



OCEEJBD 6 Practice Test

[Home](#) / [My courses](#) / [OCEEJBD 6 Practice Test](#) / [Full Tests](#) / [Practice Test II](#)

0.



Started on Monday, 22 October 2018, 9:14 PM

State Finished

Completed on Monday, 22 October 2018, 9:15 PM

Time taken 16 secs

Grade 0.00 out of 60.00 (0%)

Result

| QUESTION 1 | NOT ANSWERED | Ask our Experts |
|--|---|-----------------|
| Given a bean class declaration: package com.whizlabs; @Stateless(name = "Whizlabs") @Local(WhizlabsInterface.class) public class WhizlabsBean implements WhizlabsInterface { // bean class body } Which TWO of the following are name entries representing the business interface of the above bean in the naming context of the containing module, provided no deployment descriptor is used? Please select : | <input type="checkbox"/> A. Java:module/Whizlabs <input type="checkbox"/> B. java:module/WhizlabsBean <input type="checkbox"/> C. java:module/WhizlabsInterface <input type="checkbox"/> D. java:module/Whizlabs!com.whizlabs.WhizlabsBean <input type="checkbox"/> E. java:module/Whizlabs!com.whizlabs.WhizlabsInterface <input type="checkbox"/> F. java:module/com.whizlabs.WhizlabsBean!WhizlabsInterface | MARK FOR REVIEW |

Your answer is incorrect.

Answer: A and E

Explanation:

As per the EJB 3.1 Specification (subsection 4.4.1), the syntax for a session bean JNDI name is as follows:

java:module/<bean-name>[:<fully-qualified-interface-name>]

If the bean exposes only one client interface (or alternatively has only a no-interface view), the last section, [<fully-qualified-interface-name>], may be left out.

The correct answers are: java:module/Whizlabs, java:module/Whizlabs!com.whizlabs.WhizlabsInterface

[Submit your Feedback/Queries to our Experts](#)

| QUESTION 2 | NOT ANSWERED | Ask our Experts |
|--|--------------|-----------------|
| Which of the following statements is correct about passing arguments to methods of local and remote client views of session beans? Please select : <input checked="" type="radio"/> A. Methods of local business interfaces are passed by reference, while those of remote business interfaces are passed by value <input type="radio"/> B. Methods of local business interfaces are passed by value, while those of remote business interfaces are passed by reference <input type="radio"/> C. Methods of both local and remote business interfaces are passed by reference <input type="radio"/> D. Methods of both local and remote business interfaces are passed by value | | MARK FOR REVIEW |

Your answer is incorrect.

Answer: A

Explanation:

Section 3.2.1 of the EJB 3.1 Specification declares that the arguments and results of the methods of the remote business interface are passed by value, while section 3.2.1 announces the arguments and results of the methods of the local client view are passed by reference.

The correct answer is: Methods of local business interfaces are passed by reference, while those of remote business interfaces are passed by value

[Submit your Feedback/Queries to our Experts](#)

| QUESTION 3 | NOT ANSWERED | Ask our Experts |
|---|--------------|-----------------|
| Given a stateful session bean and two interfaces: | | MARK FOR REVIEW |

```

@Local
public interface MyLocal {
    public int getNumber();
    public void setNumber(int number);
}

@Remote
public interface MyRemote {
    public int getNumber();
    public void setNumber(int number);
}

@Stateful
public class Whizlabs implements MyLocal, MyRemote {
    private int number = 0;
    public int getNumber() {
        return this.number;
    }
    public void setNumber(int number) {
        this.number = number;
    }
}

```

The above bean is used in a local client:

```

@Stateful
public class MyBean {
    @Resource
    private SessionContext context;
    @EJB(name = "local")
    private MyLocal myLocal;
    @EJB(name = "remote")
    private MyRemote myRemote;
    public void changeNumber() {
        MyLocal local1 = (MyLocal) context.lookup("local");
        MyLocal local2 = (MyLocal) context.lookup("local");
        MyRemote remote1 = (MyRemote) context.lookup("remote");
        MyRemote remote2 = (MyRemote) context.lookup("remote");
        local1.setNumber(1);
        remote1.setNumber(1);
        System.out.println(local2.getNumber() + " " + remote2.getNumber());
    }
    // other declarations
}

```

What is printed when the changeNumber method of MyBean is executed?

Please select :

- A. 0 0
- B. 0 1
- C. 1 0
- D. 1 1

Your answer is incorrect.

Answer: A

Explanation:

In case of a stateful session bean, a new instance is returned each time a reference to the bean is requested. As such, variables local1 and local2 reference different instances, the same holds true for variables remote1 and remote2. Any changes made to local1 and remote1, therefore, do not have any effect on local2 and remote2.

The correct answer is: 0 0

[Submit your Feedback/Queries to our Experts](#)

QUESTION 4

NOT ANSWERED

 MARK FOR REVIEW

[Ask our Experts](#)

Which of the following cannot be a local client of a session bean?

Please select :

- A. A stand-alone application running in the same machine as the session bean
- B. Another enterprise bean in the same container as the session bean
- C. A web component in the same server as the session bean
- D. None of the above

Your answer is incorrect.

Answer: A

Explanation:

As per the EJB 3.1 Specification (subsection 3.1), the use of a session bean's local client view entails the collocation of the local client and the session. The local client of an enterprise bean must be colocated in the same container as the bean. The local client view is not location-independent.

Subsection 3.2.2 of the Specification also notes that a local client of a session bean may be another enterprise bean or a web component.

The correct answer is: A stand-alone application running in the same machine as the session bean

Submit your Feedback/Queries to our Experts

QUESTION 5

NOT ANSWERED

Ask our Experts

Given a valid session bean in the default package:

```
@Stateless  
public class Whizlabs {  
    @Resource(name = "jdbc/myDS")  
    private DataSource dataSource;  
    @Resource  
    private SessionContext sessionContext;  
    public void doSomething() {  
        DataSource myDataSource = (DataSource) /* insert-here */;  
        // do something with myDataSource  
    }  
    // other declarations  
}
```

Which TWO of the following statements can be put at /* insert-here */ to obtain a DataSource instance, provided no deployment descriptor elements are relevant?

Please select :

- A. sessionContext.lookup("jdbc/myDS")
- B. sessionContext.lookup("Whizlabs(dataSource")
- C. sessionContext.lookup("java:module/Whizlabs(dataSource")
- D. sessionContext.lookup("java:dbc/myDS")
- E. sessionContext.lookup("java:comp/env/jdbc/myDS")

Your answer is incorrect.

Answer: A and E

Explanation:

After a resource is bound to the component namespace, it can be looked up using a full name (option E), or relative name (option A).

Note that the name in option B corresponds to the default name of the resource. In the given scenario, however, the name element of the @Resource annotation is explicitly set to another value, meaning that the default name is unavailable in the naming context.

The correct answers are: sessionContext.lookup("jdbc/myDS"), sessionContext.lookup("java:comp/env/jdbc/myDS")

Submit your Feedback/Queries to our Experts

QUESTION 6

NOT ANSWERED

Ask our Experts

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

Please select :

- A. name
- B. beanInterface
- C. description
- D. mappedName
- E. shareable

Your answer is incorrect.

Answer: B

Explanation:

The @Resource annotation defines the following elements:

- authenticationType: The authentication type to use for this resource
- description: Description of this resource
- lookup: The name of the resource that the reference points to
- mappedName: A product specific name that this resource should be mapped to
- name: The JNDI name of the resource
- shareable: Indicates whether this resource can be shared between this component and other components
- type: The Java type of the resource

References:

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

The correct answer is: beanInterface

Submit your Feedback/Queries to our Experts

QUESTION 7

NOT ANSWERED

Ask our Experts

Which TWO of the following statements are correct about session beans?

Please select :

- A. A session bean can provide a local and a remote client view through the same interface
- B. The container can handle multiple client requests to the same stateless session bean simultaneously
- C. A stateful session bean instance can be reused for multiple clients
- D. A client cannot destroy a singleton session bean

Your answer is incorrect.

Answer: B and D

Explanation:

An interface can be exposed as either a local or a remote client view, but not both. Thus, option A is incorrect.

After being used for a client, a stateful session bean is removed. Therefore, it cannot be reused for any other client. Option C is incorrect, then.

The container can handle multiple client accesses to the same stateless session bean at the same time by dispatching those requests to different instances of the same bean class. Hence, option B is correct.

A successfully initialized singleton session bean can only be destroyed by the container during application shutdown. So, option D is correct as well.

The correct answers are: The container can handle multiple client requests to the same stateless session bean simultaneously. A client cannot destroy a singleton session bean

Submit your Feedback/Queries to our Experts

QUESTION 8

NOT ANSWERED

Ask our Experts

Given a stateful session bean:

```
@Stateful  
public class Whizlabs {  
    // some annotations  
    public void discardBean() {  
        // do something  
    }  
    // some annotations  
    public void releaseResources() {  
        // release resources  
    }  
    // other declarations  
}
```

The discardBean method is called by a client to instruct the container to discard its servicing bean instance. Meanwhile, the releaseResources method cleans up all resources that are no longer needed after the bean is discarded.

Which of the following is the correct way to achieve what is described above?

Please select :

- A. Annotate method discardBean with @Remove and releaseResources with @PreDestroy
- B. Annotate method discardBean with @PreDestroy and releaseResources with @Remove
- C. Annotate method releaseResources with @Remove and @PreDestroy, invoke the releaseResources method from within the discardBean method
- D. Annotate method discardBean with @DiscardBean and releaseResources with @Remove

Your answer is incorrect.

Answer: A

Explanation:

As per the Java EE 6 API documentation, the @Remove annotation is applied to a business method of a stateful session bean class to indicate to the container that the stateful session bean is to be removed by the container after completion of the method.

References:

<http://docs.oracle.com/javaee/6/api/javax/ejb/Remove.html>

The correct answer is: Annotate method discardBean with @Remove and releaseResources with @PreDestroy

Submit your Feedback/Queries to our Experts

QUESTION 9

NOT ANSWERED

Ask our Experts

Given a session bean:

```
@Stateless  
@Resource(name = "jdbc/myDS", type = javax.sql.DataSource.class)  
public class Whizlabs {  
    // bean class body  
}
```

When a DataSource instance is injected into the Whizlabs bean?

Please select :

- A. When the application is deployed
- B. When the bean is instantiated
- C. When a DataSource instance is needed at runtime

- D. It depends on the container implementation

Your answer is incorrect.

Answer: C

Explanation:

As per the Java EE 6 API documentation, when the @Resource annotation is applied to a field or method, the container will inject an instance of the requested resource into the application component when the component is initialized. If the annotation is applied to the component class, the annotation declares a resource that the application will look up at runtime.

References:

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

The correct answer is: When a DataSource instance is needed at runtime

Submit your Feedback/Queries to our Experts

QUESTION 10

NOT ANSWERED

Ask our Experts

Given a session bean declaration:

```
@Stateless  
public class WhizlabsBean {  
    public WhizlabsBean() {  
        // some initialization logic  
    }  
    @PostConstruct  
    public void initialize() {  
        // more initialization logic  
    }  
    // other declarations  
}
```

Which of the following statements is correct about the above bean when a method of its no-interface view is invoked?

Please select :

- A. Only the initialization logic in the no-argument constructor is executed
- B. Only the initialization logic in the initialize method is executed
- C. An EJBException is thrown
- D. The given declaration may lead to unpredictable consequences

Your answer is incorrect.

Answer: D

Explanation:

The EJB 3.1 Specification (subsection 3.4.4) declares: The developer of an enterprise bean that exposes a no-interface view must not make any assumptions about the number of times the bean class no-arg constructor will be called. For example, it is possible that the acquisition of a client reference to the no-interface view will result in the invocation of the bean-class constructor. It is recommended that the bean developer place component initialization logic in a @PostConstruct method instead of the bean class no-arg constructor.

The correct answer is: The given declaration may lead to unpredictable consequences

Submit your Feedback/Queries to our Experts

QUESTION 11

NOT ANSWERED

Ask our Experts

Given a scenario where a client receives an EJBException when invoking an asynchronous method of a session bean client view. Which of the following can be the cause of the exception?

Please select :

- A. The container does not support asynchronous method invocation
- B. The container has not dispatched the asynchronous method due to a certain internal issue
- C. The container has dispatched the asynchronous method, and the method execution results in an exception
- D. None of the above

Your answer is incorrect.

Answer: B

Explanation:

The EJB 3.1 Specification (subsection 3.4.8) declares:

The client should expect to receive a system exception (in the form of a javax.ejb.EJBException) on the client thread if the container has problems allocating the internal resources required to support the asynchronous method. If a system exception is received on the client thread, the client can expect that the container will not be able to dispatch the asynchronous method. The client may wish to retry the asynchronous method at a later time.

If no system exception is received, then the client can expect that the container will make an attempt to dispatch the asynchronous method. An exception resulting from the asynchronous method execution (e.g. an authorization failure, transaction commit failure, application exception, etc.) will be available via the Future<V> object.

The correct answer is: The container has not dispatched the asynchronous method due to a certain internal issue

Submit your Feedback/Queries to our Experts

QUESTION 12

NOT ANSWERED

[Ask our Experts](#)

Given a valid stateful session bean:
 @Stateful
 public class Whizlabs {
 @PostConstruct
 /* <access-modifier> */ void initialize() { ... }
 @Remove
 /* <access-modifier> */ void remove() { ... }
 @PreDestroy
 /* <access-modifier> */ void cleanUp() { ... }
 // other declarations
 }

Which access modifiers should be assigned to the above methods?

Please select :

- A. All three methods must be public
- B. remove must be public, while initialize and cleanUp should be non-public
- C. remove and cleanUp should be public, while initialize should be non-public
- D. initialize must be public, remove should be public, while cleanUp should be non-public
- E. All three methods should be non-public

Your answer is incorrect.

Answer: B

Explanation:

A @Remove method is designated to be called by clients. It must be part of an exposed client view, thus must be public.

Lifecycle callback methods are called by the container during the initialization and clean-up processes of a bean instance. Therefore, it should not be exposed to clients. As a result, methods initialize and cleanUp should not be public.

The correct answer is: remove must be public, while initialize and cleanUp should be non-public

[Submit your Feedback/Queries to our Experts](#)

QUESTION 13

NOT ANSWERED

[Ask our Experts](#)

Given a scenario where a client invokes a method on a stateful session bean's business interface and receives a NoSuchEJBException. Which TWO of the following are possible reasons for this exception?

Please select :

- A. The container has not generated an implementation for the business interface
- B. The container has removed the bean instance after the expiration of a timeout
- C. The session bean instance is in the passivated state
- D. The bean instance has been removed as a result of the client invoking a @Remove method

Your answer is incorrect.

Answer: B and D

Explanation:

The container provides an implementation of a session bean's business interface such that when the client invokes a method on the instance of the business interface, the business method on the session bean instance and any interceptor methods are invoked as needed. Thus, the situation described in option A never happens.

If a client invokes a session object whose session bean instance has been passivated, the container will activate the instance. Hence, option C is incorrect as well.

As per the Java EE 6 API documentation, a NoSuchEJBException is thrown if an attempt is made to invoke a business method on a stateful session or singleton object that no longer exists.

References:

<http://docs.oracle.com/javaee/6/api/javax/ejb/NoSuchEJBException.html>

The correct answers are: The container has removed the bean instance after the expiration of a timeout. The bean instance has been removed as a result of the client invoking a @Remove method

[Submit your Feedback/Queries to our Experts](#)

QUESTION 14

NOT ANSWERED

[Ask our Experts](#)

Which of the following is NOT a valid value for the acknowledgment mode when creating a JMS non-transacted session?

Please select :

- A. AUTO_ACKNOWLEDGE
- B. CLIENT_ACKNOWLEDGE
- C. DUPS_OK_ACKNOWLEDGE
- D. SESSION_TRANSACTED
- E. None of the above

Your answer is incorrect.

Answer: D

Explanation:

The Session interface defines four constant field embodying the acknowledgment mode of a session, including:

- AUTO_ACKNOWLEDGE: With this acknowledgment mode, the session automatically acknowledges a client's receipt of a message either when the session has successfully returned from a call to receive or when the message listener the session has called to process the message successfully returns.
- CLIENT_ACKNOWLEDGE: With this acknowledgment mode, the client acknowledges a consumed message by calling the message's acknowledgment method.
- DUPS_OK_ACKNOWLEDGE: This acknowledgment mode instructs the session to lazily acknowledgment the delivery of messages.
- SESSION_TRANSACTED: This value is returned from the method getAcknowledgeMode if the session is transacted.

The SESSION_TRANSACTED mode is only used for a demonstration purpose when the Session.getAcknowledgeMode method is called. It cannot be passed as an argument to the Connection.createSession method.

References:

<http://docs.oracle.com/javaee/6/api/javax/jms/Session.html>

The correct answer is: SESSION_TRANSACTED

Submit your Feedback/Queries to our Experts

QUESTION 15

NOT ANSWERED

Ask our Experts

MARK FOR REVIEW

Given a JMS Session and two Queue objects, referenced by variables session, queue1 and queue2, respectively. What happens when the following code fragment is executed?

```
MessageProducer producer = session.createProducer(queue1);
TextMessage message = session.createTextMessage();
message.setText("Whizlabs");
producer.send(queue2, message);
```

Please select :

- A. The message is sent to queue1
- B. The message is sent to queue2
- C. AJMSEException is thrown
- D. An UnsupportedOperationException is thrown

Your answer is incorrect.

Answer: D

Explanation:

The overloaded send method of the MessageProducer interface that specifies the destination as the first parameter is used for unidentified producers only. That is, this method just applies if you pass null as the first argument to the Session.createProducer method. Invoking method send(Destination, Message) on an identified producer will result in an UnsupportedOperationException. Note that this exception is also thrown if a destination is not passed in when invoking the send method on an unidentified producer.

References:

[http://docs.oracle.com/javaee/6/api/javax/jms/MessageProducer.html#send\(javax.jms.Destination, javax.jms.Message\)](http://docs.oracle.com/javaee/6/api/javax/jms/MessageProducer.html#send(javax.jms.Destination, javax.jms.Message))

The correct answer is: An UnsupportedOperationException is thrown

Submit your Feedback/Queries to our Experts

QUESTION 16

NOT ANSWERED

Ask our Experts

MARK FOR REVIEW

Which of the following statements is correct about the use of queues and topics in a JMS application?

Please select :

- A. An application can use multiple queues and multiple topics at the same time
- B. An application can use either multiple queues or multiple topics, but not both
- C. An application can use both queues and topics; if this is the case, no more than one queue and one topic is allowed; there are no constraints on the number of queues or topics if just one type of destinations is used
- D. An application can use either one queue or one topic

Your answer is incorrect.

Answer: A

Explanation:

There are no constraints on the number of queues and topics a JMS application can use. You may combine both point-to-point, which uses queues, and publish/subscribe, which uses topics, in a single application.

The correct answer is: An application can use multiple queues and multiple topics at the same time

Submit your Feedback/Queries to our Experts

QUESTION 17

NOT ANSWERED

Ask our Experts

MARK FOR REVIEW

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

Please select :

- A. activationConfig
- B. listenerMethod

- B. messageMethod
- C. messageListenerInterface
- D. name
- E. mappedName

Your answer is incorrect.

Answer: B

Explanation:

The @MessageDriven annotation defines five elements shown below:

- activationConfig: Activation config properties
- description: A string describing the message driven bean
- mappedName: A product specific name (e.g. global JNDI name of a queue)
- messageListenerInterface: Message-listener interface
- name: The ejb-name for this bean

The message listener method is defined by the listener interface, not by metadata annotations or the deployment descriptor.

The correct answer is: listenerMethod

[Submit your Feedback/Queries to our Experts](#)

QUESTION 18

NOT ANSWERED

[Ask our Experts](#)

What happens if the message listener method of a message-driven bean throws a system exception when handling a message?

Please select :

- A. The message is re-delivered from the destination
- B. The @PreDestroy callback method of the bean, if existent, is executed to clean up resources
- C. The client stops sending messages
- D. None of the above

Your answer is incorrect.

Answer: A

Explanation:

When a system exception is thrown, the bean instance is discarded and the message is not acknowledged (by the container if the bean uses bean-managed transaction demarcation, or as part of the transaction commit if it uses container-managed transaction demarcation), meaning that the message is delivered again. Therefore, option A is the correct answer.

The @PreDestroy callback method is just invoked when the bean is destroyed in "normal" conditions. It is not called when the EJB container crashes or a system exception is thrown. Thus, option B is incorrect.

From the client perspective, the message consumer continues to exist in the event of the bean destruction. So, it keeps sending new messages to the same destination, where those messages are delegated to another bean instance.

The correct answer is: The message is re-delivered from the destination

[Submit your Feedback/Queries to our Experts](#)

QUESTION 19

NOT ANSWERED

[Ask our Experts](#)

Given a valid code fragment executed by the client of a messaging application:

```
userTransaction.begin();
// make updates to database 1
producer.send(message);
userTransaction.commit();
```

The producer.send(message) expression sends a message to a queue and asks the message consumer, which is a message-driven bean, to extract content of the message. This message-driven bean uses bean-managed transaction demarcation, executing the following code fragment within the message listener method:

```
userTransaction.begin();
// update database 2 with received messages
// a system exception is thrown here
userTransaction.commit();
```

Which databases are updated after the producer.send(message) method is invoked in the client?

Please select :

- A. Database 1 only
- B. Database 2 only
- C. Both databases
- D. Neither of the databases

Your answer is incorrect.

Answer: A

Explanation:

A system exception is thrown within the message listener method of the message-driven bean, thus the bean is discarded and the transaction in which it is running is rolled back. This means database 2 is not updated.

The client of a message-driven bean does not have a direct connection with the bean and is unaware of anything happening to it. Therefore, database 1 is updated successfully.

The correct answer is: Database 1 only

Submit your Feedback/Queries to our Experts

QUESTION 20

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Which of the following statements is correct about a message-driven bean's destination and its clients?

Please select :

- A. Clients must be located on the same machines as the message destination
- B. A client can only reference to the message destination by resource injection
- C. A client can only reference to the message destination by lookup in the client's JNDI namespace
- D. None of the above

Your answer is incorrect.

Answer: D

Explanation:

As per the EJB 3.1 Specification (subsection 5.3), a client's JNDI namespace may be configured to include the destinations or endpoints of message-driven beans installed in multiple EJB containers located on multiple machines on a network. The actual locations of an enterprise bean and EJB container are, in general, transparent to the client using the enterprise bean. References to message destinations can be injected, or they can be looked up in the client's JNDI namespace.

The correct answer is: None of the above

Submit your Feedback/Queries to our Experts

QUESTION 21

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Given an interceptor class containing an around-invoke method. This method is associated with a stateless session bean method. Assume the bean method is invoked by two clients: once by the first client and twice by the second. How many instances of the interceptor class are created?

Please select :

- A. One
- B. Two
- C. Three
- D. One or two
- E. Two or three
- F. One, two or three

Your answer is incorrect.

Answer: F

Explanation:

The lifecycle of an interceptor class instance is the same as that of the target class instance with which it is associated. Therefore, the number of interceptor instances is the same as the number of bean instances. However, such bean instances can be pooled or created/destroyed, depending on the container's decision. As a result, you cannot know how many instances of the bean are used to service those two clients.

The correct answer is: One, two or three

Submit your Feedback/Queries to our Experts

QUESTION 22

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Given an enterprise bean and its super class:

```
public class Whizlabs {
    @AroundInvoke
    public Object interceptWhizlabs(InvocationContext ctx) throws Exception {
        System.out.print("Whizlabs ");
    }
}
@Stateful
@Interceptors(InterceptorA.class)
public class MyBean extends Whizlabs implements A {
    @Interceptors(InterceptorB.class)
    public void doSomething() { ... }
    @AroundInvoke
    public Object interceptMyBean(InvocationContext ctx) throws Exception {
        System.out.print("MyBean ");
    }
    // other declarations
}
```

The following are the declarations of InterceptorA and InterceptorB:

```
@Interceptor
public class InterceptorA {
```

```

@AroundInvoke
public Object interceptA(InvocationContext ctx) throws Exception {
    System.out.print("InterceptorA ");
}
}

@Interceptor
public class InterceptorB {
    @AroundInvoke
    public Object interceptB(InvocationContext ctx) throws Exception {
        System.out.print("InterceptorB ");
    }
}

```

Which of the following gets printed on the console when the MyBean.doSomething method is invoked, provided no associated deployment descriptor is existent?

Please select :

- A. InterceptorB InterceptorA MyBean Whizlabs
- B. InterceptorA InterceptorB MyBean Whizlabs
- C. InterceptorB InterceptorA Whizlabs MyBean
- D. InterceptorA InterceptorB Whizlabs MyBean
- E. Whizlabs MyBean InterceptorB InterceptorA
- F. MyBean Whizlabs InterceptorA InterceptorB

Your answer is incorrect.

Answer: D

Explanation:

As per the Interceptors 1.1 Specification (subsection Multiple Method Interceptor Methods), interceptor methods defined on interceptor classes are invoked first, then, order:

- If any method level interceptors are defined for the target class method that is to be invoked, the methods defined on those interceptor classes are invoked in the same order as the specification of those interceptor classes in the Interceptors annotation applied to that target class method.
- If a target class has superclasses, any around-invoke methods defined on those superclasses are invoked, most general superclass first.
- The around-invoke method, if any, on the target class itself is invoked.

The correct answer is: InterceptorA InterceptorB Whizlabs MyBean

[Submit your Feedback/Queries to our Experts](#)

QUESTION 23

NOT ANSWERED

Ask our Experts

Given an interceptor class:

```

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

```

The above interceptor class is specified in a bean:

```

@Stateless
public class Whizlabs implements A {
    @Interceptors(MyInterceptor.class)
    public int sum(int num1, int num2) {
        return num1 + num2;
    }

    @AroundInvoke
    Object intercept(InvocationContext context) throws Exception {
        System.out.print("Inside");
        return context.proceed();
    }

    // other declarations
}

```

What is printed when a client invokes the Whizlabs.sum method with arguments 1 and 2?

Please select :

- A. 3
- B. "Inside"
- C. "Outside"
- D. "InsideOutside"
- E. "OutsideInside"

Your answer is incorrect.

Answer: E

Explanation:

Both interceptor methods shown above interpose on the invocation of the sum method. Since methods in an interceptor class are invoked before methods in the target class itself, the string "Outside" gets printed before "Inside".

The correct answer is: "OutsideInside"

[Submit your Feedback/Queries to our Experts](#)

QUESTION 24**NOT ANSWERED****Ask our Experts**

Which of the following statements is NOT correct about the @Interceptors annotation?

Please select :

- A. When the @Interceptors annotation applies to a method of a superclass, it applies to inherited methods in subclasses as well
- B. All interceptor methods can be specified by the @Interceptors annotation on both class-level and method-level
- C. If multiple interceptors are defined for a target class, the order in which they are invoked is determined by the order in which they are specified in the @Interceptors annotation
- D. The @Interceptors annotation has no attribute other than value that refers to an ordered list of interceptors

Your answer is incorrect.**Answer: B**

Explanation:

As per the Java EE 6 API documentation, only method interception or timeout method interception may be specified by a method-level @Interceptors declaration. This means that lifecycle event callback interceptors cannot be indicated by decorating a method with @Interceptors.

References:

<http://docs.oracle.com/javaee/6/api/javax/interceptor/Interceptors.html>

The correct answer is: All interceptor methods can be specified by the @Interceptors annotation on both class-level and method-level

[Submit your Feedback/Queries to our Experts](#)

QUESTION 25**NOT ANSWERED****Ask our Experts**

Given an enterprise bean:

@Stateless

```
public class Whizlabs {
    @RolesAllowed("group1")
    public void methodA() { ... }
    public void methodB() { ... }
    // other declarations
}
```

The following is a snippet taken from the EJB deployment descriptor:

```
<method-permission>
    <role-name>group2</role-name>
    <method>
        <ejb-name>Whizlabs</ejb-name>
        <method-name>methodA</method-name>
    </method>
</method-permission>
```

Which of the statements given below is correct?

Please select :

- A. Only clients assigned to role group1 can invoke methodA
- B. Only clients assigned to role group2 can invoke methodA
- C. No clients can invoke methodB
- D. None of the above

Your answer is incorrect.**Answer: B**

Explanation:

The method-permission element in the deployment descriptor overrides the values specified in the @RolesAllowed annotation, thus option B is correct while option A is not.

There are no security constraints on the methodB method, making it accessible to all clients. Option C is incorrect, then.

The correct answer is: Only clients assigned to role group2 can invoke methodA

[Submit your Feedback/Queries to our Experts](#)

QUESTION 26**NOT ANSWERED****Ask our Experts**

Which of the following statements is correct about securing an EJB application?

Please select :

- A. Security policies should be programmatically specified to maximize the bean provider's ability with respect to security management; the deployment descriptor

is used only in the case bean code is unable to handle

- B. A security role is, in fact, a user group in the operational environment
- C. The security principal under which a method invocation is performed is that of the component's caller
- D. Once a set of security roles is defined with metadata annotations, it can only be either mapped to the operational environment as is, or completely overridden using the deployment descriptor
- E. None of the above

Your answer is incorrect.

Answer: E

Explanation:

As per the EJB 3.1 Specification (subsection 17.1), the EJB architecture encourages the bean provider to implement the enterprise bean class without hard-coding the security policies and mechanisms into the business methods. In most cases, the enterprise bean's business methods should not contain any security-related logic. Option A is incorrect, then.

Assignment of security roles is a job of the deployer or system administrator. Each role is usually mapped to a user group in the operational environment, but this is not a requirement. Security roles and user groups are two independent things. Hence, option B is incorrect.

The security principal under which a method invocation is performed is typically, not always, that of the component's caller. By specifying a run-as identity, it is possible to specify that a different principal be substituted for the execution of the methods of the bean. As such, option C is incorrect.

The set of security roles used by the application is taken to be the aggregation of the security roles defined by the security role names specified in the @DeclareRoles and @RolesAllowed annotations. The bean provider may augment the set of security roles defined for the application by annotations in this way by means of the security-role deployment descriptor element. Therefore, option D is incorrect.

The correct answer is: None of the above

[Submit your Feedback/Queries to our Experts](#)

QUESTION 27

[NOT ANSWERED](#)

[Ask our Experts](#)

Given a fragment of the EJB deployment descriptor:

```
<session>
  <ejb-name>Whizlabs</ejb-name>
  <ejb-class>com.Whizlabs</ejb-class>
  <security-role-ref>
    <role-name>my-name</role-name>
    <role-link>my-link</role-link>
  </security-role-ref>
  <!-- other elements -->
</session>
```

Which of the following statements is correct?

Please select :

- A. One of the security roles defined in the security-role element of the deployment descriptor must be my-link
- B. The declaration of bean Whizlabs must be decorated with the @DeclareRoles("my-name") annotation
- C. The role-link element in the above fragment can be removed if the Whizlabs bean is annotated with RolesAllowed("my-name")
- D. None of the above

Your answer is incorrect.

Answer: C

Explanation:

The value of the role-link element refers to a security role used by the application, which can be defined by the security role names specified in the @DeclareRoles or @RolesAllowed annotation, as well as the security-role element in the deployment descriptor. So, option A is incorrect.

The security-role-ref element has been set in the deployment descriptor, thus, the @DeclareRoles annotation becomes useless. As such, option B is incorrect.

If a security role reference is not linked to a security role, the container will map the reference name to the security role of the same name. By declaring the RolesAllowed("my-name") annotation, you can map a role reference named my-name in the bean code to the security role my-name.

The correct answer is: The role-link element in the above fragment can be removed if the Whizlabs bean is annotated with RolesAllowed("my-name")

[Submit your Feedback/Queries to our Experts](#)

QUESTION 28

[NOT ANSWERED](#)

[Ask our Experts](#)

Which of the following is NOT a goal of the Java EE security architecture?

Please select :

- A. Transparency: Application component providers should not have to know anything about security to write an application
- B. Extensibility: The use of platform services by security-aware applications must not compromise application portability
- C. Flexibility: The security mechanisms and declarations used by applications should not impose a particular security policy
- D. Secure interoperability: Application components must be able to invoke services provided in a Java EE product from a different vendor, whether with the same or a different security policy
- E. None of the above

Your answer is incorrect.

Answer: E

Explanation:

As per the Java EE 6 Platform Specification, the Java EE security architecture has 9 goals:

1. Portability
2. Transparency
3. Isolation
4. Extensibility
5. Flexibility
6. Abstraction
7. Independence
8. Compatibility testing
9. Secure interoperability

The correct answer is: None of the above

Submit your Feedback/Queries to our Experts

QUESTION 29

NOT ANSWERED

[Ask our Experts](#)[MARK FOR REVIEW](#)

Which of the following methods of stateless session beans cannot invoke methods of the SessionContext interface?

Please select :

- A. @PostConstruct methods
- B. @PreDestroy methods
- C. Business methods from a business interface
- D. Business methods from web service endpoint
- E. Timeout callback methods
- F. None of the above

Your answer is incorrect.**Answer: F****Explanation:**

The SessionContext interface supplies access to the runtime session context that the container provides to a session bean instance and the interface is accessible to all methods of the bean.

The correct answer is: Timeout callback methods

Submit your Feedback/Queries to our Experts

QUESTION 30

NOT ANSWERED

[Ask our Experts](#)[MARK FOR REVIEW](#)

Which TWO of the following can provide a web service client view?

Please select :

- A. Stateless session beans
- B. Stateful session beans
- C. Singleton session beans
- D. Message-driven beans

Your answer is incorrect.**Answer: A and C****Explanation:**

As per the EJB 3.1 Specification (subsection 3.1), the client of a stateless session bean or singleton session bean may be a web service client. Only a stateless session bean or singleton session bean may provide a web service client view.

The correct answers are: Stateless session beans, Singleton session beans

Submit your Feedback/Queries to our Experts

QUESTION 31

NOT ANSWERED

[Ask our Experts](#)[MARK FOR REVIEW](#)

Given a stateless session bean:

```
@Stateless  
@WebService(serviceName = "WhizlabsService")  
public class WhizlabsBean implements Whizlabs {  
    @WebMethod  
    public double methodA(int arg) { ... }  
    public String methodB(String arg) { ... }  
    // other declarations  
}
```

Which methods shown above are exposed as part of a web service client view?

Please select :

- A. methodA only

- B. methodB only
- C. Both methodA and methodB
- D. Neither methodA nor methodB

Your answer is incorrect.

Answer: A

Explanation:

Generally, a business method that is exposed to web service clients must be annotated with @WebMethod, means that only methodA is part of the web service client view. Note that if no methods are annotated with @WebMethod, all public methods are exposed to clients.

The correct answer is: methodA only

[Submit your Feedback/Queries to our Experts](#)

QUESTION 32

[NOT ANSWERED](#)

[Ask our Experts](#)

Given an enterprise bean:

```
@Stateless
public class WhizlabsBean {
    public void localMethod { ... }
    public void remoteMethod { ... }
    // other declarations
}
```

And two interfaces:

```
public interface MyLocal {
    public void localMethod();
}

public interface MyRemote {
    public void remoteMethod();
}
```

Which of the following modifications allows the WhizlabsBean bean to expose a local business interface, a remote business interface and a no-interface client view?

Please select :

- A. Annotate MyLocal and MyRemote with @Local and @Remote, respectively
- B. Annotate WhizlabsBean with @Local(MyLocal.class) and @Remote(MyRemote.class)
- C. Declare MyLocal and MyRemote in the implements clause of WhizlabsBean, and annotate this bean with @Remote(MyRemote.class)
- D. None of the above

Your answer is incorrect.

Answer: D

Explanation:

If you annotate an interface with @Local or @Remote without declaring it in the implements clause of a bean class, the interface is unrelated to the bean. Adding annotations as described in option A has nothing to do with WhizlabsBean, and this bean ends up with exposing only a no-interface view.

When annotating WhizlabsBean with @Local(MyLocal.class) and @Remote(MyRemote.class), the bean exposes a local and a remote business interface, but the no-interface view is no longer exposed. Thus, option B is incorrect.

When declaring MyLocal and MyRemote in the implements clause of WhizlabsBean, and annotate this bean with @Remote(MyRemote.class), the bean exposes only a remote business interface. It does not expose a local business interface or no-interface view.

To make WhizlabsBean expose three client views as required, MyLocal and MyRemote must be explicitly specified as local and remote business interfaces, respectively, and the bean class must be annotated with @LocalBean or the local-bean element must be specified in the deployment descriptor.

The correct answer is: None of the above

[Submit your Feedback/Queries to our Experts](#)

QUESTION 33

[NOT ANSWERED](#)

[Ask our Experts](#)

Which of the following statements is NOT correct about enterprise beans with bean-managed transaction demarcation?

Please select :

- A. A bean instance that starts a transaction must complete the transaction before it starts a new transaction
- B. While a bean instance is in a transaction, the instance must not attempt to use the resource-manager specific transaction demarcation API
- C. A session bean instance must commit a transaction before a business method or timeout callback method returns
- D. A message-driven bean instance must commit a transaction before a message listener method or timeout callback method returns

Your answer is incorrect.

Answer: C

Explanation:

The following information is taken from the EJB 3.1 Specification (subsection 13.3.3):

An instance that starts a transaction must complete the transaction before it starts a new transaction. The reason is that the Enterprise JavaBeans architecture supports flat transactions. A flat transaction cannot have any child (nested) transactions.

While an instance is in a transaction, the instance must not attempt to use the resource-manager specific transaction demarcation API (e.g. it must not invoke the commit

or rollback method on the java.sql.Connection interface or on the javax.jms.Session interface).

A stateful session bean instance may, but is not required to, commit a started transaction before a business method returns. If a transaction has not been completed by the end of a business method, the container retains the association between the transaction and the instance across multiple client calls until the instance eventually completes the transaction.

A stateless session bean instance must commit a transaction before a business method or timeout callback method returns.

A singleton session bean instance must commit a transaction before a business method or timeout callback method or PostConstruct/PreDestroy lifecycle callback interceptor method returns.

A message-driven bean instance must commit a transaction before a message listener method or timeout callback method returns.

The correct answer is: A session bean instance must commit a transaction before a business method or timeout callback method returns

[Submit your Feedback/Queries to our Experts](#)

QUESTION 34

NOT ANSWERED

[Ask our Experts](#)

[MARK FOR REVIEW](#)

Given an enterprise bean with bean-managed transaction demarcation. Which TWO of the following methods can be called within a method of this bean to ensure the transaction associated with the current thread cannot be committed?

Please select :

- A. EJBContext.rollback
- B. EJBContext.setRollbackOnly
- C. UserTransaction.rollback
- D. UserTransaction.setRollbackOnly

Your answer is incorrect.

Answer: C and D

Explanation:

The EJBContext.setRollbackOnly can only be used in a bean with container-managed transaction demarcation, while there is no rollback method defined by the EJBContext interface. Thus, options A and B are incorrect.

The UserTransaction interface defines two methods to require the current transaction not to roll back: rollback rolls back the transaction, while setRollbackOnly modifies the transaction such that the only possible outcome is to roll back the transaction.

The correct answers are: UserTransaction.rollback, UserTransaction.setRollbackOnly

[Submit your Feedback/Queries to our Experts](#)

QUESTION 35

NOT ANSWERED

[Ask our Experts](#)

[MARK FOR REVIEW](#)

Given a scenario where a method of a stateless session bean with bean-managed transaction demarcation is invoked by a local client. This method starts a transaction, but the transaction is not explicitly committed or rolled back in the method as well as in any interceptor and callback methods. Which of the following is NOT an action of the container in such a situation?

Please select :

- A. Logs an application error
- B. Rolls back the started transaction
- C. Discards the bean instance
- D. Throws the an EJBException
- E. None of the above

Your answer is incorrect.

Answer: E

Explanation:

As per the EJB 3.1 Specification (subsection 13.6.1), if a stateless or singleton session bean instance starts a transaction in a business method or interceptor method, it must commit the transaction before the business method (or all its interceptor methods) returns. The container must detect the case in which a transaction was started, but not completed, in the business method or interceptor method for the business method, and handle it as follows:

- Log this as an application error to alert the system administrator.
- Roll back the started transaction.
- If this a stateless session bean, discard the bean instance.
- Throw the javax.ejb.EJBException.

The correct answer is: None of the above

[Submit your Feedback/Queries to our Experts](#)

QUESTION 36

NOT ANSWERED

[Ask our Experts](#)

[MARK FOR REVIEW](#)

Given four methods in stateless session beans with container-managed transaction demarcation. These methods have the following transaction attributes:
methodA: Supports
methodB: Required
methodC: Mandatory
methodD: NotSupported

Which of the following invocation chains is correct if the client calling methodA does not run within a transaction?
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 (T1) -> methodC (T2) -> methodD (No)
- B. methodA (T1) -> methodB (T2) -> methodC (T2) -> methodD (No)
- C. methodA (No) -> methodB (T1) -> methodC (T1) -> methodD (No)
- D. methodA (No) -> methodB (T1) -> methodC (T1) -> methodD (T1)

Your answer is incorrect.

Answer: C

Explanation:

The transaction attribute of methodC is Mandatory, implying methodC must execute in the same transaction as methodB. Thus, option A is incorrect.

The transaction attribute of methodB is Required, therefore methodB must run within the same transaction as methodA. So, option B is incorrect as well.

The transaction attribute of methodD is NotSupported, so methodD must not run within a transaction. Option D is incorrect, therefore.

References:

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

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

Submit your Feedback/Queries to our Experts

QUESTION 37

NOT ANSWERED

Ask our Experts

Which TWO of the following transaction attributes can be used on message listener methods of message-driven beans with container-managed transaction demarcation?

Please select :

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

Your answer is incorrect.

Answer: B and E

Explanation:

As per the EJB 3.1 Specification (subsection 13.6.3), Only the NOT_SUPPORTED and REQUIRED transaction attributes may be used for message-driven bean message listener methods. The use of the other transaction attributes is not meaningful for message-driven bean message listener methods because there is no pre-existing client transaction context (REQUIRES_NEW, SUPPORTS) and no client to handle exceptions (MANDATORY, NEVER).

The correct answers are: Required, NotSupported

Submit your Feedback/Queries to our Experts

QUESTION 38

NOT ANSWERED

Ask our Experts

Given two class declarations:

```
@TransactionAttribute(Mandatory)
public class SuperClass {
    public void methodA () { ... }
    // other declarations
}
@Stateless
public class SubClass extends SuperClass {
    public void methodA () { ... }
    public void methodB () { ... }
    // other declarations
}
```

Which of the following is correct about transaction attributes of the methods shown above, provided no deployment descriptor exists?

Please select :

- A. Transaction attribute of methodA is Mandatory, while that of methodB is Required
- B. Transaction attribute of methodA is Required, while that of methodB is Mandatory
- C. Transaction attribute of both methods is Mandatory
- D. Transaction attribute of both methods is Required

Your answer is incorrect.

Answer: D

Explanation:

Although methodA is declared in SuperClass with transaction attribute Mandatory, it has been overridden in SubClass. No attribute is explicitly specified in the subclass, meaning that the transaction attribute of both methods is Required.

The correct answer is: Transaction attribute of both methods is Required

Submit your Feedback/Queries to our Experts

QUESTION 39

NOT ANSWERED

Ask our Experts

How can you specify that an enterprise bean has bean-managed transaction demarcation?

Please select :

- A. The bean class must be annotated @TransactionManagement(BEAN)
- B. The bean class or any method thereof must be annotated @TransactionManagement(BEAN)
- C. The bean class must be annotated @TransactionManagement(BEAN), or the transaction-type element of the bean in the deployment descriptor must be set to Bean
- D. Nothing needs to be specified, as bean-managed is the default mode of transaction demarcation

Your answer is incorrect.

Answer: C

Explanation:

The @TransactionManagement annotation can only be declared on a type, not on any method. As such, option B is incorrect.

The default transaction demarcation mode of enterprise beans is container-managed. Thus, option D is incorrect.

You can declare that an enterprise bean uses bean-managed transaction demarcation by either annotating the bean class with @TransactionManagement(BEAN) or setting the transaction-type element of the bean in the deployment descriptor to Bean.

The correct answer is: The bean class must be annotated @TransactionManagement(BEAN), or the transaction-type element of the bean in the deployment descriptor must be set to Bean

Submit your Feedback/Queries to our Experts

QUESTION 40

NOT ANSWERED

Ask our Experts

Which of the following tasks should be implemented by enterprise beans?

Please select :

- A. Transaction management
- B. Security authorization
- C. Database access
- D. None of the above

Your answer is incorrect.

Answer: C

EJB containers provide system-level services, such as transaction management or security authorization, for enterprise beans, thus beans do not need to get involved in those tasks. Instead, their jobs are to supply business logic, including access to databases.

References:

<http://docs.oracle.com/javaee/6/tutorial/doc/gipmb.html#gipk>

The correct answer is: Database access

Submit your Feedback/Queries to our Experts

QUESTION 41

NOT ANSWERED

Ask our Experts

Which of the following is NOT a programming restriction imposed on enterprise bean implementations?

Please select :

- A. Enterprise beans must not define static fields
- B. Enterprise beans, except for singleton session beans, must not use thread synchronization primitives to synchronize execution of multiple instances

- C. Enterprise beans must not use the java.io package to attempt to access files and directories in the file system
- D. None of the above

Your answer is incorrect.

Answer: A

Explanation:

The EJB 3.1 Specification (subsection 21.2.2) declares:

- An enterprise bean must not use read/write static fields. Using read-only static fields is allowed. Therefore, it is recommended that all static fields in the enterprise bean class be declared as final. This rule is required to ensure consistent runtime semantics because while some EJB containers may use a single JVM to execute all enterprise bean's instances, others may distribute the instances across multiple JVMs.
- An enterprise bean must not use thread synchronization primitives to synchronize execution of multiple instances, except if it is a singleton session bean with bean-managed concurrency. This is for the same reason as above. Synchronization would not work if the EJB container distributed enterprise bean's instances across multiple JVMs.
- An enterprise bean must not use the java.io package to attempt to access files and directories in the file system. The reason is the file system APIs are not well-suited for business components to access data. Business components should use a resource manager API, such as JDBC, to store data.

The correct answer is: Enterprise beans must not define static fields

[Submit your Feedback/Queries to our Experts](#)

QUESTION 42

NOT ANSWERED

[Ask our Experts](#)

Which of the following features is NOT supported by the EJB 3.1 Lite API?

Please select :

- A. Message-driven beans
- B. Interceptors
- C. Declarative security
- D. Bean-managed transactions
- E. None of the above

Your answer is incorrect.

Answer: A

Explanation:

As per the EJB 3.1 Specification (subsection 21.1), the EJB 3.1 Lite API is composed of the following subset of the EJB API:

- Stateless, Stateful, and Singleton session bean components (local and no-interface view only, synchronous method invocations only)
- Container-managed transactions / Bean-managed transactions
- Declarative and programmatic Security
- Interceptors
- Deployment Descriptor support (ejb-jar.xml)

The correct answer is: Message-driven beans

[Submit your Feedback/Queries to our Experts](#)

QUESTION 43

NOT ANSWERED

[Ask our Experts](#)

Which of the following methods can be used by enterprise beans?

Please select :

- A. java.io.File.getName()
- B. java.net.Socket.getLocalPort()
- C. java.lang.Thread.start()
- D. java.net.URL.setURLStreamHandlerFactory(URLStreamHandlerFactory fac)
- E. java.net.ServerSocket.setSocketFactory(SocketImplFactory fac)

Your answer is incorrect.

Answer: B

Explanation:

An enterprise bean must not use the java.io package to attempt to access files and directories in the file system. Thus, option A is incorrect.

The enterprise bean must not attempt to start, stop, suspend, or resume a thread, or to change a thread's priority or name. So, option C is incorrect.

The enterprise bean must not attempt to set the socket factory used by ServerSocket, Socket, or the stream handler factory used by URL. Hence, options D and E are incorrect.

The EJB architecture allows an enterprise bean instance to be a network socket client, there is nothing wrong with an invocation of the java.lang.Thread.start method within a bean method, therefore.

The correct answer is: `java.net.Socket.getLocalPort()`

[Submit your Feedback/Queries to our Experts](#)

QUESTION 44

NOT ANSWERED

[Ask our Experts](#)

[MARK FOR REVIEW](#)

Which THREE of the following services must be supported by an embeddable container?

Please select :

- A. EJB Timer Service
- B. Asynchronous session bean invocations
- C. Interceptors
- D. RMI-IIOP interoperability
- E. Transactions
- F. Security

Your answer is incorrect.

Answer: C, E and F

Explanation:

An embeddable container is required to support the EJB Lite API, thus it must provide all services mandated by this API. Those services include interceptors, container-managed/bean-managed transactions, and declarative and programmatic security.

The correct answers are: Interceptors, Transactions, Security

[Submit your Feedback/Queries to our Experts](#)

QUESTION 45

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

When must the EJB deployment descriptor (file ejb-jar.xml) be present?

Please select :

- A. An enterprise bean is packaged within an EJB-JAR file
- B. An enterprise bean is packaged directly within a WAR file
- C. Both of the above cases
- D. None of the above cases

Your answer is incorrect.

Answer: D

Explanation:

The EJB deployment descriptor is optional for both an EJB-JAR file and a WAR file. In the absence of the descriptor, bean settings are taken from only metadata annotations.

The correct answer is: None of the above cases

[Submit your Feedback/Queries to our Experts](#)

QUESTION 46

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Given an enterprise bean with two business interfaces:

@Stateless

```
public class MyBean implements MyLocal, MyRemote {  
    public void insertData(Data data) { ... }  
    public void getData(Integer id) { ... }  
}  
@Local  
public interface MyLocal {  
    void insertData(Data data);  
}  
@Remote  
public interface MyRemote {  
    void getData(DataId id);  
}
```

Assume you want to package the client views of MyBean into a JAR file to be used in a stand-alone client application, which is separated from the EJB container. Which type definitions must be included in such a JAR file?

Please select :

- A. MyBean only
- B. MyRemote only
- C. MyBean and MyRemote
- D. MyLocal and MyRemote
- E. MyRemote and DataId
- F. MyBean, MyLocal and MyRemote

Your answer is incorrect.

Answer: E

Explanation:

The client application is separated from the EJB application, thus it cannot access the local business interface. As such, the packaged JAR file just needs to contain type definitions for the remote business interface, including MyRemote and DataId, on which this interface depends.

DEFINITIONS FOR THE REMOTE BUSINESS INTERFACE, INCLUDING MYREMOTE AND DATALD, ON WHICH THIS INTERFACE DEPENDS.

The correct answer is: MyRemote and DataId

Submit your Feedback/Queries to our Experts

QUESTION 47

NOT ANSWERED

Ask our Experts

Which of the following information can be overridden in the EJB deployment descriptor once declared using metadata annotations?

Please select :

- A. Enterprise bean's name
- B. Method permissions
- C. Enterprise bean' business interface
- D. Session bean's state management type (stateful or stateless)

Your answer is incorrect.

Answer: B

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 method permissions are not structural information.

The correct answer is: Method permissions

Submit your Feedback/Queries to our Experts

QUESTION 48

NOT ANSWERED

Ask our Experts

Assume you have an enterprise bean, packaged along with its local client view within a JAR file, which in turn is packaged in a WAR file. Where is this view accessible?

Please select :

- A. Inside the enclosing JAR file only
- B. Inside the enclosing WAR file only
- C. Inside the application

Your answer is incorrect.

Answer: B

Explanation:

As per the EJB 3.1 Specification (subsection 20.4.3), the local client view (including the no-interface view) of an enterprise bean component defined within a WAR file is only required to be accessible to components within the same WAR file. Applications needing access to the local client view of an enterprise bean from a different module in the same application should use an EJB-JAR file to define the enterprise bean that exposes the local client view.

The correct answer is: Inside the enclosing WAR file only

Submit your Feedback/Queries to our Experts

QUESTION 49

NOT ANSWERED

Ask our Experts

Which of the following is NOT a correct way to declare a checked exception as an application exception?

Please select :

- A. Just list the exception in the throws clause of methods constituting the client views of an enterprise bean
- B. Annotate the exception with the @ApplicationException metadata annotation
- C. Denote the exception in the deployment descriptor with the application-exception element
- D. None of the above

Your answer is incorrect.

Answer: D

Explanation:

Application exceptions that are checked exceptions may be defined as such by being listed in the throws clauses of the methods of the bean's business interface, no-interface view, home interface, component interface, and web service endpoint. Hence, option A is incorrect.

The solutions declared in options B and C apply to both checked and unchecked exceptions, thus these options are incorrect as well.

The correct answer is: None of the above

Submit your Feedback/Queries to our Experts

QUESTION 50

NOT ANSWERED

Ask our Experts

Which of the following annotations allows annotated methods to throw checked exceptions?

Please select :

- A. @PostConstruct

- B. @PreDestroy
- C. @Timeout
- D. None of the above

Your answer is incorrect.

Answer: D

Explanation:

All the above methods are invoked by the container and none of them can throw a checked exception.

The correct answer is: None of the above

[Submit your Feedback/Queries to our Experts](#)

QUESTION 51

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Given an enterprise session bean with container-managed transaction demarcation. Which TWO of the following statements are correct if a bean's business method throws a system exception while running in the context of a local client's transaction?

Please select :

- A. The caller receives the original exception
- B. The caller receives an EJBException
- C. The transaction is marked for rollback
- D. The transaction may or may not be marked for rollback

Your answer is incorrect.

Answer: B and C

Explanation:

When a system exception is thrown in the given situation, the client receives `EJBTransactionRolledbackException`, a subclass of `EJBException`. Continuing the transaction is fruitless.

The correct answers are: The caller receives an `EJBException`, The transaction is marked for rollback

[Submit your Feedback/Queries to our Experts](#)

QUESTION 52

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Which of the following are application exceptions?

Please select :

- A. DuplicateKeyException
- B. EJBTransactionRequiredException
- C. NoSuchObjectException
- D. ObjectNotFoundException

Your answer is incorrect.

Answer: A and D

`EJBTransactionRequiredException` and `NoSuchObjectException` are subclasses of `EJBException` and `RemoteException`, respectively. Therefore, they are system exceptions.

`DuplicateKeyException` and `ObjectNotFoundException` are subclasses of `CreateException` and `FinderException`, respectively. As such, these exceptions are application exceptions.

Notes:

- In addition to `CreateException`, `RemoveException` and `FinderException`, there are only two more standard exceptions for entities, namely `DuplicateKeyException` (subclass of `CreateException`) and `ObjectNotFoundException` (subclass of `FinderException`).
- The `NoSuchObjectException` is explicitly mentioned when discussing the `RemoveException` in the EJB 3.1 Specification, so you may need to remember it.

The correct answers are: `DuplicateKeyException`, `ObjectNotFoundException`

[Submit your Feedback/Queries to our Experts](#)

QUESTION 53

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Which of the following declarations tells the container to guarantee that singleton bean A is initialized before singleton bean B?

Please select :

- A. `@DependsOn("B")`
`@Singleton`
`public class A { ... }`
`@Singleton`
`public class B { ... }`
- B. `@DependsOn(B.class)`
`@Singleton`
`public class A { ... }`
`@Singleton`
`public class B { ... }`
- C. `@Singleton`

```

public class A { ... }
@Singleton
@DependsOn("A")
public class B { ... }

 D. @Singleton
public class A { ... }
@Singleton
@DependsOn(A.class)
public class B { ... }

```

Your answer is incorrect.

Answer: C

Explanation:

The @DependsOn annotation is used to ensure that all singleton beans with which a singleton has a DependsOn relationship have been initialized before the singleton's PostConstruct method is called. Its value element indicates ejb-names of singleton components the annotated bean depends on.

References:

<http://docs.oracle.com/javaee/6/api/javax/ejb/DependsOn.html>

The correct answer is: @Singleton

```

public class A { ... }
@Singleton
@DependsOn("A")
public class B { ... }

```

[Submit your Feedback/Queries to our Experts](#)

QUESTION 54

NOT ANSWERED

[Ask our Experts](#)

Given a singleton session bean:

```

@Singleton
@AccessTimeout(10)
public class Whizlabs {
    @AccessTimeout(20)
    public void methodA() { ... }
    public void methodB() { ... }
    // other declarations
}

```

What is the amount of time that a concurrent access attempt should block before timing out when accessing methodA and methodB, respectively?

Please select :

- A. 30 and 10 milliseconds
- B. 20 and 10 milliseconds
- C. 20 and 10 seconds
- D. 10 seconds and 20 milliseconds for methodA, 10 seconds for methodB

Your answer is incorrect.

Answer: B

Explanation:

The @AccessTimeout can be specified on a business method or on a bean class (or superclass). An @AccessTimeout annotation specified on a class applies the access timeout to all business methods of that class. If @AccessTimeout is specified on both a class and on a business method of that class, the method-level annotation takes precedence. By default, the unit of the value specified in this annotation is millisecond. If you want to indicate the value in a different unit, use the unit element.

References:

<http://docs.oracle.com/javaee/6/api/javax/ejb/AccessTimeout.html>

The correct answer is: 20 and 10 milliseconds

[Submit your Feedback/Queries to our Experts](#)

QUESTION 55

NOT ANSWERED

[Ask our Experts](#)

Which TWO of the following client views may not be supported by singleton session beans?

Please select :

- A. Component interfaces
- B. Home interfaces
- C. Business interfaces
- D. No-interface view
- E. Web service endpoint interface

Your answer is incorrect.

Answer: A and B

Explanation:

Component interfaces and home interfaces are used for beans written to the EJB 2.1 and earlier APIs. Enterprise beans used with EJB 3.0 or later, including singleton, are

not required to support those client views.

The correct answers are: Component interfaces, Home interfaces

[Submit your Feedback/Queries to our Experts](#)

QUESTION 56

NOT ANSWERED

[Ask our Experts](#)

Given two class declarations:

@Lock(READ)

public class SomeClass {

 public void methodA () { ... }

 public void methodB () { ... }

 // other declarations

}

@Singleton

public class Whizlabs extends SomeClass {

 public void methodA () { ... }

 @Lock(WRITE)

 public void methodC () { ... }

 // other declarations

}

What are concurrency locking attributes of methodA, methodB and methodC of bean Whizlabs, respectively, provided there is no associated deployment descriptor?

Please select :

- A. READ, READ, WRITE
- B. WRITE, READ, WRITE
- C. READ, WRITE, READ
- D. READ, WRITE, WRITE

Your answer is incorrect.

Answer: B

Explanation:

By default, the concurrency attribute of a method is WRITE. Therefore, even though the attribute of methodA is declared as READ in SomeClass, its effective value is WRITE, which is the result of the overriding in class Whizlabs.

The concurrency locking attributes of methodB and methodC are READ and WRITE, respectively, as explicitly specified in the class level and method level.

The correct answer is: WRITE, READ, WRITE

[Submit your Feedback/Queries to our Experts](#)

QUESTION 57

NOT ANSWERED

[Ask our Experts](#)

Given a stateless session bean having a business method that throws a system exception. Which of the following actions the bean and its associated interceptors must perform prior to the exception being thrown to ensure data integrity?

Please select :

- A. Mark the transaction in which the bean method participates for rollback
- B. Perform necessary clean-up operations (in lifecycle callback interceptor methods)
- C. Both the above
- D. None of the above

Your answer is incorrect.

Answer: B

Explanation:

The operation described in options A should be performed to ensure data integrity when an application exception is thrown. However, in case of a system exception, the container itself will mark the transaction for rollback. As to the clean-up job, @PreDestroy methods are not invoked as a result of such an exception. The lifecycle callback interceptor methods in the chain should perform any necessary clean-up operations as the interceptor chain unwinds.

The correct answer is: Perform necessary clean-up operations (in lifecycle callback interceptor methods)

[Submit your Feedback/Queries to our Experts](#)

QUESTION 58

NOT ANSWERED

[Ask our Experts](#)

Which of the following statements is correct about timer services?

Please select :

- A. When a timer is created in a stateless session bean instance, its associated timeout callback method is invoked only on this instance
- B. When a timer is cancelled, its associated timeout callback method is guaranteed not to be invoked
- C. Timers may survive server shutdown or container crash
- D. EJB timer service is suitable for the modeling of real-time events

Your answer is incorrect.

Answer: C

Explanation:

The timeout callback method invocation for a timer that is created for a stateless session bean or a message-driven bean may be called on any bean instance in the pooled state.

If a timer is cancelled, its associated timeout callback method is typically not called. However, in the event of race conditions, extraneous calls to the timeout callback method may still occur.

The EJB timer service is a coarse-grained timer notification service that is designed for use in the modeling of application-level processes. It is not intended for the modeling of real-time events. Timeout callback methods are not guaranteed to be executed at the exact time specified at timer creation.

Persistent timers survive container crash, server shutdown, and the activation/passivation and load/store cycles of the enterprise beans that are registered with them. Therefore, option C is the correct answer.

The correct answer is: Timers may survive server shutdown or container crash

[Submit your Feedback/Queries to our Experts](#)

QUESTION 59

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Given a stateless session bean that does not use timer service correctly:

@Stateless

```
public class Whizlabs implements TimedObject {
    @Resource
    private TimerService timerService;
    public void createTimer() {
        ScheduleExpression schedule = new ScheduleExpression().hour(12);
        Timer timer = timerService.createCalendarTimer(schedule);
    }
    public void ejbTimeout(Timer timer) {
        timeoutCallback();
    }
    @Timeout
    public void timeoutCallback() {
        System.out.println("Timeout");
    }
}
```

Which of the following changes is NOT a correct way to print a message on the console at midday every day?

Note that all modifications described in options A, B and C are independent of each other, and no timeout-related elements are existent in the deployment descriptor.

Please select :

- A. Remove the @Timeout annotation on the timeoutCallback method
- B. Remove the TimedObject interface from the implements clause on the bean class declaration
- C. Move the @Timeout annotation from the timeoutCallback method to ejbTimeout
- D. None of the above

Your answer is incorrect.**Answer: D**

Explanation:

All timers created via one of the TimerService timer creation methods for a particular component use a single timeout callback method. This method may be a method annotated with the @Timeout annotation (or a method specified as a timeout method in the deployment descriptor) or the bean may implement the TimedObject interface.

The solution in option A makes the Whizlabs bean implements the TimedObject interface appropriately, while the solution in option B registers timeoutCallback as the only timeout callback method.

When a bean implements the TimedObject interface, the @Timeout annotation is unnecessary. However, there is nothing wrong if you specify this annotation on the ejbTimeout method defined by that interface. Hence, the solution in option C is also valid. Note that the @Timeout annotation is not allowed to be declared on any method other than ejbTimeout.

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 correct about the relationship between timers and transaction?

Please select :

- A. If an enterprise bean creates or cancels a timer within a transaction, and the transaction is rolled back, the creation or cancellation will still be in effect
- B. If an enterprise bean creates or cancels a timer within a transaction, and the transaction is rolled back, the creation or cancellation will be revoked
- C. If a timeout callback method of a bean runs within a transaction, and the transaction is rolled back, the container will always retry the timeout
- D. None of the above

Your answer is incorrect.**Answer: B**

Explanation:

As per the EJB 3.1 Specification (subsection 18.2.8), an enterprise bean typically creates a timer within the scope of a transaction. If the transaction is then rolled back, the timer creation is rolled back. An enterprise bean typically cancels a timer within a transaction. If the transaction is rolled back, the container rescinds the timer cancellation.

The container just retries the timeout if the transaction is started by the container itself, right before the execution of the timeout callback. This occurs in case the bean has container-managed transaction demarcation, and the transaction attribute of the callback is Required or RequiresNew.

The correct answer is: If an enterprise bean creates or cancels a timer within a transaction, and the transaction is rolled back, the creation or cancelation will be revoked

[Submit your Feedback/Queries to our Experts](#)

[Finish review](#)

Company

[About Us](#)

Contact us

[Live Chat](#)

support@whizlabs.com

Communities

[Discussions](#)

[Blog](#)

Follow Us



Copyright © 2018 Whizlabs Software Pvt Ltd. All rights reserved.