



## OCEEJBD 6 Practice Test

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



0.



**Started on** Monday, 22 October 2018, 9:17 PM

**State** Finished

**Completed on** Monday, 22 October 2018, 9:18 PM

**Time taken** 50 secs

**Grade** 0.00 out of 60.00 (0%)

### Result

QUESTION 1	NOT ANSWERED	MARK FOR REVIEW
<p>Which of the following statements is correct about client access to a session bean in another application collocated in the same container as the client?</p> <p>Please select :</p> <ul style="list-style-type: none"><li><input type="radio"/> A. The access can only be done through a web service client view</li><li><input type="radio"/> B. The access can only be done through a remote or web service client view</li><li><input type="radio"/> C. The access can be done through a local client view in all compliant containers</li><li><input type="radio"/> D. The access may or may not be done through a local client view, depending on the container in use</li></ul> <p><b>Your answer is incorrect.</b></p> <p><b>Answer: D</b></p> <p><b>Explanation:</b></p> <p>As per the EJB 3.1 Specification (subsection 3.2.2), access to an enterprise bean through the local client view is only required to be supported for local clients packaged within the same application as the enterprise bean that provides the local client view. Compliant implementations of this specification may optionally support access to the local client view of an enterprise bean from a local client packaged in a different application. The configuration requirements for inter-application access to the local client view are vendor-specific and are outside the scope of this specification. Applications relying on inter-application access to the local client view are non-portable.</p> <p>The correct answer is: The access may or may not be done through a local client view, depending on the container in use</p> <p><a href="#">Submit your Feedback/Queries to our Experts</a></p>		

QUESTION 2	NOT ANSWERED	MARK FOR REVIEW
<p>Given a bean class declaration:</p> <pre>package com.whizlabs; @Stateful(name = "Whizlabs") @LocalBean @Remote(WhizlabsInterface.class) public class WhizlabsBean implements WhizlabsInterface {     // bean class body }</pre> <p>Which of the following is the name entry representing the no-interface view of the above bean in the naming context of the containing module, provided no deployment descriptor is used?</p> <p>Please select :</p> <ul style="list-style-type: none"><li><input type="radio"/> A. java:module/Whizlabs</li><li><input type="radio"/> B. java:module/WhizlabsBean</li><li><input type="radio"/> C. java:module/Whizlabs!com.whizlabs.WhizlabsBean</li><li><input type="radio"/> D. java:module/com.whizlabs.WhizlabsBean!Whizlabs</li><li><input type="radio"/> E. 2 and either 0 or 1</li></ul> <p><b>Your answer is incorrect.</b></p> <p><b>Answer: C</b></p> <p><b>Explanation:</b></p> <p>As per the EJB 3.1 Specification (subsection 4.4.1), the syntax for a session bean JNDI name is as follows:</p> <pre>java:module&lt;bean-name&gt;[!&lt;fully-qualified-interface-name&gt;]</pre> <p>Note that the part [&lt;fully-qualified-interface-name&gt;] can only be left out if the bean has only one client interface or only a no-interface view. In the given scenario, the bean class has a remote business interface and a no-interface view, thus the fully qualified name of the interface must be specified.</p> <p>The correct answer is: java:module/Whizlabs!com.whizlabs.WhizlabsBean</p> <p><a href="#">Submit your Feedback/Queries to our Experts</a></p>		

Given several class declarations:

```

public class Data implements Serializable {
    private String value;
    // getter and setter
}
@Stateless
@Remote(MyInterface.class)
public class WhizlabsBean implements Whizlabs {
    public void setUpperCase(Data data) {
        data.setValue(data.getValue().toUpperCase());
    }
}
@Stateless
public class MyBean {
    @EJB
    private Whizlabs whizlabs;
    public void processData() {
        Data data = new Data();
        data.setValue("whizlabs");
        whizlabs.setUpperCase(data);
        System.out.println(data.getValue());
    }
}

```

It is known that `setUpperCase(Data)` is a method defined by the `Whizlabs` interface. What happens when the `MyBean.processData` is executed?

Please select :

- A. The string "whizlabs" is printed on the console
- B. The string "WHIZLABS" is printed on the console
- C. An exception is thrown at runtime

**Your answer is incorrect.**

**Answer: A**

Explanation:

Arguments and results of the business methods of a remote business interface are passed by value, thus after a `Data` instance is passed from `MyBean` to `WhizlabsBean` using serialization/deserialization, `MyBean` and `WhizlabsBean` work on different instances of the `Data` class. Any changes made to the instance in `MyBean` is invisible to `WhizlabsBean`, and vice versa.

The correct answer is: The string "whizlabs" is printed on the console

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

Given a session bean declaration:

```

@Stateful
@Local(Whizlabs.class)
public class WhizlabsBean implements Whizlabs {
    // class body
}

```

A client obtains references to the `Whizlabs` business interface by the following dependency injections:

```

@EJB WhizlabsBean whizlabs1;
@EJB WhizlabsBean whizlabs2;
@EJB WhizlabsBean whizlabs3;

```

After the injections, the following statements are executed:

```

boolean check1 = whizlabs1.equals(whizlabs1);
boolean check2 = whizlabs1.equals(whizlabs2);
boolean check3 = whizlabs2.equals(whizlabs3);

```

What are the values of variables `check1`, `check2` and `check3`, respectively?

Please select :

- A. true, false and false
- B. false, false and false
- C. true, true and true
- D. true, true and false

**Your answer is incorrect.**

**Answer: A**

Explanation:

As per the EJB 3.1 Specification (subsection 3.4.7.1), a stateful session object has a unique identity that is assigned by the container at the time the object is created. Therefore, the value of `check1` is true while that of `check2` is false.

The same section of the Specification also announces that stateful session bean references to different interface types or between an interface type and a no-interface view or to different session bean instances will not have the same identity. This means the value of `check3` is false.

The correct answer is: true, false and false

Submit your Feedback/Queries to our Experts

QUESTION 5

NOT ANSWERED

Ask our Experts

MARK FOR REVIEW

Which of the following is a valid callback method for stateless session beans?

Please select :

- A. @PostActivate  
public void postActivate() { ... }
- B. @PreDestroy  
protected void preDestroy(String string) { ... }
- C. @PostConstruct  
public boolean postConstruct() { ... }
- D. @PreDestroy  
final void preDestroy() { ... }
- E. @PostConstruct  
static void postConstruct() { ... }
- F. None of the above

**Your answer is incorrect.**

**Answer: D**

**Explanation:**

The @PostActivate and @PrePassivate annotations apply to stateful session beans only. Thus, option A is incorrect.

A callback method must not have any parameters and its return type must be void. Hence, options B and C are incorrect.

A callback method must not be static, then option E is incorrect.

@PostConstruct and @PreDestroy callback methods can be declared as final. Therefore, option D is the correct answer.

The correct answer is: @PreDestroy

final void preDestroy() { ... }

Submit your Feedback/Queries to our Experts

QUESTION 6

NOT ANSWERED

Ask our Experts

MARK FOR REVIEW

Which of the following statements is correct about instances of a stateless session bean class in an EJB container?

Please select :

- A. A bean instance cannot contain any state across client-invoked method calls
- B. Requests from the same client within the same transaction must be delegated to the same instance
- C. After servicing a client, the bean instance must be destroyed
- D. The container creates a new instance only when there is a client call to one of its methods
- E. None of the above

**Your answer is incorrect.**

**Answer: E**

**Explanation:**

A stateless session bean instance has no state for a specific client. However, fields of the instance can contain the state across client-invoked method calls. Examples of such state include an open database connection and an object reference to an enterprise bean object. Option A is incorrect, then.

Since all instances of a stateless session bean are equivalent, the container can choose to delegate a client-invoked method to any available instance. This means that the container may delegate the requests from the same client within the same transaction to different instances. Hence, option B is incorrect.

After servicing a client, the container often puts the bean instance back to the method-ready pool. The instance is destroyed only if it is not needed to handle the current client work load. Thus, option C is incorrect.

The container can perform the instance creation at any time - there is no direct relationship to a client's invocation of a business method. As such, option D is incorrect.

The correct answer is: None of the above

Submit your Feedback/Queries to our Experts

QUESTION 7

NOT ANSWERED

Ask our Experts

MARK FOR REVIEW

Which of the following is NOT a requirement for a session synchronization method specified by means of metadata annotations (a method annotated with @AfterBegin, @AfterCompletion or @BeforeCompletion)?

Please select :

- A. The method must not be declared as final
- B. The method must not be declared as static
- C. The method must have access modifier public
- D. The return type must be void
- E. None of the above

**Your answer is incorrect.**

**Answer: C**

**Explanation:**

The annotated method may have any access type: public, private, protected, or package-level. All requirements described in options A, B and D must be met.

The correct answer is: The method must have access modifier public

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

QUESTION 8

NOT ANSWERED

[Ask our Experts](#)

Given a stateful session bean:

@Stateful

```
public class Whizlabs implements SessionBean {  
    public void ejbActivate() { ... }  
    public void ejbPassivate() { ... }  
    public void ejbRemove() { ... }  
    public void setSessionContext(SessionContext ctx) { ... }  
    // other declarations  
}
```

Which method is called when a Whizlabs bean instance is activated from its "passive" state, provided there is no associated deployment descriptor?

Please select :

- A. ejbActivate
- B. ejbPassivate
- C. ejbRemove
- D. setSessionContext
- E. None of the above

**Your answer is incorrect.**

**Answer: A**

**Explanation:**

The ejbPassivate method is called when a stateful session bean instance enters the "passive" state, not when it is activated. Hence, option B is incorrect.

The ejbRemove method is invoked by the container before the life of the session object comes to an end. This happens as a result of an invocation of the remove operation, or when the container decides to terminate the session object after a timeout. It has nothing to do with passivation, meaning that option C is incorrect.

The setSessionContext method sets the associated session context, which is called only after the instance creation. Therefore, option D is incorrect.

The ejbActivate method is called when a stateful session bean instance is activated from its "passive" state. As such, option A is the correct answer.

The correct answer is: ejbActivate

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

QUESTION 9

NOT ANSWERED

[Ask our Experts](#)

Given an asynchronous method of an enterprise bean:

@Asynchronous

```
public Future<String> asynchronousMethod() throws Exception {  
    // throw MyException  
    return new AsyncResult<String>("Whizlabs");  
}
```

Assume the above business method throws MyException, an application exception. Which exception is thrown when the following code fragment is executed by a client, with myBean referencing a valid client view of the containing bean of the asynchronousMethod method?

```
Future<Integer> result = myBean.asynchronousMethod();  
System.out.println(result.get());
```

Please select :

- A. MyException
- B. EJBException
- C. ExecutionException
- D. IllegalStateException

**Your answer is incorrect.**

**Answer: C**

**Explanation:**

Unlike the invocation of a synchronous business method, an exception thrown from the bean method is always wrapped in the Future return value (if the asynchronous method has a return type of Future). As such, the myBean.asynchronousMethod invocation does not throw any exception to the client.

When the get method is invoked on the result, an ExecutionException is thrown to indicate that an exception occurs within the bean method. The original exception can be inspected by invoking the getCause method on that ExecutionException instance.

The correct answer is: ExecutionException

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

QUESTION 10

[Ask our Experts](#)

NOT ANSWERED

MARK FOR REVIEW

Given an enterprise bean:

```
@Stateless  
public class WhizlabsBean implements Whizlabs {  
    @Resource  
    SessionContext context;  
    // other declarations  
}
```

Which of the following expressions can be used within WhizlabsBean to obtain an object that can be used to invoke the bean through the Whizlabs business interface view?

Please select :

- A. context.getBusinessObject(Whizlabs.class)
- B. context.getLocalInterface(Whizlabs.class)
- C. context.getCurrentBean()
- D. context.getBusinessInterface(Whizlabs.class)

**Your answer is incorrect.**

**Answer: A**

Explanation:

As per the Java EE 6 API documentation, the getBusinessObject method of the SessionContext interface obtain an object that can be used to invoke the current bean through a particular business interface view or its no-interface view. This interface does not define any methods shown in other options.

References:

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

The correct answer is: context.getBusinessObject(Whizlabs.class)

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

QUESTION 11

NOT ANSWERED

Ask our Experts

Given two session beans:

```
@Stateless  
public class Whizlabs {  
    @Asynchronous  
    public void asyncMethod() { ... }  
    // other declarations  
}  
  
@Stateless  
public class MyBean {  
    @EJB  
    Whizlabs whizlabs;  
    public void myMethod() {  
        whizlabs.asyncMethod();  
        // do something  
    }  
    // other declarations  
}
```

Which of the following statements is correct if the MyBean.myMethod method is invoked, provided there is no associated deployment descriptor?

Please select :

- A. MyBean.myMethod and Whizlabs.asyncMethod runs within the same transaction
- B. MyBean.myMethod and Whizlabs.asyncMethod runs within separate transactions
- C. MyBean.myMethod runs within a transaction, while Whizlabs.asyncMethod does not
- D. MyBean.myMethod does not run within a transaction, while Whizlabs.asyncMethod does
- E. Neither MyBean.myMethod nor Whizlabs.asyncMethod runs within a transaction

**Your answer is incorrect.**

**Answer: B**

Explanation:

There are no transaction-related annotations declared on the given beans, implying that both beans use container-managed transaction demarcation and the transaction attributes of two methods in question are Required. As such, both MyBean.myMethod and Whizlabs.asyncMethod run within transactions.

According to the EJB 3.1 Specification (subsection 4.5.3), client transaction context does not propagate with an asynchronous method invocation. As a result, the container starts a new transaction before dispatching the asynchronous business method.

The correct answer is: MyBean.myMethod and Whizlabs.asyncMethod runs within separate transactions

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

QUESTION 12

NOT ANSWERED

Ask our Experts

Which of the following is a valid declaration of an EJB reference in the deployment descriptor?

Please select :

- A. <ejb-local-ref>  
<ejb-ref-name>ejb/Whizlabs</ejb-ref-name>  
<ejb-ref-type>Session</ejb-ref-type>  
<local>com.whizlabs.Whizlabs</local>  
</ejb-local-ref>
- B. <ejb-ref>  
<ejb-ref-name>ejb/Whizlabs</ejb-ref-name>  
<ejb-ref-type>com.whizlabs.Whizlabs</ejb-ref-type>  
<remote>com.whizlabs.Whizlabs</remote>  
</ejb-ref>
- C. <ejb-ref>  
<ejb-ref-name>ejb/Whizlabs</ejb-ref-name>  
<ejb-ref-type>Stateless</ejb-ref-type>  
<local>com.whizlabs.Whizlabs</local>  
</ejb-ref>
- D. <ejb-ref>  
<ejb-ref-name>ejb/Whizlabs</ejb-ref-name>  
<ejb-ref-type>Session</ejb-ref-type>  
<local>com.whizlabs.Whizlabs</local>  
</ejb-ref>
- E. <ejb-local-ref>  
<ejb-ref-name>ejb/Whizlabs</ejb-ref-name>  
<ejb-ref-type>Stateful</ejb-ref-type>  
<remote>com.whizlabs.Whizlabs</remote>  
</ejb-local-ref>

**Your answer is incorrect.**

**Answer: A**

Explanation:

The ejb-ref element does not contain the local element, nor does ejb-local-ref contain remote. Thus, options C, D and E are incorrect in the first place.

The ejb-ref-type element specifies the expected type of the enterprise bean: its value must be either Entity or Session. As such, option B is incorrect as well.

The correct answer is: <ejb-local-ref>

```
<ejb-ref-name>ejb/Whizlabs</ejb-ref-name>
<ejb-ref-type>Session</ejb-ref-type>
<local>com.whizlabs.Whizlabs</local>
</ejb-local-ref>
```

Submit your Feedback/Queries to our Experts

QUESTION 13

NOT ANSWERED

Ask our Experts

Which of the following objects must be created by a connection rather than a session in a JMS application?

Please select :

- A. Message producers and consumers
- B. Messages
- C. Queue browsers
- D. Temporary destinations
- E. None of the above

**Your answer is incorrect.**

**Answer: E**

Explanation:

As per the Oracle's Java EE 6 Tutorial, a session is a single-threaded context for producing and consuming messages. You use sessions to create the following:

- Message producers
- Message consumers
- Messages
- Queue browsers
- Temporary queues and topics

References:

<http://docs.oracle.com/javaee/6/tutorial/doc/bnch.html#bnccen>

The correct answer is: None of the above

Submit your Feedback/Queries to our Experts

QUESTION 14

NOT ANSWERED

Ask our Experts

Given a JMS Connection object referenced by variable connection. What happens when this variable is used in a statement as follows:

Session session = connection.createSession(true, 4);

Please select :

- A. A IMSException is thrown as an acknowledgement mode value of 4 is not supported.

- B. The created session is not transacted
- C. The created session is transacted
- D. None of the above

**Your answer is incorrect.**

Answer: C

Explanation:

When creating a session using the Connection.createSession method with the first argument being true, the created session is transacted and the second argument to the invocation is ignored. Thus, options A and B are not correct, while option C is.

References:

[http://docs.oracle.com/javaee/6/api/javax/jms/Connection.html#createSession\(boolean, int\)](http://docs.oracle.com/javaee/6/api/javax/jms/Connection.html#createSession(boolean, int))

The correct answer is: The created session is transacted

Submit your Feedback/Queries to our Experts

QUESTION 15

NOT ANSWERED

Ask our Experts

What happens when the following code fragment is executed, given variable session references a JMS transacted Session instance?

```
MessageProducer producer = session.createProducer(outQueue);
TextMessage outMessage = session.createTextMessage();
outMessage.setText("Whizlabs");
outMessage.setJMSReplyTo(inQueue);
producer.send(outMessage);
consumer = session.createConsumer(inQueue);
TextMessage inMessage = (TextMessage) consumer.receive();
String message = inMessage.getText();
System.out.println(message);
session.commit();
```

Please select :

- A. The program hangs
- B. The program completes successfully but nothing gets printed
- C. The String "Whizlabs" is printed on the console
- D. A JMSEException is thrown

**Your answer is incorrect.**

Answer: A

Explanation:

A message sent during a transaction is not actually sent until the transaction is committed. The execution of method consumer.receive, therefore, blocks indefinitely, waiting for a message that will never come.

References:

<http://docs.oracle.com/javaee/6/tutorial/doc/bncfu.html#bncgh>

The correct answer is: The program hangs

Submit your Feedback/Queries to our Experts

QUESTION 16

NOT ANSWERED

Ask our Experts

Which of the following interfaces gives a message-driven bean instance access to the instance's context maintained by the container?

Please select :

- A. Context
- B. MessageDriven
- C. MessageDrivenBean
- D. MessageDrivenContext
- E. None of the above

**Your answer is incorrect.**

Answer: D

Explanation:

As per the Java EE 6 API documentation, the MessageDrivenContext interface provides access to the runtime message-driven context that the container provides for a message-driven bean instance. The container passes the MessageDrivenContext interface to an instance after the instance has been created. The message-driven context remains associated with the instance for the lifetime of the instance.

All methods of the MessageDrivenContext interface are inherited from interface EJBContext.

References:

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

The correct answer is: MessageDrivenContext

Submit your Feedback/Queries to our Experts

<b>QUESTION 17</b>	<b>NOT ANSWERED</b>	<b>Ask our Experts</b>
		▼ MARK FOR REVIEW

Which of the following is NOT a requirement for a message-driven bean class?

Please select :

- A. It must implement the MessageListener interface
- B. It must be a top-level public class
- C. It must not be final or abstract
- D. It must have a public no-argument constructor
- E. It must not define the finalize method

**Your answer is incorrect.**

**Answer: A**

**Explanation:**

A message-driven bean class just implements the MessageListener interface in case it is a JMS consumer. This requirement does not apply to all message-driven beans. Therefore, option A is the correct answer.

All other options are requirements for a message-driven bean class.

Note that a message-driven bean class must have a no-argument constructor does not mean that you must always define such a constructor. If you do not define any constructor, the default constructor is automatically public since the class access modifier is public.

The correct answer is: It must implement the MessageListener interface

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

<b>QUESTION 18</b>	<b>NOT ANSWERED</b>	<b>Ask our Experts</b>
		▼ MARK FOR REVIEW

Which of the following modifiers is allowable on the message listener method of a message-driven bean?

Please select :

- A. private
- B. static
- C. final
- D. None of the above

**Your answer is incorrect.**

**Answer: D**

**Explanation:**

As per the EJB 3.1 Specification (subsection 5.6.4), the message-driven bean class must define the message listener methods. The signature of a message listener method must follow these rules:

- The method must be declared as public
- The method must not be declared as final or static

The correct answer is: None of the above

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

<b>QUESTION 19</b>	<b>NOT ANSWERED</b>	<b>Ask our Experts</b>
		▼ MARK FOR REVIEW

Which of the following annotations associates the annotated message-driven bean with a JMS Topic using a non-durable subscription when the bean is deployed in the container?

Please select :

- A. @MessageDriven(activationConfig = {@ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Topic")})
- B. @MessageDriven(activationConfig = {@ActivationConfigProperty(propertyName = "destination", propertyValue = "javax.jms.Topic"), @ActivationConfigProperty(propertyName = "subscriptionDurability", propertyValue = "NonDurable")})
- C. @MessageDriven(activationConfig = {@ActivationConfigProperty(destination = "Topic", subscriptionDurability = "NonDurable")})
- D. @MessageDriven(activationConfig = {@ActivationConfigProperty(subscriptionDurability = "NonDurable")})

**Your answer is incorrect.**

**Answer: A**

**Explanation:**

In order to specify activation config properties, you must use the activationConfig element, whose value is an array of @ActivationConfigProperty. This annotation defines only two elements: propertyName and propertyValue. Thus, options C and D are incorrect in the first place.

Option B is also incorrect since the value of the first propertyName element must be destinationType rather than destination.

The default value of property subscriptionDurability is NonDurable, so it does not need to be explicitly set, and option A is the correct answer.

The correct answer is: @MessageDriven(activationConfig = {@ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Topic")})

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

Given an enterprise bean and its super class:

```
public class Whizlabs {
    @AroundTimeout
    public Object intercept(InvocationContext ctx) throws Exception {
        System.out.print("Whizlabs ");
    }
}
@Stateful
public class MyBean extends Whizlabs implements A {
    @Timeout
    public void timesUp() { ... }
    @AroundTimeout
    public Object intercept(InvocationContext ctx) throws Exception {
        System.out.print("MyBean ");
    }
    // other declarations
}
```

And an interceptor class:

```
@Interceptor
public class MyInterceptor {
    @AroundTimeout
    public Object intercept(InvocationContext ctx) throws Exception {
        System.out.print("MyInterceptor ");
    }
}
```

Here is part of the EJB deployment descriptor:

```
<assembly-descriptor>
    <interceptor-binding>
        <target-name>MyBean</target-name>
        <interceptor-class>MyInterceptor</interceptor-class>
    </interceptor-binding>
</assembly-descriptor>
```

When the timeout callback method timesUp is triggered, which of the following is printed on the console?

Please select :

- A. MyBean MyInterceptor
- B. Whizlabs MyBean MyInterceptor
- C. MyBean Whizlabs MyInterceptor
- D. MyInterceptor MyBean
- E. MyInterceptor Whizlabs MyBean
- F. MyInterceptor MyBean Whizlabs

**Your answer is incorrect.**

**Answer: A**

Explanation:

If an around-timeout method is overridden by another method (regardless of whether that method is itself an around-timeout method), it will not be invoked. As such, the string Whizlabs is never printed.

When the deployment descriptor is used to augment the interceptors specified in annotations, the interceptor methods specified in the deployment descriptor will be invoked after those specified in annotations. Thus, MyInterceptor is printed after MyBean.

The correct answer is: MyBean MyInterceptor

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

Given an interceptor class:

```
@Interceptor
public class MyInterceptor {
    private int invoke = 0;
    private int passivate = 0;
    // getters
    @PrePassivate
    public void passivateBean() {
        passivate++;
    }
    @AroundInvoke
    Object intercept(InvocationContext context) throws Exception {
        invoke++;
        return context.proceed();
    }
}
```

The above interceptor class is specified within a bean:

```
@Stateful  
public class Whizlabs implements A {  
    @Interceptors(MyInterceptor.class)  
    public int triple(int arg) {  
        return 3 * arg;  
    }  
    // other declarations  
}
```

Assume there is a client that calls the Whizlabs.triple method twice in the same transaction, and there are no interceptor-related elements in the deployment descriptor. What are the values of the invoke and passivate fields of MyInterceptor after the second call, respectively?

Please select :

- A. 0 and 0
- B. 0 and 1
- C. 2 and 0
- D. 2 and 1
- E. 2 and either 0 or 1

**Your answer is incorrect.**

**Answer: C**

Explanation:

The lifecycle of an interceptor instance is the same as that of the target class instance with which it is associated. Therefore, a MyInterceptor instance is maintained across invocations from the same client. The invoke variable, therefore, is increased twice and reaches the value of 2.

The passivateBean method is invoked when a bean instance is passivated. In the given scenario, both invocations occur in the same transaction, meaning that the bean is not passivated. Thus, the passivate field is not increased.

The correct answer is: 2 and 0

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 {  
        return null;  
    }  
}
```

The above interceptor class is specified within a bean:

```
@Stateful  
public class Whizlabs {  
    @Interceptors(MyInterceptor.class)  
    public String upperCase(String text) {  
        return text.toUpperCase();  
    }  
    // other declarations  
}
```

What is returned when a local client invokes the Whizlabs.upperCase method with argument "whizlabs"?

Please select :

- A. null
- B. "whizlabs"
- C. "WHIZLABS"
- D. An exception is thrown

**Your answer is incorrect.**

**Answer: A**

Since the MyInterceptor.intercept interposes on the invocation without proceeding to the target method, this method is not invoked. The result of the invocation is returned directly from the MyInterceptor.intercept method, which is null.

The correct answer is: null

Submit your Feedback/Queries to our Experts

QUESTION 23

NOT ANSWERED

▼ MARK FOR REVIEW

[Ask our Experts](#)

Which of the following statements is NOT correct about around-invoke interceptors?

Please select :

- A. Only one around-invoke method may be present on a given class
- B. Around-invoke methods can only have non-private level access

- C. An around-invoke method must not be declared as final or static
- D. An around-invoke method can invoke any component or resource that the method on which it interposes can invoke
- E. Around-invoke interceptors can throw any exception allowed by the throws clause of the target method
- F. Around-invoke method invocations occur within the same transaction and security context as the method on which they are interposing

**Your answer is incorrect.**

**Answer: B**

Explanation:

As per the Oracle's Java EE 6 Tutorial, only one around-invoke interceptor method per class is allowed. Around-invoke interceptor methods have the following form:

@AroundInvoke

```
visibility Object method-name(InvocationContext) throws Exception { ... }
```

Around-invoke interceptor methods can have public, private, protected, or package-level access, and must not be declared static or final.

An around-invoke interceptor can call any component or resource callable by the target method on which it interposes, have the same security and transaction context as the target method, and run in the same Java virtual machine call-stack as the target method.

Around-invoke interceptors can throw any exception allowed by the throws clause of the target method. They may catch and suppress exceptions, and then recover by calling the InvocationContext.proceed method.

References:

<http://docs.oracle.com/javaee/6/tutorial/doc/gkedm.html#gkey>

The correct answer is: Around-invoke methods can only have non-private level access

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

QUESTION 24

NOT ANSWERED

Ask our Experts

Given an enterprise bean declaration:

```
@RunAs("whizlabs")
public class Whizlabs implements MyInterface {
    // class body
}
```

Which of the following EJB deployment descriptor elements can be used to specify that the caller principal of methods within the Whizlabs bean is propagated from their calling methods?

Please select :

- A. <security-identity>
 <run-as>
 <role-name></role-name>
 </run-as>
 </security-identity>
- B. <security-identity>
 <run-as>
 <role-name>\*</role-name>
 </run-as>
 </security-identity>
- C. <security-identity>
 <run-as>
 <role-name>caller</role-name>
 </run-as>
 </security-identity>
- D. <security-identity>
 <run-as>
 <role-name>use-caller-identity</role-name>
 </run-as>
 </security-identity>
- E. It is impossible to override the value specified in the @RunAs annotation

**Your answer is incorrect.**

**Answer: D**

Explanation:

The following is an extract taken from the EJB 3.1 Specification (subsection 17.1):

If the deployment descriptor is used to specify the security principal, the bean provider or the application assembler can use the security-identity deployment descriptor element to specify the security identity. If the security-identity deployment descriptor element is not specified and if a run-as identity has not been specified by the use of the @RunAs annotation or if use-caller-identity is specified as the value of the security-identity element, the caller principal is propagated from the caller to the callee. If the run-as element is specified, a security principal that has been assigned to the specified security role will be used. The application assembler is permitted to override a security identity value set or defaulted by the bean provider.

The correct answer is: <security-identity>

```
<run-as>
<role-name>use-caller-identity</role-name>
</run-as>
</security-identity>
```

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

<b>QUESTION 25</b>	<b>NOT ANSWERED</b>	<b>Ask our Experts</b>
--------------------	---------------------	------------------------

Which of the following statements is correct about specifying the method-permission element in the EJB deployment descriptor?

Please select :

- A. There can only be one method-permission element in the deployment descriptor
- B. There can be more than one method-permission element in the deployment descriptor; if that is the case, all elements except for the first one are ignored
- C. The method permissions relation is defined as the union of all the method permissions defined in individual method-permission elements; a security role can appear in only one element
- D. None of the above

**Your answer is incorrect.**

**Answer: D**

**Explanation:**

As per the EJB 3.1 Specification (subsection 17.3.2.2), the method permissions relation is defined as the union of all the method permissions defined in the individual method-permission elements. A security role or a method may appear in multiple method-permission elements.

The correct answer is: None of the above

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

<b>QUESTION 26</b>	<b>NOT ANSWERED</b>	<b>Ask our Experts</b>
--------------------	---------------------	------------------------

Which of the following is a responsibility of the deployer with respect to security?

Please select :

- A. Configure the principal delegation for inter-component calls
- B. Make sure that the security view defined in the deployment descriptor by the bean provider and application assembler is mapped to the operational environment
- C. Define logical security roles
- D. Specify permissions on enterprise bean methods

**Your answer is incorrect.**

**Answer: A**

**Explanation:**

As per the EJB 3.1 Specification (subsection 17.4.3), the deployer is responsible for configuring the principal delegation for inter-component calls. The deployer must follow any instructions supplied by the bean provider and/or application assembler (for example, provided in the RunAs metadata annotations, the run-as elements of the deployment descriptor, in the description elements of the annotations or deployment descriptor, or in a deployment manual). Therefore, option A is the correct answer.

The EJB Specification (subsection 17.4.5) also declares that the deployer can use the security view defined in the deployment descriptor by the bean provider and application assembler merely as "hints" and may change the information whenever necessary to adapt the security policy to the operational environment. So, option B is incorrect.

Defining logic security roles and specifying permissions on enterprise bean methods are tasks of the bean provider and application assembler.

The correct answer is: Configure the principal delegation for inter-component calls

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

<b>QUESTION 27</b>	<b>NOT ANSWERED</b>	<b>Ask our Experts</b>
--------------------	---------------------	------------------------

Given two class declarations:

```
@RolesAllowed("super")
public class SuperClass {
    public void methodA() { ... }
    public void methodB() { ... }
    // other declarations
}
@Stateless
public class SubClass extends SuperClass implements MyInterface {
    @RolesAllowed("sub")
    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 can only be accessed by role sub; methodB and methodC can only be accessed by role super
- B. methodA and methodB can only be accessed by role super; methodC allows access from all clients
- C. methodA and methodB can only be accessed by roles sub and super, respectively; methodC allows access from all clients
- D. methodA can only be accessed by role sub; methodB and methodC allows access from all clients

**Your answer is incorrect.**

**Answer: C**

Explanation:

From the declaration of SuperClass, both methodA and B allow access from role super. However, the permission value for methodA is overridden in SubClass. So, methodA allows access from role sub, while methodB allows from role super.

methodC implementation is declared inside SubClass with no security metadata. As a result, it allows access from all clients.

The correct answer is: methodA and methodB can only be accessed by roles sub and super, respectively; methodC allows access from all clients

Submit your Feedback/Queries to our Experts

QUESTION 28

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Which of the following annotations does NOT specify roles allowed to invoke methods of the annotated bean class?

Please select :

- A. @DeclareRoles
- B. @RolesAllowed
- C. @PermitAll
- D. @DenyAll
- E. None of the above

**Your answer is incorrect.**

**Answer: A**

Explanation:

The @DeclareRoles annotation is used to declare logical security roles for the application. It has nothing to do with method permission. The following are descriptions of the remaining annotations:

@RolesAllowed: Specifies the list of roles permitted to access method(s) in an application. The value of the RolesAllowed annotation is a list of security role names. This annotation can be specified on a class or on method(s).

@PermitAll: Specifies that all security roles are allowed to invoke the specified method(s) - i.e that the specified method(s) are "unchecked". It can be specified on a class or on methods.

@DenyAll: Specifies that no security roles are allowed to invoke the specified method(s) - i.e that the methods are to be excluded from execution in the Java EE container.

The correct answer is: @DeclareRoles

Submit your Feedback/Queries to our Experts

QUESTION 29

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Which of the following session beans should be removed by their clients rather than by the container after being used?

Please select :

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

**Your answer is incorrect.**

**Answer: A**

Explanation:

The EJB 3.1 Specification (subsection 3.4.5) declares:

A client may remove a stateful session bean by invoking a method of its business interface designated as a @Remove method.

The lifecycle of a stateless session bean does not require that it be removed by the client. Removal of a stateless session bean instance is performed by the container, transparently to the client.

The lifecycle of a singleton session bean does not require that it be removed by the client. Removal of a singleton session bean instance is performed by the container, transparently to the client.

The correct answer is: Stateful session beans

Submit your Feedback/Queries to our Experts

QUESTION 30

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Given a session bean declaration:

```
// some annotations  
@Remote(Whizlabs.class)  
public class WhizlabsBean implements Whizlabs {  
    // class body  
}
```

A client obtains references to the Whizlabs business interface by the following dependency injections:

@EJB Whizlabs whizlabs1;

@EJB Whizlabs whizlabs2;

It is known that the following expression evaluates to true in a client method:

whizlabs1.equals(whizlabs2)

Whizlabs1.equals(whizlabs2)

What can you tell about the WhizlabsBean session bean?

Please select :

- A. It must be a stateless session bean
- B. It must be a stateful session bean
- C. It must be a singleton session bean
- D. It must be a stateless or singleton session bean
- E. It can be a session bean of any type

**Your answer is incorrect.**

**Answer: D**

Explanation:

All business object references of the same interface type for the same stateless session bean, or the same singleton session bean have the same object identity, which is assigned by the container. On the other hand, a stateful session object has a unique identity that is assigned by the container at the time the object is created. As a result, the whizlabs1.equals(whizlabs2) evaluates to true if the session bean is stateless or singleton, false if it is stateful.

The correct answer is: It must be a stateless or singleton session bean

Submit your Feedback/Queries to our Experts

QUESTION 31

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Given a situation where both @PostConstruct and @PreDestroy callback interceptor methods of a session bean run within transaction contexts. What is the type of that bean?

Please select :

- A. Stateful
- B. Stateless
- C. Singleton
- D. Stateless or singleton
- E. Stateful or stateless

**Your answer is incorrect.**

**Answer: C**

Explanation:

The EJB Specification (subsection 4.3.4) declares that the PostConstruct and PreDestroy lifecycle callback interceptor methods for stateless and stateful session beans execute in an unspecified transaction context. The PostConstruct and PreDestroy lifecycle callback interceptor methods for singleton beans execute in a transaction context determined by the bean's transaction management type and any applicable transaction attribute.

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 is NOT a requirement for a business method of a session bean exposed through a business interface or no-interface view?

Please select :

- A. Its name must not start with ejb
- B. It must be declared as public
- C. It must not be declared as final or static
- D. Its argument and return value types must be legal types for RMI/IIOP if the method corresponds to a method on a remote business interface
- E. Its argument and return value types for a method must be legal types for JAX-WS/JAX-RPC if the method corresponds to a method on the containing bean's web service endpoint
- F. None of the above

**Your answer is incorrect.**

**Answer: F**

Explanation:

As per the EJB 3.1 Specification (subsection 4.9.6), the session bean class may define zero or more business methods whose signatures must follow these rules:

- The method names can be arbitrary, but they must not start with "ejb" to avoid conflicts with the callback methods used by the EJB architecture.
- The business method must be declared as public.
- The method must not be declared as final or static.
- The argument and return value types for a method must be legal types for RMI/IIOP if the method corresponds to a business method on the session bean's remote business interface or remote interface.
- The argument and return value types for a method must be legal types for JAX-WS / JAX-RPC if the method is a web service method or corresponds to a method on the session bean's web service endpoint.
- The throws clause may define arbitrary application exceptions.

The correct answer is: None of the above

**QUESTION 33**

NOT ANSWERED

[Ask our Experts](#)

When does the EJBContext.getRollbackOnly method return true if it is invoked in an enterprise bean with container-managed transaction demarcation?

Please select :

- A. The current transaction has been marked for rollback by the containing enterprise bean only
- B. The current transaction has been marked for rollback by the containing enterprise bean or by other enterprise beans
- C. The current transaction has been marked anywhere, even outside the EJB container
- D. This method cannot be invoked in a bean with container-managed transaction demarcation

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

Explanation:

As per the EJB 3.1 Specification (subsection 13.3.4.3), an enterprise bean with container-managed transaction demarcation can use the getRollbackOnly method of its EJBContext object to test if the current transaction has been marked for rollback. The transaction might have been marked for rollback by the enterprise bean itself, by other enterprise beans, or by other components (outside of the EJB specification scope) of the transaction processing infrastructure.

The correct answer is: The current transaction has been marked anywhere, even outside the EJB container

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

**QUESTION 34**

NOT ANSWERED

[Ask our Experts](#)

Which THREE of the following transaction attributes can be used on @PostConstruct/@PreDestroy lifecycle callback interceptor methods of singleton session beans with container-managed transaction demarcation?

Please select :

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

**Your answer is incorrect.****Answer: D, E and F**

Explanation:

As per the EJB 3.1 Specification (subsection 13.3.7), for a singleton session bean's PostConstruct/PreDestroy lifecycle callback interceptor methods, only the REQUIRED, REQUIRES\_NEW, and NOT\_SUPPORTED transaction attributes may be used.

The correct answers are: NotSupported, Required, RequiresNew

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

**QUESTION 35**

NOT ANSWERED

[Ask our Experts](#)

Given a scenario where a client associated with transaction T1 calls a method of a stateful session bean instance, which is already associated with transaction T2. What will the container do with respect to transaction management?

Please select :

- A. Suspends T1, invokes the bean method in the context of T2, then resumes T1 when T2 is committed or rolled back
- B. Suspends T1, invokes the bean method in the context of T2, then resumes T1 when the method and all associated interceptor methods complete
- C. Suspends T2, invokes the bean method in the context of T1, then resumes T2 when T1 is committed or rolled back
- D. Suspends T2, invokes the bean method in the context of T1, then resumes T2 when the method and all associated interceptor methods complete

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

Explanation:

As per the EJB 3.1 Specification (subsection 13.6.1), if the client is associated with a transaction T1, and the instance is already associated with a transaction T2, the container suspends the client's transaction association and invokes the method with the transaction context that is associated with the instance (T2). The container resumes the client's transaction association (T1) when the method (together with any associated interceptor methods) completes. This case can never happen for a stateless session bean, singleton session bean, or a message-driven bean: it can only happen for a stateful session bean.

The correct answer is: Suspends T1, invokes the bean method in the context of T2, then resumes T1 when the method and all associated interceptor methods complete

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

**QUESTION 36**

NOT ANSWERED

[Ask our Experts](#)

Which of the following cannot be declared with a transaction attribute, given the enclosing enterprise bean has container-managed transaction demarcation?

Please select :

- A. Message listener methods of message-driven beans
- B. Timeout callback methods
- C. Session beans' web service endpoint methods
- D. Singleton session beans' PostConstruct/PreDestroy callback interceptor methods
- E. None of the above

**Your answer is incorrect.**

Answer: E

Explanation:

As per the EJB 3.1 Specification (subsection 13.3.7), a transaction attribute is a value associated with each of the following methods:

- a method of a bean's business interface
- a method exposed through the bean class no-interface view
- a message listener method of a message-driven bean
- a timeout callback method
- a session bean's web service endpoint method
- for beans written to the EJB 2.1 and earlier client view, a method of a session or entity bean's home or component interface
- a singleton session bean's PostConstruct/PreDestroy lifecycle callback interceptor methods

The correct answer is: None of the above

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

QUESTION 37

NOT ANSWERED

[Ask our Experts](#)

Which of the following is a valid fragment in the deployment descriptor?

Please select :

- A. <container-transaction>  
<method>  
<ejb-name>MyBean</ejb-name>  
<method-name>methodA</method-name>  
<method-name>methodB</method-name>  
</method>  
<trans-attribute>Required</trans-attribute>  
</container-transaction>
- B. <container-transaction>  
<method>  
<ejb-name>MyBean</ejb-name>  
<method-name>methodA</method-name>  
</method>  
<method>  
<ejb-name>MyBean</ejb-name>  
<method-name>methodA</method-name>  
<method-param/>  
</method>  
<trans-attribute>Required</trans-attribute>  
</container-transaction>
- C. <container-transaction>  
<method>  
<ejb-name>MyBean</ejb-name>  
<method-name>methodA</method-name>  
</method>  
<method>  
<ejb-name>MyBean</ejb-name>  
<method-name>methodA</method-name>  
<method-params></method-params>  
</method>  
<trans-attribute>Required</trans-attribute>  
</container-transaction>
- D. <container-transaction>  
<method>  
<ejb-name>MyBean</ejb-name>  
<method-name>methodA</method-name>  
</method>  
<trans-attribute>Required</trans-attribute>  
</container-transaction>  
<container-transaction>  
<method>  
<ejb-name>MyBean</ejb-name>  
<method-name>methodA</method-name>  
</method>  
<trans-attribute>Required</trans-attribute>  
</container-transaction>

**Your answer is incorrect.**

Answer: C

Explanation:

There can only be one method-name subelement inside a method element, thus option A is incorrect.

Option B is incorrect as the specification of the method-param element is invalid.

Option D is incorrect as there must be at most one container-transaction element that uses the same style for a given method name. To understand what a style is, please see subsection 13.3.7.2.1 of the EJB 3.1 Specification.

The correct answer is: <container-transaction>

```
<method>
<ejb-name>MyBean</ejb-name>
<method-name>methodA</method-name>
</method>
<method>
<ejb-name>MyBean</ejb-name>
<method-name>methodA</method-name>
<method-params></method-params>
</method>
<trans-attribute>Required</trans-attribute>
</container-transaction>
```

Submit your Feedback/Queries to our Experts

**QUESTION 38****NOT ANSWERED****Ask our Experts**

Given a method invocation chain among three methods in stateless session beans with container-managed transaction demarcation:

methodA (No) --> methodB (T1) --> methodC (T2)

Which of the following transaction attributes are possible for methodB and methodC, respectively?

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. Required and Required
- B. RequiresNew and RequiresNew
- C. NotSupported and Supports
- D. Never and Mandatory

**Your answer is incorrect.**

**Answer: B**

Explanation:

If the transaction attribute of methodC is Required, it will be executed in the same transaction context as methodC. Thus, option A is incorrect.

If the transaction attribute of methodB is NotSupported or Never, methodB would be executed without a transaction context. Therefore, options C and D are both incorrect.

References:

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

The correct answer is: RequiresNew and RequiresNew

Submit your Feedback/Queries to our Experts

**QUESTION 39****NOT ANSWERED****Ask our Experts**

Given a session bean declaration:

```
@Stateless
@TransactionManagement(TransactionManagementType.BEAN)
public class WhizlabsBean {
    @Resource
    private EJBContext context;
    public void doSomething() throws Exception {
```

```

UserTransaction userTransaction = context.getUserTransaction();
userTransaction.begin();
userTransaction.begin();
// do something
userTransaction.commit();
userTransaction.commit();
}
// other declarations
}

```

What happens when the userTransaction.begin method is invoked the second time?

Please select :

- A. The first transaction is suspended, then the second starts
- B. The second transaction starts, nested within the first one
- C. The container throws a NotSupportedException
- D. The container throws an IllegalStateException

**Your answer is incorrect.**

**Answer: C**

Explanation:

As per the EJB 3.1 Specification (subsection 13.6.1), when an instance attempts to start a transaction using the begin method of the javax.transaction.UserTransaction interface while the instance has not committed the previous transaction, the container must throw the javax.transaction.NotSupportedException in the begin method.

The correct answer is: The container throws a NotSupportedException

Submit your Feedback/Queries to our Experts

QUESTION 40

NOT ANSWERED

Ask our Experts

In which of the following situations a container-managed transaction is rolled back when an application exception is thrown?

Please select :

- A. Only if the @ApplicationException(rollback = true) annotation is declared on the exception definition
- B. Only if the setRollbackOnly method of the EJBContext interface is invoked
- C. Only if either or both conditions declared in options A and B are satisfied
- D. Conditions in options A and B are not the only ways to roll back the transaction

**Your answer is incorrect.**

**Answer: C**

As per the Java documentation, there are two ways to roll back a container-managed transaction. First, if a system exception is thrown, the container will automatically roll back the transaction. Second, by invoking the setRollbackOnly method of the EJBContext interface, the bean method instructs the container to roll back the transaction. If the bean throws an application exception, the rollback is not automatic but can be initiated by a call to setRollbackOnly.

In addition, the @ApplicationException annotation is applied to an exception to denote that it is an application exception and should be reported to the client directly (i.e., unwrapped). The rollback element of this annotation indicates whether the container should cause the transaction to rollback when the exception is thrown.

References:

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

<https://docs.oracle.com/javaee/6/api/javax/ejb/ApplicationException.html>

The correct answer is: Only if either or both conditions declared in options A and B are satisfied

Submit your Feedback/Queries to our Experts

QUESTION 41

NOT ANSWERED

Ask our Experts

Which of the following permissions should EJB containers deny?

Please select :

- A. java.security.AllPermission
- B. java.util.PropertyPermission
- C. java.lang.RuntimePermission
- D. java.net.SocketPermission

**Your answer is incorrect.**

**Answer: A**

Explanation:

The following is a table defining the security permissions for EJB containers:

Permission name	EJB Container policy
java.security.AllPermission	deny
java.awt.AWTPermission	deny
java.io.FilePermission	deny
java.net.NetPermission	deny

<code>java.util.PropertyPermission</code>	grant "read", deny all other
<code>java.lang.reflect.ReflectPermission</code>	deny
<code>java.lang.RuntimePermission</code>	grant "queuePrintJob", deny all other
<code>java.lang.SecurityPermission</code>	deny
<code>java.io.SerializablePermission</code>	deny
<code>java.net.SocketPermission</code>	grant "connect", deny all other

Note that only three permissions, including `PropertyPermission`, `RuntimePermission` and `SocketPermission`, are partially granted. All the others are denied.

The correct answer is: `java.security.AllPermission`

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

QUESTION 42

NOT ANSWERED

[Ask our Experts](#)

[MARK FOR REVIEW](#)

Which of the following operations is allowable for enterprise beans?

Please select :

- A. Obtain and set a class loader
- B. Create and set a security manager
- C. Change the input and output streams
- D. Connect to network servers

**Your answer is incorrect.**

**Answer: D**

Explanation:

As per the EJB 3.1 Specification (subsection 21.2.2), the enterprise bean must not attempt to create a class loader; obtain the current class loader; set the context class loader; set security manager; create a new security manager; stop the JVM; or change the input, output, and error streams. These functions are reserved for the EJB container. Allowing the enterprise bean to use these functions could compromise security and decrease the container's ability to properly manage the runtime environment.

The EJB architecture allows an enterprise bean instance to be a network socket client, but it does not allow it to be a network server. Allowing the instance to become a network server would conflict with the basic function of the enterprise bean - to serve the EJB clients.

The correct answer is: Connect to network servers

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

QUESTION 43

NOT ANSWERED

[Ask our Experts](#)

[MARK FOR REVIEW](#)

Which of the following features must be supported by the EJB 3.1 Lite API?

Please select :

- A. Remote access to session beans
- B. Asynchronous session bean invocations
- C. Web service endpoints
- D. EJB Timer Service
- E. None of the above

**Your answer is incorrect.**

**Answer: E**

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: None of the above

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

QUESTION 44

NOT ANSWERED

[Ask our Experts](#)

[MARK FOR REVIEW](#)

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

Please select :

- A. Enterprise bean's class
- B. Enterprise bean's web service endpoint interface
- C. Enterprise bean's type (session, message-driven or entity)
- D. None of the above

**Your answer is incorrect.**

**Answer: D**

Explanation:

As per the EJB 3.1 Specification (subsection 19.1), 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.

All the given options are about structural information.

**The correct answer is: None of the above**

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

**QUESTION 45**

NOT ANSWERED

[Ask our Experts](#)

Given an EJB-JAR file named beans.jar located in the classpath. This JAR file contains enterprise beans and their related types. The following code fragment is used to specify that module when initializing an embeddable container:

```
Properties props = new Properties();
props.setProperty(EJBContainer.MODULES, "beans.jar");
EJBContainer container = EJBContainer.createEJBContainer(props);
```

Which of the following statements is correct, provided there is no deployment descriptor?

Please select :

- A. The second argument passed in the props.setProperty method must be changed to "beans"
- B. The method invocation EJBContainer.createEJBContainer(props) is invalid
- C. If the beans.jar were not in the classpath, there would be no way for it to be programmatically added to the initialization process
- D. None of the above

**Your answer is incorrect.**

Answer: A

Explanation:

When specifying an EJB-JAR file for an embeddable container, the module name is the unqualified file name excluding the ".jar" extension. Therefore, option A is the correct answer.

If the beans.jar file were not in the classpath, it could still be added to the initialization by tweaking the given code as follows:

```
Properties props = new Properties();
File ejbJarFile = new File(<path-to-beans.jar>);
props.setProperty(EJBContainer.MODULES, ejbJarFile);
EJBContainer container = EJBContainer.createEJBContainer(props);
```

The correct answer is: The second argument passed in the props.setProperty method must be changed to "beans"

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

**QUESTION 46**

NOT ANSWERED

[Ask our Experts](#)

Which TWO of the following components must be supported by an embeddable container?

Please select :

- A. Session beans
- B. Message-driven beans
- C. Entity beans
- D. Java Persistence

**Your answer is incorrect.**

**Answer: A and D**

Explanation:

An embeddable container is required to support the EJB Lite API, thus it must be able to handle all components mandated by this API, including session beans and Java persistence.

The correct answers are: Session beans, Java Persistence

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

**QUESTION 47**

NOT ANSWERED

[Ask our Experts](#)

Given a WAR file structure:

whizlabs.war

```
WEB-INF
  web.xml
  ejb-jar.xml
  classes
  lib
  documents
```

Where can you put the class file containing the declaration of an enterprise bean, namely MyBean.class, into?

Please select :

- A. Directly under the assembly root
- B. Directly inside the WEB-INF directory
- C. Inside the classes directory
- D. Inside the lib directory
- E. Inside the documents directory
- F. All of the above

**Your answer is incorrect.**

**Answer: C**

Explanation:

Enterprise beans may be packaged within a WAR module as Java programming language class files or within a JAR file that is bundled within the WAR module. To include enterprise bean class files in a WAR module, the class files should be in the WEB-INF/classes directory. To include a JAR file that contains enterprise beans in a WAR module, add the JAR to the WEB-INF/lib directory of the WAR module.

The correct answer is: Inside the classes directory

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

[QUESTION 48](#)

[NOT ANSWERED](#)

[Ask our Experts](#)

Which TWO of the following are considered system exceptions?

Please select :

- A. RemoteException and its subclasses
- B. RuntimeException and its subclasses that are not application exceptions
- C. Exceptions that are annotated @SystemException
- D. Exceptions that are not annotated @ApplicationException

**Your answer is incorrect.**

**Answer: A and B**

As per the EJB 3.1 Specification (subsection 14.2.2), a system exception is an exception that is a java.rmi.RemoteException (or one of its subclasses) or a RuntimeException that is not an application exception.

There is no such an annotation like @SystemException, hence option C is incorrect.

Checked exceptions (except for RemoteException and its subclasses) are not automatically considered application exceptions, unless they are listed in the throws clause of the methods constituting a client view of an enterprise bean. Thus, option D is incorrect.

The correct answers are: RemoteException and its subclasses, RuntimeException and its subclasses that are not application exceptions

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

[QUESTION 49](#)

[NOT ANSWERED](#)

[Ask our Experts](#)

Given a session bean (with container-managed transaction demarcation) business method running outside a transaction. At some point the method throws an application exception (not specified as causing rollback). Which of the following actions the container will take?

Please select :

- A. Log the exception
- B. Re-throw the exception
- C. Discard the bean instance (except when the bean is singleton)
- D. Throw an EJBException to the caller

**Your answer is incorrect.**

**Answer: B**

Explanation:

The actions described in options A, C and D are what container would have done if a system application had been thrown. When an application exception is thrown, the container only re-throws the exception.

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

The correct answer is: Re-throw the exception

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

[QUESTION 50](#)

[NOT ANSWERED](#)

[Ask our Experts](#)

Which TWO of the following statements are correct about application exceptions?

Please select :

- A. The purpose of application exceptions is to give clients a chance to recover transactions from exceptions
- B. An application exception automatically results in marking the transaction for rollback
- C. Subelements of the application-exception deployment descriptor element override element values of the @ApplicationException annotation
- D. The deployment descriptor cannot override the values of elements of the @ApplicationException annotation

**Your answer is incorrect.**

Answer: A and C

Explanation:

As per the EJB 3.1 Specification (subsection 14.1.2), an application exception thrown by an enterprise bean instance should not automatically rollback a client's transaction unless the application exception was defined to cause transaction rollback. The client should typically be given a chance to recover a transaction from an application exception.

Subelements of the application-exception deployment descriptor element may be specified to override the values of elements of the @ApplicationException annotation.

The correct answers are: The purpose of application exceptions is to give clients a chance to recover transactions from exceptions, Subelements of the application-exception deployment descriptor element override element values of the @ApplicationException annotation

[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 management. Which TWO of the following statements are correct if a bean's business method throws an application exception while running in the context of a transaction that the container started immediately before dispatching the business method?

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 is not marked for rollback

**Your answer is incorrect.**

**Answer: A and D**

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, The transaction is not marked for rollback

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

QUESTION 52

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Given a session bean business method with a job of reading ZIP files uploaded from a client. If a file is malformed, the client must be given a chance to resend the file. Which exception should the bean method throw when receiving an invalid file?

Please select :

- A. DataFormatException
- B. EJBException
- C. RuntimeException
- D. RemoteException

**Your answer is incorrect.**

**Answer: A**

Explanation:

EJBException, RuntimeException and RemoteException are all system exceptions. These exceptions force the client transaction to roll back, making it impossible for the client to fix the problem. On the contrary, DataFormatException is an application exception, allowing the client to continue operating in the current transaction.

The correct answer is: DataFormatException

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

QUESTION 53

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Which annotation can be used to control the order of destruction of singleton session beans?

Please select :

- A. @Destruction
- B. @DependsOn
- C. @Remove
- D. @Release

**Your answer is incorrect.**

**Answer: B**

Explanation:

In addition to managing the order of singleton bean initialization, the @DependsOn also controls the order of destruction. This annotation guarantees that all singleton beans with which a singleton bean has a DependsOn relationship are still available during PreDestroy.

The correct answer is: @DependsOn

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

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

Given a singleton session bean:

```
@Singleton
@AccessTimeout(1000)
public class Whizlabs {
    public void methodA() { ... }
    public void methodB() { ... }
    // other declarations
}
```

It is known that both methodA and methodB take minutes to complete. At some point, a client invokes methodA, and another client invokes methodB right after. What happens in such a situation?

Please select :

- A. Both methods run right after they are invoked
- B. Both methods must wait a second before running
- C. methodA runs right after it is invoked, while methodB waits a second before doing its job
- D. methodA runs right after it is invoked, and methodB must wait until methodA completes to proceed
- E. methodA runs right after it is invoked, and an exception is thrown after one second

**Your answer is incorrect.**

**Answer: E**

**Explanation:**

The Whizlabs singleton bean is not explicitly assigned a concurrency management type and concurrency lock. As such, they take default values: the concurrency type is container-managed, and the lock type is WRITE.

When the first request accesses methodA, it acquires a WRITE lock, blocking any other request. Thus, the invocation on methodB cannot proceed. Due to the specification of the @AccessTimeout annotation, this invocation waits one second and ConcurrentAccessTimeoutException is thrown as this duration is not enough for methodA to complete.

The correct answer is: methodA runs right after it is invoked, and an exception is thrown after one second

Submit your Feedback/Queries to our Experts

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

Given two singleton session beans packaged in different EJB-JAR files within the same EAR:

```
// packaged in a.jar
Singleton
public class A { ... }

// packaged in b.jar
Singleton
public class B { ... }
```

Which of the following annotations must be declared on class B to indicate that it must be initialized after A?

Please select :

- A. @DependsOn("a!A")
- B. @DependsOn("a#A")
- C. @DependsOn("a.jar#a")
- D. The described goal is unachievable

**Your answer is incorrect.**

**Answer: C**

**Explanation:**

The annotation in option C demonstrates the correct syntax to refer to an enterprise bean packaged within a different module in the same application.

The correct answer is: @DependsOn("a.jar#a")

Submit your Feedback/Queries to our Experts

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

Given two class declarations:

```
@Lock(READ)
public class SomeClass {
    public void methodA () { ... }
    @Lock(WRITE)
    public void methodB () { ... }
    // other declarations
}

@Singleton
public class Whizlabs extends SomeClass {
    @Lock(READ)
    public void methodA () { ... }
```

```
public void methodC () { ... }  
// other declarations  
}
```

What are concurrency locking attributes of methodA, methodB and methodC of bean Whizlabs, respectively?

Please select :

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

**Your answer is incorrect.**

**Answer: A**

Explanation:

By default, the concurrency attribute of a method is WRITE, thus the attribute of methodC is WRITE. Meanwhile, the attributes of methodA and methodB are READ and WRITE as explicitly specified in the method level.

The correct answer is: READ, WRITE, WRITE

Submit your Feedback/Queries to our Experts

QUESTION 57

NOT ANSWERED

Ask our Experts

Which of the following is NOT a benefit the EJB technology?

Please select :

- A. Provide system-level services to enterprise beans
- B. Make the client thinner
- C. Improve component portability
- D. None of the above

**Your answer is incorrect.**

**Answer: D**

Explanation:

As per the Oracle's Java EE 6 Tutorial, enterprise beans simplify the development of large, distributed applications.

First, because the EJB container provides system-level services to enterprise beans, the bean developer can concentrate on solving business problems. The EJB container, rather than the bean developer, is responsible for system-level services, such as transaction management and security authorization.

Second, because the beans rather than the clients contain the application's business logic, the client developer can focus on the presentation of the client. The client developer does not have to code the routines that implement business rules or access databases. As a result, the clients are thinner, a benefit that is particularly important for clients that run on small devices.

Third, because enterprise beans are portable components, the application assembler can build new applications from existing beans. Provided that they use the standard APIs, these applications can run on any compliant Java EE server.

References:

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

The correct answer is: None of the above

Submit your Feedback/Queries to our Experts

QUESTION 58

NOT ANSWERED

Ask our Experts

Which of the following is NOT an attribute of the @Schedule annotation?

Please select :

- A. month
- B. dayOfMonth
- C. hour
- D. minute
- E. second
- F. millisecond

**Your answer is incorrect.**

**Answer: F**

Explanation:

Note that even though timer duration is expressed in terms of milliseconds, timer service is not intended for uses with such a precision. As a result, the millisecond attribute is unnecessary and not specified for the @Schedule annotation. The unit of millisecond is used for timer duration just because it is the unit of time granularity used by APIs of the Java SE platform.

References:

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

The correct answer is: millisecond

Submit your Feedback/Queries to our Experts

**QUESTION 59**

NOT ANSWERED

**Ask our Experts**

Which TWO of the following requirements are NOT correct about a timeout callback method?

Please select :

- A. It must have signature void <method>(Timer timer), where <method> designates the method name
- B. It must have non-private level access
- C. It must not be declared as final or static
- D. It must not throw application exceptions

**Your answer is incorrect.**

Answer: A and B

Explanation:

As per the EJB 3.1 Specification (subsection 18.2.5.3), a timeout callback method must have one of the two signatures below, where <METHOD> designates the method name.

`void <METHOD>()`

`void <METHOD>(Timer timer)`

A timeout callback method can have public, private, protected, or package level access. A timeout callback method must not be declared as final or static.

Timeout callback methods must not throw application exceptions.

The correct answers are: It must have signature void <method>(Timer timer), where <method> designates the method name, It must have non-private level access

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

**QUESTION 60**

NOT ANSWERED

**Ask our Experts**

Given a timeout callback method with automatic timer creation:

```
@Schedule(dayOfWeek = "[6-0]")
public void timeoutCallback(Timer timer) { ... }
```

When is the above callback triggered?

Please select :

- A. At 0:00 every day
- B. At 0:00 every weekday
- C. At 0:00 every Saturday and Sunday
- D. The value of dayOfWeek attribute is invalid

**Your answer is incorrect.**

**Answer: C**

Explanation:

When constraining an attribute of the @Schedule annotation to a range of values [x-y] with x being larger than y, [x-y] is equivalent to [x-max],[min-y], where max is the largest value of the attribute and min is the smallest. Therefore, [6-0] is the same as [6-7],[0-0], or 6,7,0. Both 7 and 0 refer to Sunday, while 6 corresponds to Saturday.

References:

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

The correct answer is: At 0:00 every Saturday and Sunday

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

[Finish review](#)

#### Company

[About Us](#)

#### Contact us

[Live Chat](#)

[support@whizlabs.com](mailto:support@whizlabs.com)

#### Communities

[Discussions](#)  
[Blog](#)

#### Follow Us

