at the destination

Your answer is incorrect.

Answer: D and E

Explanation:

The EJB 3.1 Specification (subsection 5.4.17) specifies that a message-driven bean is associated with a destination or endpoint when the bean is deployed in the container. It is the responsibility of the deployer to associate the message-driven bean with a destination or endpoint. Therefore, option A is incorrect.

This Specification (subsection 5.4.17.1) also declares that the deployer should avoid associating more than one message-driven bean with the same JMS Queue. If there are multiple JMS consumers for a queue, JMS does not define how messages are distributed between the queue receivers. Thus, option B is incorrect.

The same subsection of the Specification also says that if a non-durable topic subscription is used, it is the container's responsibility to make sure that the message driven bean subscription is active in order to ensure that messages are not missed as long as the EJB server is running. Messages may be missed, however, when a bean is not available to service them. This will occur, for example, if the EJB server goes down for any period of time. So, option C is incorrect.

Option D is correct since subsection 5.7.4 of the Specification requires the container to ensure that only one thread can be executing an instance at any time.

Option E is also correct, due to subsection 5.4.11 of the Specification: No guarantees are made as to the exact order in which messages are delivered to the instances of the message-driven bean class, although the container should attempt to deliver messages in order when it does not impair the concurrency of message processing. Message-driven beans should therefore be prepared to handle messages that are out of sequence.

The correct answers are: A message-driven bean instance is never executed by more than one thread at the same time, Messages are delivered to message-driven beans in the same order they arrive at the destination

Submit your Feedback/Queries to our Experts

QUESTION 17 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following is NOT a standard activation configuration property for a JMS message-driven bean?

Please select :

- A. acknowledgeMode
- B. messageSelector
- C. destinationType
- D. topicDurability

Your answer is incorrect.

Answer: D

Explanation:

As per the Java EE 6 API documentation, the following standard properties are recognized for JMS message driven beans:

- acknowledgeMode: This property is used to specify the JMS acknowledgement mode for the message delivery when bean-managed transaction demarcation is used. Its values are Auto_acknowledge or Dups_ok_acknowledge. If this property is not specified, JMS auto acknowledge semantics are assumed.

- messageSelector: This property is used to specify the JMS message selector to be used in determining which messages a JMS message driven bean is to receive.

- destinationType: This property is used to specify whether the message driven bean is intended to be used with a queue or a topic. The value must be either javax.jms.Queue or javax.jms.Topic.

- subscriptionDurability: If the message driven bean is intended to be used with a topic, this property may be used to indicate whether a durable or non-durable subscription should be used. The value of this property must be either Durable or NonDurable

References:

http://docs.oracle.com/javaee/6/api/javax/ejb/ActivationConfigProperty.html

The correct answer is: topicDurability

Submit your Feedback/Queries to our Experts

QUESTION 18 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which interface must a JMS message-driven class implement?

Please select :

- A. MessageDriven
- B. MessageDrivenBean
- C. MessageContext
- D. MessageListener
- E. None of the above

Your answer is incorrect.

Answer: D

Explanation:

The EJB 3.1 Specification (subsection 5.6.2) declares that a message-driven class must implement, directly or indirectly, the message listener interface required by the messaging type that it supports or the methods of the message listener interface. In the case of JMS, this is the MessageListener interface.

MessageDriven is an annotation, while the MessageContext interface has nothing to do with JMS or message-driven beans. MessageDrivenBean is a valid interface, but implementing it is optional, not required.

The correct answer is: MessageListener

Submit your Feedback/Queries to our Experts

QUESTION 19 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a message-driven bean that implements two message listener interfaces:

```
// some annotations
public class Whizlabs implements MyListener1, MyListener2 {
   // class body
}
```

Which of the following statements is correct about the above bean class, given there are no deployment descriptor elements associated with this bean?

Please select :

- A. Either MyListener1 or MyListener2 must be specified in the messageListenerInterface element of the @MessageDriven annotation to make the bean valid
- B. Either MyListener1 or MyListener2 must be specified in the @ListenerInterface annotation declared on the given bean to make the bean valid
- C. The given bean is invalid since it does not implement the MessageListener interface
- D. When the given bean is triggered, message listener methods of both implemented interfaces are executed, with the one defined by MyListener1 running first

Your answer is incorrect.

Answer: A

Explanation:

As per the EJB 3.1 Specification (subsection 5.4.2), if the message-driven bean class implements more than one interface other than java.io.Serializable, java.io.Externalizable, or any of the interfaces defined by the javax.ejb package, the message listener interface must be specified by the messageListenerInterface element of the MessageDriven annotation or the messaging-type element of the message-driven deployment descriptor element.

Note that the @ListenerInterface annotation does not exist, while the @MessageListener interface just needs to be implemented if the given bean is a JMS message-driven bean.

The correct answer is: Either MyListener1 or MyListener2 must be specified in the messageListenerInterface element of the @MessageDriven annotation to make the bean valid

Submit your Feedback/Queries to our Experts

QUESTION 20 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given an enterprise bean:

```
@Stateless
@Interceptors(InterceptorA.class)
public class Whizlabs implements A {
   @Interceptors(InterceptorB.class)
   @ExcludeClassInterceptors
   @ExcludeDefaultInterceptors
   public void doSomething() { ... }
   // other declarations
}
```

Here is part of the deployment descriptor:

```
<assembly-descriptor>
   <interceptor-binding>
      <target-name>Whizlabs</target-name>
      <interceptor-class>InterceptorC</interceptor-class>
   </interceptor-binding>
</assembly-descriptor>
```

Assume InterceptorA, InterceptorB and InterceptorC are declared with appropriate annotations. Which interceptor methods are invoked when the Whizlabs.doSomething method is called (the classes shown in options below are the containing classes of respective interceptor methods)?

Please select :

- A. InterceptorA, InterceptorB and InterceptorC

- B. InterceptorB and InterceptorC
- C. InterceptorB only
- D. No interceptor methods will be invoked
- E. The declaration of the Whizlabs.doSomething method is invalid

Your answer is incorrect.

Answer: B

Explanation:

There is nothing wrong with method Whizlabs.doSomething. Its metadata annotations exclude class level and default interceptors. InterceptorC is not a default interceptor as it is bound to bean Whizlabs. As a result, interceptor methods of InterceptorB and InterceptorC are invoked.

The correct answer is: InterceptorB and InterceptorC

Submit your Feedback/Queries to our Experts

QUESTION 21 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given an interceptor class:

```
@Interceptor
public class MyInterceptor{
  @AroundInvoke
  Object intercept(InvocationContext context) throws Exception {
    Object timer = context.getTimer();
    // do something
  }
}
```

What happens when the expression context.getTimer() is executed?

Please select :

- A. It returns the most recently created Timer instance associated with the interceptor's target instance
- B. It returns a collection of all Timer instances associated with the interceptor's target instance
- C. It returns a null value
- D. It throws an exception

Your answer is incorrect.

Answer: C

Explanation:

As per the Java EE 6 API documentation, the InvocationContext.getTimer method returns the timer object associated with a timeout method invocation on the target class, or a null value for method and lifecycle callback interceptor methods.

References:

http://docs.oracle.com/javaee/6/api/javax/interceptor/InvocationContext.html#getTimer()

The correct answer is: It returns a null value

Submit your Feedback/Queries to our Experts

QUESTION 22 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given an interceptor class:

```
@Interceptor
public class MyInterceptor {
  @AroundInvoke
  Object intercept(InvocationContext context) throws Exception {
    Object object = context.proceed();
    System.out.print("Intercepting");
    return object;
  }
}
```

The above interceptor class is specified within a bean:

```
@Stateless
public class Whizlabs {
  @Interceptors(MyInterceptor.class)
  public String printText(String text) {
    System.out.print(text);
    return text;
  }
  // other declarations
}
```

What is printed when a local client invokes the Whizlabs.printText method with argument "Whizlabs"?

Please select :

- A. "Whizlabs"
- B. "WhizlabsIntercepting"

C. "InterceptingWhizlabs"

D. The given interceptor method is invalid

Your answer is incorrect.

Answer: B

Explanation:

When the printText method is invoked, the interceptor method interpose on the invocation. This interceptor method instructs the container to invoke the target method first through the context.proceed method. After that, the string "Intercepting" is printed out from the interceptor method.

The correct answer is: "WhizlabsIntercepting"

Submit your Feedback/Queries to our Experts

QUESTION 23 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which TWO of the following statements are NOT correct about lifecycle callback interceptors?

Please select :

A. Lifecycle callback interceptor methods may be defined for session beans and message driven beans

B. Regarding metadata annotations, only the @PostConstruct and @PreDestroy annotations designate lifecycle callback interceptor methods

C. All lifecycle callback interceptor methods are invoked in an unspecified transaction context

D. All lifecycle callback interceptor methods are invoked in an unspecified transaction context

E. A given class may not have more than one lifecycle callback interceptor method for the same lifecycle event

F. A single lifecycle callback interceptor method may be used to interpose on multiple callback events

Your answer is incorrect.

Answer: B and D

Explanation:

The following is an excerpt from the EJB 3.1 Specification (subsection 12.5):

Lifecycle callback interceptor methods may be defined for session beans and message driven beans.

Interceptor methods for lifecycle event callbacks can be defined on an interceptor class and/or directly on the bean class. The @PostConstruct, @PreDestroy, @PostActivate, and @PrePassivate annotations are used to define an interceptor method for a lifecycle callback event.

Lifecycle callback interceptor methods are invoked in an unspecified security context. Lifecycle callback

interceptor methods are invoked in an unspecified transaction context, except for Singleton @PostConstruct / @PreDestroy methods, whose transaction context is based on their associated transaction attribute.

In addition, the Interceptors 1.1 Specification informs that lifecycle callback interceptor methods may be defined on superclasses of the target class or interceptor classes. However, a given class may not have more than one lifecycle callback interceptor method for the same lifecycle event. Any subset or combination of lifecycle callback annotations may be specified on a given class.

References:

http://docs.oracle.com/javaee/6/tutorial/doc/gkedm.html#gkecr

The correct answers are: Regarding metadata annotations, only the @PostConstruct and @PreDestroy annotations designate lifecycle callback interceptor methods, All lifecycle callback interceptor methods are invoked in an unspecified transaction context

Submit your Feedback/Queries to our Experts

QUESTION 24 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given three interceptor classes:

```
@Interceptor
public class InterceptorA {
   @AroundInvoke
   Object intercept(InvocationContext context) throws Exception {
      context.getContextData().put("a", 1);
      return context.proceed();
   }
}
@Interceptor
public class InterceptorB {
   @AroundInvoke
   Object intercept(InvocationContext context) throws Exception {
      System.out.println(context.getContextData().size());
      return context.proceed();
   }
}
@Interceptor
public class InterceptorC {
   @AroundInvoke
   Object intercept(InvocationContext context) throws Exception {
      context.getContextData().put("c", 3);
      return context.proceed();
   }
}
```

The above interceptor classes are specified within a bean:

```
@Stateless
public class Whizlabs implements A {
   @Interceptors({InterceptorA.class, InterceptorB.class, InterceptorC.class})
   public void doSomething() { ... }
   // other declarations
}
```

What is printed when the Whizlabs.doSomething method is invoked by a local client, provided there is no associated deployment descriptor?

Please select :

- A. 0
- B. 1
- C. 2
- D. Nothing, as a NullPointerException is thrown

Your answer is incorrect.

Answer: B

Explanation:

When the around-invoke method in InterceptorB is invoked, the context data of the invocation contains only one entry, added by InterceptorA. InterceptorC also adds another entry, but this is done after the print statement is executed. As a result, the invocation context.getContextData().size() returns 1.

The correct answer is: 1

Submit your Feedback/Queries to our Experts

QUESTION 25 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following is NOT a responsibility of the deployer with respect to security?

Please select :

- A. Assign principals and/or groups of principals used for managing security in the operational environment to the security roles defined in the application
- B. Define the mapping from the security domain and principal realm used at the application level to the security domain and principal realm of the resource manager
- C. Link security role references to security roles
- D. None of the above

Your answer is incorrect.

Answer: C

Explanation:

Except for linking security role references to security roles, which is the job of the application assembler, the two others described in options A and B are responsibilities of the deployer.

The correct answer is: Link security role references to security roles

Submit your Feedback/Queries to our Experts

QUESTION 26 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given two class declarations:

```
@RolesAllowed("alpha")
public class SuperClass {
   public void methodA() { ... }
   @RolesAllowed("beta")
   public void methodB() { ... }
   // other declarations
}
@Stateless
@RolesAllowed("gamma")
public class SubClass extends SuperClass implements MyInterface {
   public void methodA() { ... }
   public void methodC() { ... }
   // other declarations
}
```

Assume the MyInterface interface defines methodA, methodB and methodC, and there are no security-related elements in the deployment descriptor. Identify the correct statement among the following when the business interface MyInterface is accessed:

Please select :

- A. methodA and methodC allow access by role gamma only, methodB allows access from all clients
- B. methodA allows access by role alpha only, methodB allows access by role beta only, and methodC allows access by role gamma only
- C. methodA and methodC allow access by role gamma only, methodB allows access by role beta only
- D. methodA, methodB and methodC all allow access by role gamma only

Your answer is incorrect.

Answer: C

Explanation:

methodA and methodC allows access from clients in role gamma as specified on the SubClass declaration. methodB allows access from clients in role beta only as annotated within SuperClass. Note that even though methodB is inherited in SubClass, its permission value is not overridden since there is no overriding in the subclass.

The correct answer is: methodA and methodC allow access by role gamma only, methodB allows access by role beta only

Submit your Feedback/Queries to our Experts

QUESTION 27 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a fragment of the deployment descriptor:

```
<method-permission>
  <role-name>role1</role-name>
  <role-name>role2</role-name>
  <method>
    <ejb-name>JavaSE</ejb-name>
    <method-name>prepareForExam</method-name>
  </method>
  <method>
    <ejb-name>JavaEE</ejb-name>
    <method-name>takeAnExam</method-name>
  </method>
</method-permission>
```

Which of the following statements is correct about the above fragment?

Please select :

- A. It is invalid as having more than one role-name element
- B. It is invalid as having more than one method element
- C. It is invalid as having more than one role-name element and more than one method element at the same time
- D. It is invalid as role-name elements must have been wrapped within a role-names element, and method elements wrapped within a methods element
- E. It is a valid method-permission element

Your answer is incorrect.

Answer: E

Explanation:

As per the EJB 3.1 Specification (subsection 17.3.2.2), each method-permission element includes a list of one or more security roles and a list of one or more methods. All the listed security roles are allowed to invoke all the listed methods. Each security role in the list is identified by the role-name element, and each method (or a set of methods) is identified by the method element. An optional description can be associated with a method-permission element using the description element.

The correct answer is: It is a valid method-permission element

Submit your Feedback/Queries to our Experts

QUESTION 28 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following operators/methods can be used by a client to test two EJB 3.x remote/local view references for equality?

Please select :

- A. The == operator
- B. The Object.equals method
- C. The EJBObject.isIdentical method
- D. The SessionContext.isEqual method
- E. None of the above

Your answer is incorrect.

Answer: B

Explanation:

The == operator can not be used to test for equality of Java objects of any type, while the SessionContext interface does not define a method named isEqual. Thus, options A and D are incorrect.

The EJBObject.isIdentical method is used for EJB 2.x beans only. Option C is incorrect as well.

The EJB 3.1 Specification (subsection 3.4.7) declares that a client can test two EJB 3.x Remote/Local view references for identity by means of the Object.equals and Object.hashCode methods. As such, option B is the correct answer.

The correct answer is: The Object.equals method

Submit your Feedback/Queries to our Experts

QUESTION 29 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a valid stateles session bean:

```
@Stateless
public class Whizlabs {
  @PersistenceContext
  private EntityManager entityManager;
```

```
  public void manageEntities() {
    // do something with entityManager
  }
  // other declarations
}
```

What can you tell about the manageEntities method, provided there is no associated deployment descriptor?

Please select :

- A. It must be a @PostConstruct method
- B. It may be either a @PostConstruct or @PreDestroy method
- C. It may be a timeout callback method
- D. It must be a business method

Your answer is incorrect.

Answer: C

Explanation:

As per Table 2 in the EJB 3.1 Specification (subsection 4.7.2), lifecycle callback methods do not have access to an EntityManager, while business methods and timeout callback method do.

The correct answer is: It may be a timeout callback method

Submit your Feedback/Queries to our Experts

QUESTION 30 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which types of session beans support loopback calls (a situation where an outbound call from a bean results in an inbound call on the same instance)?

Please select :

- A. Stateful and Stateless
- B. Singleton only
- C. Stateless and singleton
- D. Stateful, stateless and singleton

Your answer is incorrect.

Answer: B

Explanation:

The following is the content of subsection 4.10.13 of the EJB 3.1 Specification:

The container must ensure that only one thread can be executing a stateless or stateful session bean instance at any time. Therefore, stateful and stateless session beans do not have to be coded as reentrant. One implication of this rule is that an application cannot make loopback calls to a stateless or stateful session bean instance.

The correct answer is: Singleton only

Submit your Feedback/Queries to our Experts

QUESTION 31 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which type of session beans supports concurrent access to a bean instance?

Please select :

- A. Singleton
- B. Stateless
- C. Stateful
- D. None of the above

Your answer is incorrect.

Answer: A

Explanation:

A singleton session bean has only one instance, which is shared between all clients and supports concurrent access.

The correct answer is: Singleton

Submit your Feedback/Queries to our Experts

QUESTION 32 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following statements is NOT correct about the message listener method of a message-driven bean with container-managed transaction demarcation?

Please select :

- A. The container throws an IllegalStateException when the method calls EJBContext.getUserTransaction
- B. The container throws an IllegalStateException if the method calls EJBContext.getRollbackOnly when it is declared with transaction attribute NotSupported
- C. The container ensures that the transaction will never commit if the method calls EJBContext.setRollbackOnly when it is declared with transaction attribute Required

D. None of the above

Your answer is incorrect.

Answer: D

Explanation:

As per the EJB 3.1 Specification (subsection 13.6.3.5), if an instance of a message-driven bean with container-managed transaction demarcation attempts to invoke the getUserTransaction method of the EJBContext interface, the container must throw and log the java.lang.IllegalStateException. Note this is true for session beans as well.

The descriptions in options B and C are operations the container must implement in respective situations. These are declared in subsections 13.6.3.3 and 13.6.3.4 of the EJB 3.1 Specification.

The correct answer is: None of the above

Submit your Feedback/Queries to our Experts

QUESTION 33 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which THREE of the following transaction attributes can be used on methods of an enterprise bean with container-managed transaction demarcation that implements the SessionSynchronization interface?

Please select :

- A. Required
- B. RequiresNew
- C. Mandatory
- D. NotSupported
- E. Supports
- F. Never

Your answer is incorrect.

Answer: A, B and C

Explanation:

As per the EJB 3.1 Specification (subsection 13.3.7), if an enterprise bean implements the javax.ejb.SessionSynchronization interface or uses at least one of the session synchronization annotations, only the following values may be used for the transaction attributes of the bean's methods: REQUIRED, REQUIRES_NEW, MANDATORY.

The above restriction is necessary to ensure that the enterprise bean is invoked only in a transaction. If the bean were invoked without a transaction, the container would not be able to send the transaction synchronization calls.

The correct answers are: Required, RequiresNew, Mandatory

Submit your Feedback/Queries to our Experts

QUESTION 34 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given four methods in stateless session beans with container-managed transaction demarcation. These methods are declared with the following transaction attributes:
methodA: Required
methodB: NotSupported
methodC: Never
methodD: RequiresNew
Which of the following invocation chains is correct if the client calling methodA runs within transaction T1?
Note: The arrow (-->) represents a method invocation; while T1, T2 denotes transaction 1, transaction 2 and No means the associated method runs outside a transaction.

Please select :

- A. methodA (T1) --> methodB (No) --> methodC (No) --> methodD (T2)
- B. methodA (T1) --> methodB (T1) --> methodC (No) --> methodD (T2)
- C. methodA (T2) --> methodB (No) --> methodC (No) --> methodD (T2)
- D. methodA (No) --> methodB (No) --> methodC (No) --> methodD (T1)

Your answer is incorrect.

Answer: A

Explanation:

The transaction attribute of methodB is NotSupported, thus methodB must not run within a transaction. Option B is incorrect, therefore.

The transaction attribute of methodA is Required, so methodA must run within the same transaction as its client. Options C and D are both incorrect, then.

References:

http://docs.oracle.com/javaee/6/tutorial/doc/bncij.html#bncik

The correct answer is: methodA (T1) --> methodB (No) --> methodC (No) --> methodD (T2)

Submit your Feedback/Queries to our Experts

QUESTION 35 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following statements is correct about transaction demarcation?

Please select :

- A. Message-driven beans can work with bean-managed, not container-managed, transaction demarcation
- B. Session beans can be designed with container-managed and bean-managed transaction demarcations at the same time
- C. Entity beans must use container-managed transaction demarcation
- D. Enterprise beans cannot access resource managers that do not support an external transaction coordinator

Your answer is incorrect.

Answer: C

Explanation:

As per the EJB 3.1 Specification (subsection 13.3.1), a session bean or a message-driven bean can be designed with bean-managed transaction demarcation or with container-managed transaction demarcation. (But it cannot be both at the same time).

An EJB 2.1 or EJB 1.1 entity bean must always use container-managed transaction demarcation. An EJB 2.1 or EJB 1.1 entity bean must not be designated with bean-managed transaction demarcation.

If an enterprise bean needs to access a resource manager that does not support an external transaction coordinator, the bean provider should design the enterprise bean with container-managed transaction demarcation and assign the NOT_SUPPORTED transaction attribute to the bean class or to all the bean's methods.

The correct answer is: Entity beans must use container-managed transaction demarcation

Submit your Feedback/Queries to our Experts

QUESTION 36 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a session bean declaration:

```
@Stateless
@TransactionManagement(TransactionManagementType.BEAN)
public class WhizlabsBean implements MyInterface {
  @Resource
  private EJBContext context;
  public void doSomething() throws Exception {
    UserTransaction userTransaction = context.getUserTransaction();
    userTransaction.begin();
    userTransaction.setRollbackOnly();
    // do something
    userTransaction.commit();
  }
  // other declarations
}
```

What happens when the userTransaction.commit method is invoked?

Please select :

- A. The current transaction is committed
- B. The current transaction is rolled back
- C. The container throws an IllegalStateException
- D. The container throws a RollbackException
- E. The container throws a SystemException

Your answer is incorrect.

Answer: B

Explanation:

When the userTransaction.setRollbackOnly method is invoked, the only possible outcome of the current transaction is rollback. Thus, the transaction is rolled back when method commit is invoked.

Option C is incorrect since an IllegalStateException is thrown only if the current thread is not associated with a transaction. Option D is incorrect as a RollbackException is thrown only if the transaction has been rolled back, not marked for rollback. Option E is incorrect because a SystemException is just thrown if the transaction manager encounters an unexpected error condition.

References:

http://docs.oracle.com/javaee/6/api/javax/transaction/UserTransaction.html#commit()

The correct answer is: The current transaction is rolled back

Submit your Feedback/Queries to our Experts

QUESTION 37 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given two session beans packaged in the same module with container-managed transaction demarcation: BeanA and BeanB. BeanA is a stateless bean, containing a method named methodA with transaction attribute Require; BeanB is a singleton bean, defining a method named methodB with transaction attribute RequiresNew. A local client running within a transaction calls methodA, which in turn calls methodB. When executing, methodB throws a system exception. Which of the following statements is correct, given no exceptions are caught in either methodB or methodA?

Please select :

- A. Both bean instances are discarded
- B. Neither of the bean instances is discarded
- C. The client transaction is marked for rollback
- D. The client gets an EJBException but it may continue within the same transaction

Your answer is incorrect.

Answer: C

Explanation:

When a system exception is thrown from methodB, the transaction that the container started before the execution of methodB is rolled back, but the client transaction is unaffected. BeanB is a singleton session bean, so it will not be discarded (it would be if were a stateless or stateful session bean). methodA will receives an EJBException as a result of the system exception being thrown from methodB.

Since EJBException is also a system exception and methodA does not handle it, the container will throw an EJBTransactionRolledbackException and discard the BeanA instance. The transaction attribute of methodA is Required, implying that methodA and the client run in the same transaction. The exception thrown from methodA will mark this transaction for rollback.

The correct answer is: The client transaction is marked for rollback

Submit your Feedback/Queries to our Experts

QUESTION 38 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a situation where a method of a stateless session bean with container-managed transaction demarcation handles data submitted from clients. The transaction attribute of this method is Mandatory. If the received data is invalid, the method will throw a custom application exception of type InvalidDataException. You are required to write the code that allows clients to fix the problem and retry the submission. Which of the following is a correct solution to achieve that?

Please select :

- A. Calling the UserTransaction.setRollbackOnly method before throwing the exception
- B. Calling the EJBContext.setRollbackOnly method before throwing the exception
- C. Calling the EJBContext.setRollbackOnly(false) method before throwing the exception
- D. Annotating the application exception with @ApplicationException(rollback = true)
- E. Annotating the application exception with @ApplicationException(rollback = false)

Your answer is incorrect.

Answer: E

The UserTransaction.setRollbackOnly can only be used in beans with bean-managed transaction demarcation. Thus, option A is incorrect.

Invoking EJBContext.setRollbackOnly or annotating the exception with @ApplicationException(rollback = true) causes the transaction to be marked for rollback. This will prevent clients from retrying data submission. As a result, options B and D are both incorrect.

The EJBContext interface does not define a method with the signature setRollbackOnly(boolean). Hence, option C is incorrect.

Annotating the application exception with @ApplicationException(rollback = false) gives clients a chance to continue its job in the existing transaction. You may also achieve this by not annotating the exception with this annotation, or not assigning a value to the rollback element of this annotation, since the default value of the rollback element is false.

The correct answer is: Annotating the application exception with @ApplicationException(rollback = false)

Submit your Feedback/Queries to our Experts

QUESTION 39 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

The Java EE platform has a lightweight profile. What is it called and what is the type of application it targets at?

Please select :

- A. It is called Lite Profile, targeting at desktop applications
- B. It is called Web Profile, targeting at next-generation web applications
- C. It is called Reduced Profile, targeting at trivial enterprise applications
- D. None of the above

Your answer is incorrect.

Answer: B

Explanation:

As per the Oracle's Java EE 6 Tutorial, the Java EE 6 platform introduces a lightweight Web Profile targeting at next-generation web applications, as well as a Full Profile that contains all Java EE technologies and provides the full power of the Java EE 6 platform for enterprise applications.

References:

http://docs.oracle.com/javaee/6/tutorial/doc/giqvh.html

The correct answer is: It is called Web Profile, targeting at next-generation web applications

Submit your Feedback/Queries to our Experts

QUESTION 40 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which THREE of the following APIs required by the Java EE 6 platform are included in the Java SE platform?

Please select :

- A. JACC
- B. JAXB
- C. JAX-WS
- D. JMS
- E. JSP
- F. SAAJ

Your answer is incorrect.

Answer: B, C and F

Explanation:

As per the Oracle's Java EE 6 Tutorial, Java Architecture for XML Binding (JAXB), SOAP with Attachments API for Java (SAAJ), and Java API for XML Web Services (JAX-WS) specifications are included in the Java SE platform; while Java Message Service API (JMS), Java Authorization Contract for Containers (JACC) and JavaServer Pages (JSP) are implemented by Java EE containers.

References:

http://docs.oracle.com/javaee/6/tutorial/doc/girdr.html

The correct answers are: JAXB, JAX-WS, SAAJ

Submit your Feedback/Queries to our Experts

QUESTION 41 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

In which of the following cases all enterprise beans must be presented with the same JNDI API namespace?

Please select :

- A. Beans deployed in the same container
- B. Beans of the same application
- C. Beans deployed within the same WAR file
- D. None of the above

Your answer is incorrect.

Answer: C

Explanation:

As per the EJB 3.1 Specification (subsection 21.3.2), the EJB specification does not require that all the enterprise beans deployed in a container be presented with the same JNDI API namespace. However, all enterprise beans deployed within the same WAR are presented with the same JNDI namespace. In addition, all the instances of the same enterprise bean that is deployed within an EJB-JAR must be presented with the same JNDI API namespace.

The correct answer is: Beans deployed within the same WAR file

Submit your Feedback/Queries to our Experts

QUESTION 42 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following statements is NOT correct about JNDI lookup in an enterprise bean?

Please select :

- A. A field or method may have any access qualifier but must not be static
- B. A field that is an injection target must not be final
- C. When applying to JavaBeans properties, the annotation is declared on the getter
- D. The JNDI name is always relative to the java:comp/env naming context
- E. None of the above

Your answer is incorrect.

Answer: C

As per the EJB 3.1 Specification (subsection 16.2.2), a field or method, which is an injection target, may have any access qualifier (public, private, etc.) but must not be static.

If a field is injected, it must not be final. By default, the name of the field is combined with the name of the class in which the annotation is used and is used directly as the name in the bean's naming context.

Environment entries may also be injected into the bean through bean methods that follow the naming conventions for JavaBeans properties. The annotation is applied to the set method for the property, which is the method that is called to inject the environment entry. The JavaBeans property name (not the method name) is used as the default JNDI name

When a deployment descriptor entry is used to specify injection, the JNDI name and the instance variable name or property name are both specified explicitly. Note that the JNDI name is always relative to the java:comp/env naming context.

The correct answer is: When applying to JavaBeans properties, the annotation is declared on the getter

Submit your Feedback/Queries to our Experts

QUESTION 43 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which two of the following client views must be supported by an embeddable container?

Please select :

- A. No-interface views
- B. Local business interfaces
- C. Remote business interfaces
- D. JAX-WS web service endpoint
- E. JAX-RS web service endpoint

Your answer is incorrect.

Answer: A and B

Explanation:

An embeddable container is required to support the EJB Lite API, thus it only needs to provide clients with local and no-interface views.

The correct answers are: No-interface views, Local business interfaces

Submit your Feedback/Queries to our Experts

QUESTION 44 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following information should not be overridden in the EJB deployment descriptor once declared using metadata annotation?

Please select :

- A. Bean's transaction demarcation type
- B. Security identity
- C. Interceptors
- D. Environment entries

Your answer is incorrect.

Answer: A

Explanation:

There are two basic kinds of metadata information: enterprise beans' structural information and application assembly information. The structural information cannot, in general, be changed because doing so could break the enterprise bean's function. On the other hand, assembly level information can be changed without breaking the enterprise bean's function.

Among the given options, only bean's transaction demarcation type is structural information.

The correct answer is: Bean's transaction demarcation type

Submit your Feedback/Queries to our Experts

QUESTION 45 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following statements is correct about an embeddable container?

Please select :

- A. Client code may instantiate an EJB container that runs within its own JVM and classloader
- B. An embeddable EJB container must support the Full EJB API
- C. Enterprise beans must be customized to be able to operate in an embeddable container
- D. Modules must be explicitly specified to be recognized by an embeddable container

Your answer is incorrect.

Answer: A

Explanation:

The following extracts are taken from the EJB 3.1 Specification (subsections 22.1 and 22.2):

Embeddable usage requirements allow client code to instantiate an EJB container that runs within its own JVM and classloader. The client uses a spec-defined bootstrapping API to start the container and identify the set of enterprise bean components for execution.

The embeddable EJB container provides a managed environment with support for the same basic services that exist within a Java EE runtime: injection, access to a component environment, container-managed transactions, etc. In general, enterprise bean components are unaware of the kind of managed environment in which they are running. This allows maximum reusability of enterprise components across a wide range of testing and deployment scenarios without significant rework.

The embeddable container is instantiated using a bootstrapping API defined within the javax.ejb package. By default, the embeddable container uses the JVM class path to scan for the enterprise bean modules to be initialized. The client can override this behavior during setup by specifying an alternative set of target modules.

The correct answer is: Client code may instantiate an EJB container that runs within its own JVM and classloader

Submit your Feedback/Queries to our Experts

QUESTION 46 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given the MyBean session bean with a no-interface view, packaged within a stand-alone EJB-JAR file named myBean.jar. Also given a javax.naming.Context instance created by the following code fragment:
EJBContainer container = EJBContainer.createEJBContainer();
Context context = container.getContext();
Which of the following statements can be used to obtain a reference to MyBean in a local client?

Please select :

- A. MyBean myBean = (MyBean) context.lookup("java:app/MyBean");
- B. MyBean myBean = (MyBean) context.lookup("java:app/myBean/MyBean");
- C. MyBean myBean = (MyBean) context.lookup("java:module/myBean/MyBean");
- D. MyBean myBean = (MyBean) context.lookup("java:global/myBean/MyBean");

Your answer is incorrect.

Answer: D

Explanation:

Session bean references are identified using the portable global JNDI name syntax, which is:

java:global[/application name]/module name/enterprise bean name[/interface name]

References:

http://docs.oracle.com/javaee/6/tutorial/doc/gkcrr.html#glhur

The correct answer is: MyBean myBean = (MyBean) context.lookup("java:global/myBean/MyBean");

Submit your Feedback/Queries to our Experts

QUESTION 47 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a client invokes an enterprise bean method and receiving an EJBException. Which of the following statements is correct about this situation?

Please select :

- A. The bean's method has been completed
- B. The bean's method has not been completed
- C. The client's transaction has been marked for rollback
- D. The client's transaction may or may not have been marked for rollback

Your answer is incorrect.

Answer: D

Explanation:

The client cannot know whether the bean method has been completed as the exception could be thrown from

within that method, or the container threw the exception after the method returned.

The transaction may not necessarily be marked for rollback. This might occur, for example, when the communication subsystem on the client-side has not been able to send the request to the server.

The correct answer is: The client's transaction may or may not have been marked for rollback

Submit your Feedback/Queries to our Experts

QUESTION 48 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given an enterprise session bean with container-managed transaction management. Which TWO of the following statements are correct if a bean's business method throws an application exception while running outside a transaction?

Please select :

- A. The caller receives the original exception
- B. The caller receives an EJBException
- C. If the client executes in a transaction, the transaction is not marked for rollback
- D. If the client executes in a transaction, the transaction may or may not be marked for rollback

Your answer is incorrect.

Answer: A and C

Explanation:

When an application exception is thrown in the given situation, the client receives the original exception. If the client executes in a transaction, the client's transaction is not marked for rollback, and client can continue its work.

The correct answers are: The caller receives the original exception, If the client executes in a transaction, the transaction is not marked for rollback

Submit your Feedback/Queries to our Experts

QUESTION 49 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given the declaration of a method:

```
public void invokeRemote() {
   try {
      // invoke a remote business method of a stateful session bean
   } catch (IOException e) {
      e.printStackTrace();
   }
}
```

The above method calls a remote business interface business method of a stateful session bean with container-managed transaction demarcation from within the try block. This invocation is made in the context of a transaction, and the invoked bean method is declared with transaction attribute Required. After the calling, the client prints an exception and its backtrace to the error stream. Which of the following statements is correct about the described situation?

Please select :

- A. The transaction has been marked for rollback
- B. The enterprise bean has been destroyed
- C. The original exception has been logged in the container on which the target bean was running
- D. None of the above

Your answer is incorrect.

Answer: D

Explanation:

IOException is a superclass of RemoteException and can handle both system exception (if the exception is RemoteException and thereof) and application exceptions, therefore you cannot even tell whether the received exception is a system or application exception. The statements in options A, B and C are true only if the received exception is a system exception.

The correct answer is: None of the above

Submit your Feedback/Queries to our Experts

QUESTION 50 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a session bean (with container-managed transaction demarcation) business method running in the context of a transaction that the container started immediately before dispatching the business method. At some point the method throws a system exception. Which of the following actions the container will take?

Please select :

- A. Re-throw the exception
- B. Mark the transaction for rollback
- C. Discard the bean instance (except when the bean is singleton)
- D. Throw an EJBTransactionRolledbackException to the caller

Your answer is incorrect.

Answer: B

Explanation:

The container would have re-thrown the exception only if the exception had been an application exception. Hence, option A is incorrect.

The transaction is started by the container, so it will roll back the transaction instead of just marking for rollback. Thus, option B is incorrect.

The container will throw an EJBException or RemoteException or any of their subclasses rather than EJBTransactionRolledbackException in particular, meaning option D is also incorrect.

Please refer to the EJB 3.1 Specification (subsection 14.3.1) for more details.

The correct answer is: Mark the transaction for rollback

Submit your Feedback/Queries to our Experts

QUESTION 51 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following exceptions the onMessage of a JMS message-driven bean can throw?

Please select :

- A. RemoteException
- B. EJBException
- C. NamingException
- D. None of the above

Your answer is incorrect.

Answer: B

Explanation:

The onMessage of the MessageListener interface does not define any exception in the throws clause. Therefore, it cannot throw any checked exception, such as RemoteException or NamingException.

The correct answer is: EJBException

Submit your Feedback/Queries to our Experts

QUESTION 52 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

In which of the following cases a ConcurrentAccessException is thrown when a loopback call (a situation where an outbound call from a bean results in an inbound call on the same instance) occurs on a singleton session bean?

Please select :

- A. A thread that already holds a Read lock accesses a method associated with a Read lock
- B. A thread that already holds a Read lock accesses a method associated with a Write lock
- C. A thread that already holds a Write lock accesses a method associated with a Read lock
- D. A thread that already holds a Write lock accesses a method associated with a Write lock

Your answer is incorrect.

Answer: B

Explanation:

Here is an excerpt from the EJB 3.1 Specification (subsection 4.8.5.1):

If a loopback call occurs on a Singleton that already holds a Write lock on the same thread:

- If the target of the loopback call is a Read method, the Read lock must always be granted immediately, without releasing the original Write lock.

- If the target of the loopback call is a Write method, the call must proceed immediately, without releasing the original Write lock.

If a loopback call occurs on a Singleton that holds a Read lock on the same thread (but does not also hold a Write lock on the same thread):

- If the target of the loopback call is a Read method, the call must proceed immediately, without releasing the original Read lock.

- If the target of the loopback call is a Write method, an IllegalLoopbackException (a subclass of ConcurrentAccessException) must be thrown to the caller.

The correct answer is: A thread that already holds a Read lock accesses a method associated with a Write lock

Submit your Feedback/Queries to our Experts

QUESTION 53 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a singleton session bean:

```
@Singleton
@Startup
public class Whizlabs {
  @PostConstruct
  void initialize() {
    System.out.println("Initializing");
  }
  public void doSomething() { ... }
  // other declarations
}
```

How many times the string "Initializing" is printed on the console if the doSomething method is invoked twice by local clients, provided the container is running on only one JVM and there is no deployment descriptor?

Please select :

- A. One
- B. Two
- C. Three
- D. Either one or two
- E. Either two or three

Your answer is incorrect.

Answer: A

Explanation:

Since Whizlabs is a singleton bean, it is instantiated just once for the duration of the application. As a result, the @PostConstruct callback method is executed once.

The correct answer is: One

Submit your Feedback/Queries to our Experts

QUESTION 54 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a singleton session bean:

```
@Singleton
public class Whizlabs {
  public Whizlabs() {
    System.out.println("Constructor");
  }
  @PostConstruct
  initialize() {
    System.out.println("Callback");
  }
  // other declarations
}
```

What is printed on the console after the application is deployed in the container, provided there is no associated deployment descriptor?

Please select :

- A. "Constructor" only
- B. "Callback" only
- C. "Constructor", then "Callback"
- D. "Callback", then "Constructor"
- E. Nothing
- F. It is container-specific

Your answer is incorrect.

Answer: F

Explanation:

Without the specification of the @Startup annotation and the equivalent deployment descriptor element, the container is responsible for deciding when to initialize a singleton bean instance. This may or may not be done during the application startup sequence. Thus, you cannot say anything for sure about what is printed after the application is deployed. Note that, however, whenever the bean is instantiated, both "Constructor" and "Callback" are printed, with "Constructor" coming first.

The correct answer is: It is container-specific

Submit your Feedback/Queries to our Experts

QUESTION 55 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a singleton session bean:

```
@Singleton
public class Whizlabs {
   private Properties properties;
   public void setProperties(Properties properties) {
      this.properties = properties
   }
   public Properties getProperties() {
      return this.properties;
   }
   // other declarations
}
```

Which of the following modifications will make the setProperties method accessible to only a single request at a time, while allowing the getProperties to be concurrently invoked my multiple clients (on the condition that there is no request to any other method of the bean and no associated deployment descriptor)?

Please select :

- A. Annotate method getProperties with @Lock(READ)
- B. Annotate method setProperties with @Lock(WRITE)
- C. Annotate the bean class with ConcurrencyManagement(BEAN)
- D. Annotate the bean class with ConcurrencyManagement(CONTAINER)

Your answer is incorrect.

Answer: A

Explanation:

By default, a singleton session bean uses container-managed concurrency, with Write locks for all business methods. As such, the solutions in options B and D do not make any difference.

If the bean class is annotated as ConcurrencyManagement(BEAN), the container allows full concurrent access to the singleton bean instance and the setProperties method is no longer thread-safe. Thus, option C is incorrect.

The correct answer is: Annotate method getProperties with @Lock(READ)

Submit your Feedback/Queries to our Experts

QUESTION 56 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following is a correct way for an enterprise bean method to handle exceptions?

Please select :

- A. If the bean method encounters an error, it should simply propagate the error to the container
- B. If the bean method performs an operation that results in a checked exception that the bean method cannot recover, the bean method should throw the original exception
- C. If the bean method encounters a system exception, it should throw an EJBException
- D. None of the above

Your answer is incorrect.

Answer: A

Explanation:

If the bean method cannot recover from a checked exception, it should throw an EJBException that wraps the original one. The reason is that even the bean method is not able to handle the exception, you cannot expect the client to do that (after the container re-throws the exception to the client).

When a system exception is thrown from within a bean method, it should be propagated to the container for management and reporting. The EJBException should be thrown to the client by the container.

The correct answer is: If the bean method encounters an error, it should simply propagate the error to the container

Submit your Feedback/Queries to our Experts

QUESTION 57 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a message-driven bean with container-managed transaction demarcation. This bean defines a timeout callback method with transaction attribute RequiresNew. This method is declared with an automatically created timer. At some point in the execution, the timeout method throws a system exception. Which of the following statements is correct about the given situation?

Please select :

- A. The container retries to invoke the callback on the same bean instance within the same transaction
- B. The bean is discarded and the timeout callback is not tried again
- C. The bean is discarded, and the container retries the callback on another bean instance
- D. None of the above

Your answer is incorrect.

Answer: C

Explanation:

When a bean method throws a system exception, the bean is discarded and the transaction is marked for rollback. Thus, option A is incorrect.

As per EJB 3.1 Specification (subsection 18.4.3), if container-managed transaction demarcation is used and the REQUIRED or REQUIRES_NEW transaction attribute is specified or defaulted, the container must begin a new transaction prior to invoking the timeout callback method. If the transaction fails or is rolled back, the container must retry the timeout at least once. So, option C is correct while option B is not.

The correct answer is: The bean is discarded, and the container retries the callback on another bean instance

Submit your Feedback/Queries to our Experts

QUESTION 58 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Given a Timer instance created using the following code fragment:
ScheduleExpression schedule = new ScheduleExpression().hour("3/12");
Timer timer = timerService.createCalendarTimer(schedule);
When is the above timer triggered?

Please select :

- A. At 3:00 and 12:00 of every day
- B. At 3:00 and 15:00 of every day
- C. At 00:15 of every day
- D. The given code fragment is invalid

Your answer is incorrect.

Answer: B

Explanation:

The forward slash constrains an attribute based on a starting point and an interval, and is used to specify

"Every N { seconds | minutes | hours } within the { minute | hour | day }" (respectively). For expression x/y, the attribute is constrained to every yth value within the set of allowable values beginning at time x. The x value is inclusive.

Applying to the given scenario, the string "3/12" means that the timer is triggered at 3:00 of everyday and again every 12 hours.

References:

http://docs.oracle.com/javaee/6/api/javax/ejb/ScheduleExpression.html

The correct answer is: At 3:00 and 15:00 of every day

Submit your Feedback/Queries to our Experts

QUESTION 59 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following methods is NOT defined in the TimerService interface to create a programmatic timer?

Please select :

- A. createCalendarTimer
- B. createIntervalTimer
- C. createSingleActionTimer
- D. createTimer
- E. None of the above

Your answer is incorrect.

Answer: E

The TimerService interface defines four methods, with various overloading flavours, to create Timer instances. They are createCalendarTimer, createIntervalTimer, createSingleActionTimer, and createTimer.

References:

http://docs.oracle.com/javaee/6/api/javax/ejb/TimerService.html

The correct answer is: None of the above

Submit your Feedback/Queries to our Experts

QUESTION 60 | NOT ANSWERED | MARK FOR REVIEW | Ask our Experts

Which of the following statements is NOT correct about timers?

Please select :

- A. Non-persistent timers are cancelled in the event of application shutdown
- B. Non-persistent timers can only be created programmatically
- C. When a timer is created programmatically, it is persistent by default
- D. None of the above

Your answer is incorrect.

Answer: B

Explanation:

Unlike persistent timers, non-persistent timers are considered cancelled in the event of application shutdown, container crash, or a failure/shutdown of the JVM on which the timer was started. So, option A is incorrect.

The persistent property of a TimerConfig instance defaults to true. Hence, option C is incorrect.

Non-persistent timers can be created programmatically (by calling setPersistent(false) on the TimerConfig object passed to a timer creation method) or automatically (using @Schedule(persistent=false) or the deployment descriptor). Thus, option B is the correct answer.

The correct answer is: Non-persistent timers can only be created programmatically

Submit your Feedback/Queries to our Experts

Finish review