



OCEEJBD 6 Practice Test

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

0.



Started on Saturday, 20 October 2018, 8:50 PM

State Finished

Completed on Saturday, 20 October 2018, 8:50 PM

Time taken 23 secs

Grade 0.00 out of 60.00 (0%)

Result

Quiz navigation

[Show one page at a time](#)

[Show all correct answers](#)

[Show all incorrect answers](#)

[Finish review](#)

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32
33	34	35	36
37	38	39	40
41	42	43	44
45	46	47	48

QUESTION 1

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Which of the following statements is correct about clients of session beans?

Please select :

- A. Clients do not have direct access to session bean instances
- B. Clients of a session bean must be written in the Java programming language
- C. A session bean can provide only one client view
- D. Clients can invoke a session bean synchronously only

Your answer is incorrect.

Answer: A

The following are several excerpts from the EJB 3.1 Specification (subsection 3.1):

- A client never directly accesses instances of the session bean's class. A client accesses a session object through the session bean's client view.
- A remote client of a session bean can be another enterprise bean deployed in the same or different container; or it can be an arbitrary Java program, such as an application, applet, or servlet. The client view of a session bean can also be mapped to non-Java client environments, such as CORBA clients that are not written in the Java programming language.
- While it is possible to provide more than one client view for a session bean, typically only one will be provided.
- A client can invoke a session bean synchronously or asynchronously. An asynchronous method can return a Future<V> object that allows the client to retrieve a result value, check for exceptions, or attempt to cancel an in-progress invocation.

The correct answer is: Clients do not have direct access to session bean instances

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

QUESTION 2

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Given a session bean:

```
@Stateless
@LocalBean
public class WhizlabsBean implements Whizlabs {
    // bean class body
}
```

Which TWO of the following statements can be used to obtain a reference to the bean's no-interface view, provided there is no associated deployment descriptor?

Please select :

- A. @EJB
- WhizlabsBean whizlabs;
- B. @EJB Whizlabs whizlabs;
- C. WhizlabsBean whizlabs = (WhizlabsBean) sessionContext.lookup("ejb/Whizlabs");
- Where ejb/Whizlabs is JNDI name of the bean in the client's namespace and sessionContext is a SessionContext instance
- D. WhizlabsBean whizlabs = new WhizlabsBean();
- E. Whizlabs whizlabs = new WhizlabsBean();

Your answer is incorrect.

Answer: A and C

Explanation:

The question asks about obtaining a reference to the bean's no-interface view, thus the Whizlabs interface is unrelated. Option B is incorrect, then.

Despite the fact that the client reference for the no-interface view has type <bean-class>, the client does not have direct access to the bean, thus cannot use the new operator to acquire the reference. As such, options D and E are both incorrect.

Please note that a client can only obtain a reference to a session bean's no-interface view or business interface through dependency injection or lookup in the JNDI namespace.

lookup in the JNDI namespace.

The correct answers are: @EJB

WhizlabsBean whizlabs; WhizlabsBean whizlabs = (WhizlabsBean) sessionContext.lookup("ejb/Whizlabs");

Where ejb/Whizlabs is JNDI name of the bean in the client's namespace and sessionContext is a SessionContext instance

Submit your Feedback/Queries to our Experts

QUESTION 3

NOT ANSWERED

Ask our Experts

Assume you have obtained an instance of MyLocal, which is a local business interface of a session bean. Which of the following can take in an argument of type MyLocal?

Please select :

- A. Methods of local business interfaces only
- B. Methods of local business interfaces and no-interface views
- C. Methods of remote business interfaces only
- D. Methods of local and remote business interfaces
- E. Methods of local and remote business interfaces, as well as no-interface views

Your answer is incorrect.

Answer: B

Explanation:

As per the EJB 3.1 Specification (subsection 3.4.3), a reference to a session bean's business interface may be passed as a parameter or return value of a business interface method. If the reference is to a session bean's local business interface, the reference may only be passed as a parameter or return value of a local business interface method or a no-interface view method.

The correct answer is: Methods of local business interfaces and no-interface views

Submit your Feedback/Queries to our Experts

QUESTION 4

NOT ANSWERED

Ask our Experts

Given a session bean declaration:

@Stateful

```
public class WhizlabsBean {  
    String url = "whizlabs.com";  
    // other declarations  
}
```

Which modifier should be declared on the url field to make it accessible to clients accessing the no-interface view of WhizlabsBean?

Please select :

- A. public
- B. final
- C. static
- D. No modifiers are needed since the client access is local
- E. The described goal cannot be achieved

Your answer is incorrect.

Answer: E

Explanation:

The EJB 3.1 Specification (subsection 3.4.4) declares: When interacting with a reference to the no-interface view, the client must not make any assumptions regarding the internal implementation of the reference, such as any instance-specific state that may be present in the reference. Although the reference object is type-compatible with the corresponding bean class type, there is no prescribed relationship between the internal implementation of the reference and the implementation of the bean instance.

The correct answer is: The described goal cannot be achieved

Submit your Feedback/Queries to our Experts

QUESTION 5

NOT ANSWERED

Ask our Experts

Which of the following is NOT a method of the SessionSynchronization interface?

Please select :

- A. afterBegin
- B. beforeBegin
- C. afterCompletion
- D. beforeCompletion
- E. None of the above

Your answer is incorrect.

Answer: B

Explanation:

The SessionSynchronization interface defines three methods:

- afterBegin: Notifies a stateful session bean instance that a new transaction has started, and that the subsequent business methods on the instance will be invoked in the context of the transaction
- afterCompletion: Notifies a stateful session bean instance that a transaction commit protocol has completed, and tells the instance whether the transaction has been committed or rolled back
- beforeCompletion: Notifies a stateful session bean instance that a transaction is about to be committed

References:

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

The correct answer is: beforeBegin

Submit your Feedback/Queries to our Experts

QUESTION 6

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Which of the following statements is correct about the @Remove annotation when used within a stateful session bean class?

Please select :

- A. Only one method of a class can be annotated with the @Remove annotation
- B. If the remove-method element is used in the associated deployment descriptor, all @Remove annotations in this bean class are ignored
- C. An @Remove method must take in no argument and not return a value
- D. If the retainIfException element of the @Remove annotation is set to true, the instance is not removed if any exception is thrown from the annotated method
- E. None of the above

Your answer is incorrect.

Answer: E

Explanation:

There are no such constraints as described in options A and C, while any remove methods defined for the bean by means of the remove-method deployment descriptor element apply in addition to those defined by means of annotations. Thus, options A, B and C are incorrect.

The true value of the retainIfException element just prevents the bean instance from being removed if an application exception is thrown. The bean is always discarded if a system exception is thrown, no matter to which value this element is set.

The correct answer is: None of the above

Submit your Feedback/Queries to our Experts

QUESTION 7

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Which of the following statements is correct about the passivation of a stateful session bean?

Please select :

- A. A bean with an extended persistence context cannot be passivated
- B. A bean can only be passivated when it is not in a transaction
- C. A bean is passivated after a method is completed, and activated when the current client accesses the bean the next time
- D. A bean must use container-managed transaction demarcation to be passivated when running within a transaction

Your answer is incorrect.

Answer: B

Explanation:

The container decides upon its own discretion regarding when to passivate stateful session beans. Normally, a container does that to better manage the size of its working set. The EJB 3.1 Specification just mandates that a container may only passivate a stateful session bean instance when the instance is not in a transaction.

A bean with an extended persistence context can still be passivated, provided the following conditions are met:

- All the entities in the persistence context are serializable
- The EntityManager is serializable

The correct answer is: A bean can only be passivated when it is not in a transaction

Submit your Feedback/Queries to our Experts

QUESTION 8

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Given a business interface declaration:

```
@Remote  
public interface Whizlabs {  
    void doSomething();  
}
```

This interface represents a business interface client view for an enterprise bean:

```
@Stateless  
public class WhizlabsBean implements Whizlabs {  
    public void doSomething() { ... }  
    // other declaration  
}
```

Assume you want to designate the doSomething business method as asynchronous. This target cannot be achieved if you specify the @Asynchronous annotation on the declaration of:

Please select :

- A. The Whizlabs interface
- B. The WhizlabsBean class
- C. The doSomething method of the Whizlabs interface
- D. The doSomething method of the WhizlabsBean class
- E. None of the above

Your answer is incorrect.

Answer: E

Explanation:

The EJB 3.1 Specification (subsection 4.5.1) declares the following:

The @Asynchronous annotation is used to designate which business methods are asynchronous. @Asynchronous can be applied to a business method of a bean class or to a method of a Local/Remote business interface.

In addition, @Asynchronous can be applied to the class level of a bean class (or superclass) and to the class level of a particular Remote/Local business interface (or superinterface). If @Asynchronous is applied at the class level, all methods declared on that specific class or interface are asynchronous.

The correct answer is: None of the above

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

QUESTION 9

[NOT ANSWERED](#)

[Ask our Experts](#)

In which of the following situations a method annotated as @PreDestroy within a stateful session bean is executed?

Please select :

- A. The EJB container crashes
- B. A system exception is thrown from a method of the bean
- C. A timeout of client inactivity while the instance is in the passivated state
- D. None of the above

Your answer is incorrect.

Answer: D

Explanation:

The EJB 3.1 Specification (subsection 4.6.3) declares that the bean provider cannot assume that the container will always invoke the PreDestroy lifecycle callback interceptor method(s) (or ejbRemove method) for a session bean instance. The following scenarios result in the PreDestroy lifecycle callback interceptor method(s) not being called for an instance:

- A crash of the EJB container.
- A system exception thrown from the instance's method to the container.
- A timeout of client inactivity while the instance is in the passive state. The timeout is specified by the Deployer in an EJB container implementation-specific way.

The correct answer is: None of the above

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

QUESTION 10

[NOT ANSWERED](#)

[Ask our Experts](#)

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

Please select :

- A. name
- B. mappedName
- C. description
- D. None of the above

Your answer is incorrect.

Answer: D

Explanation:

The @Stateful annotation, the same as the @Stateless and @Singleton annotations, defines the following elements:

- description: A string describing the stateful session bean
- mappedName: A product specific name (e.g. global JNDI name) that this session bean should be mapped to
- name: The ejb-name for this bean

References:

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

The correct answer is: None of the above

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

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

Which TWO of the following JMS objects should be managed administratively?

Please select :

- A. Connection factories
- B. Connections
- C. Sessions
- D. Destinations
- E. Message producers
- F. Message consumers

Your answer is incorrect.

Answer: A and D

Explanation:

As per the Oracle's Java EE 6 Tutorial, two parts of a JMS application, destinations and connection factories, are best maintained administratively rather than programmatically. The technology underlying these objects is likely to be very different from one implementation of the JMS API to another. Therefore, the management of these objects belongs with other administrative tasks that vary from provider to provider.

References:

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

The correct answers are: Connection factories, Destinations

Submit your Feedback/Queries to our Experts

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

Which of the following statements is correct about JMS?

Please select :

- A. The kind of a connection factory must be the same as that of the corresponding destination; that is, a QueueConnectionFactory resource can only be used with a Queue, and a TopicConnectionFactory resource can only be used with a Topic
- B. A QueueConnectionFactory resource can be used with a Topic, and a TopicConnectionFactory resource can be used with a Queue; the behavior of the application depends on the kind of destination
- C. A QueueConnectionFactory resource can be used with a Topic, and a TopicConnectionFactory resource can be used with a Queue; the behavior of the application depends on the kind of connection factory

Your answer is incorrect.

Answer: B

Explanation:

As per the Oracle's Java EE 6 Tutorial, with the common interfaces, you can mix or match connection factories and destinations. That is, in addition to using the ConnectionFactory interface, you can inject a QueueConnectionFactory resource and use it with a Topic, and you can inject a TopicConnectionFactory resource and use it with a Queue. The behavior of the application will depend on the kind of destination you use and not on the kind of connection factory you use.

References:

<http://docs.oracle.com/javaee/6/tutorial/doc/bncej.html#bncei>

The correct answer is: A QueueConnectionFactory resource can be used with a Topic, and a TopicConnectionFactory resource can be used with a Queue; the behavior of the application depends on the kind of destination

Submit your Feedback/Queries to our Experts

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

Which of the following methods can be invoked on a javax.jms.Session object to create a durable consumer?

Please select :

- A. createConsumer(boolean, Destination, String)
- B. createSubscriber(boolean, Topic, String)
- C. createDurableConsumer(Topic, String)
- D. createDurableSubscriber(Topic, String)

Your answer is incorrect.

Answer: D

Explanation:

The Session interface defines two overloaded createDurableSubscriber methods to create durable subscriptions:

- createDurableSubscriber(Topic topic, String name) throws JMSException
- createDurableSubscriber(Topic topic, String name, String messageSelector, boolean noLocal) throws JMSException

References:

<http://docs.oracle.com/javaee/6/tutorial/doc/bnccfu.html#bnccg>

The correct answer is: `createDurableSubscriber(Topic, String)`

Submit your Feedback/Queries to our Experts

QUESTION 14

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

How a message listener method and its containing message-driven bean must be declared to include the message receipt as part of the transaction within which the listener method body runs?

Please select :

- A. The bean must use container-managed transaction demarcation and the method's transaction attribute is Required
- B. The bean must use container-managed transaction demarcation and the method's transaction attribute is Supports
- C. The bean must use bean-managed transaction demarcation
- D. The described goal is unachievable

Your answer is incorrect.

Answer: A

Explanation:

As per the EJB 3.1 Specification (subsection 5.4.12), when a message-driven bean using bean-managed transaction demarcation uses the `UserTransaction` interface to demarcate transactions, the message receipt that causes the bean to be invoked is not part of the transaction. If the message receipt is to be part of the transaction, container-managed transaction demarcation with the REQUIRED transaction attribute must be used.

The correct answer is: The bean must use container-managed transaction demarcation and the method's transaction attribute is Required

Submit your Feedback/Queries to our Experts

QUESTION 15

NOT ANSWERED

MARK FOR REVIEW

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.rollback();
```

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 to update to database 2. Which database(s) is updated after the above fragment is executed?

Please select :

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

Your answer is incorrect.

Answer: D

Explanation:

Since the transaction is rolled back, the update statement executed against database 1 is not executed, nor is the message sent. As a result, no updates are made to database 1 or database 2.

The correct answer is: Neither of the databases

Submit your Feedback/Queries to our Experts

QUESTION 16

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Given a message-driven bean declaration:

```
@MessageDriven
public class Whizlabs implements MessageListener, MessageDrivenBean {
    public void setMessageDrivenContext(MessageDrivenContext context) { ... }
    public void ejbRemove() { ... }
    public void onMessage(Message message) { ... }
    @PreDestroy
    public void preDestroy() { ... }
}
```

Which clean-up method is invoked by the container when an instance of the above bean class is destroyed?

Please select :

- A. ejbRemove only
- B. preDestroy only
- C. ejbRemove, then preDestroy
- D. preDestroy, then ejbRemove
- E. The given bean declaration is invalid

Your answer is incorrect.

Answer: E

Explanation:

As per the EJB 3.1 Specification (subsection 6.4.6), if the message-driven bean implements the MessageDrivenBean interface, the @PreDestroy annotation can only be applied to the ejbRemove method. Similar requirements apply to use of deployment descriptor metadata as an alternative to the use of annotations.

The correct answer is: The given bean declaration is invalid

Submit your Feedback/Queries to our Experts

QUESTION 17

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

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

Please select :

- A. They cannot be invoked by clients
- B. They have no conversational state
- C. They must be created and managed by the container
- D. They cannot contain state across the handling of different client messages
- E. None of the above

Your answer is incorrect.

Answer: D

Explanation:

The following excerpt is from the EJB 3.1 Specification (subsection 5.1):

Message-driven beans are anonymous. They have no client-visible identity.

Message-driven bean instances have no conversational state. This means that all bean instances are equivalent when they are not involved in servicing a client message.

A message-driven bean instance is created by the container to handle the processing of the messages for which the message-driven bean is the consumer. Its lifetime is controlled by the container.

A message-driven bean instance has no state for a specific client. However, the instance variables of the message-driven bean instance can contain state across the handling of client messages. Examples of such state include an open database connection and a reference to an enterprise bean.

The correct answer is: They cannot contain state across the handling of different client messages

Submit your Feedback/Queries to our Experts

QUESTION 18

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Which TWO of the following statements is NOT correct about interceptor classes?

Please select :

- A. Any number of interceptor classes may be associated with a target class
- B. An interceptor class must have a public no-argument constructor
- C. Interceptors defined using metadata annotations and the deployment descriptor are portable across all EJB compliant containers
- D. The @Interceptor annotation must be used to explicitly designate a class as an interceptor class
- E. The lifecycle of an interceptor instance is the same as that of the target class instance with which it is associated

Your answer is incorrect.

Answer: C and D

Explanation:

The following are several extracts taken from the Overview and Interceptor Life Cycle sections of the Interceptors 1.1 Specification:

- Any number of interceptor classes may be associated with a target class.
- An interceptor class must have a public no-arg constructor.
- An interceptor implementation is not required to support the deployment descriptor approach to specifying interceptor metadata.
- The @Interceptor annotation may be used to explicitly designate a class as an interceptor class. Support for this annotation is not required.
- The lifecycle of an interceptor instance is the same as that of the target class instance with which it is associated.

References:

<http://docs.oracle.com/javaee/6/tutorial/doc/gkigq.html>

The correct answers are: Interceptors defined using metadata annotations and the deployment descriptor are portable across all EJB compliant containers. The @Interceptor annotation must be used to explicitly designate a class as an interceptor class

Submit your Feedback/Queries to our Experts

QUESTION 19

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Given a singleton session bean:

```

@Singleton
public class MyBean {
    @Interceptors(MyInterceptor.class)
    public void methodA() { ... }
    @Interceptors(MyInterceptor.class)
    public void methodB() { ... }
    // other declarations
}

```

Assume methods methodA and methodB are each invoked once by a local client. How many instances of the MyInterceptor interceptor class are created?

Please select :

- A. One
- B. Two
- C. It is container-specific
- D. An interceptor class cannot be declared on two different methods of the same bean

Your answer is incorrect.

Answer: A

Explanation:

The lifecycle of an interceptor instance is the same as that of the target class instance with which it is associated. When the target instance is created, a corresponding interceptor instance is created for each associated interceptor class. These interceptor instances are destroyed when the target instance is removed. In the given scenario, there is only one instance of MyBean, hence just one instance of MyInterceptor is created.

The correct answer is: One

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

[QUESTION 20](#)

[NOT ANSWERED](#)

[MARK FOR REVIEW](#)

[Ask our Experts](#)

Which of the following interceptor methods is NOT valid?

Please select :

- A. @AroundInvoke
Object intercept(InvocationContext context) throws Exception { ... }
- B. @AroundTimeout
Object intercept(InvocationContext context) { ... }
- C. @PostConstruct
public void intercept(InvocationContext context) { ... }
- D. @PreDestroy
public void intercept() { ... }
- E. None of the above

Your answer is incorrect.

Answer: B

Explanation:

The around-timeout interceptor method in option B is invalid as the throws clause is missing. All other declarations are valid interceptors.

The correct answer is: @AroundTimeout

Object intercept(InvocationContext context) { ... }

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

[QUESTION 21](#)

[NOT ANSWERED](#)

[MARK FOR REVIEW](#)

[Ask our Experts](#)

Given an interceptor class:

```

@Interceptor
public class MyInterceptor {
    @AroundInvoke
    Object intercept(InvocationContext context) throws Exception {
        return 0;
    }
}

```

The above interceptor class is specified in a bean:

```

@Stateless
public class Whizlabs implements A {
    @Interceptors(MyInterceptor.class)
    public String lowerCase(String text) {
        return text.toLowerCase();
    }
    // other declarations
}

```

What is returned when a client invokes the Whizlabs.lowerCase method with argument "WHIZLABS"

Please select :

- A. 0
- B. "0"
- C. "whizlabs"
- D. "WHIZLABS"
- E. An exception is thrown

Your answer is incorrect.

Answer: E

Since the MyInterceptor.intercept method 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. In the given scenario, the result is 0; however this value must be cast to a String - the return type of the target method. This type of casting is not allowed, resulting in a ClassCastException.

The correct answer is: An exception is thrown

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

QUESTION 22

[NOT ANSWERED](#)

[MARK FOR REVIEW](#)

[Ask our Experts](#)

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

Please select :

- A. Manage principals
- B. Provide a security domain and one or more principal realms to enterprise beans
- C. Assign the security domain and principal realm to the application
- D. Manage resource access

Your answer is incorrect.

Answer: C

Explanation:

As per the EJB 3.1 Specification (subsection 17.4.1), the deployer is responsible for assigning the security domain and principal realm to an enterprise bean application. Multiple principal realms within the same security domain may exist, for example, to separate the realms of employees, trading partners, and customers. As such, option C is the correct answer.

Managing principals is a task of the system administrator.

Providing a security domain and one or more principal realms to enterprise beans is a requirement for the EJB container.

Managing resource access is a job of the bean provider.

The correct answer is: Assign the security domain and principal realm to the application

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

QUESTION 23

[NOT ANSWERED](#)

[MARK FOR REVIEW](#)

[Ask our Experts](#)

Given an enterprise bean:

```
@Stateless
@DeclareRoles("alpha")
@RolesAllowed("beta")
public class Whizlabs {
    // class body
}
```

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

```
<assembly-descriptor>
    <security-role>
        <role-name>gamma</role-name>
    </security-role>
</assembly-descriptor>
```

Which security roles must be mapped to principals and/or groups of principals in the operational environment that access the Whizlabs bean?

Please select :

- A. alpha only
- B. beta only
- C. gamma only
- D. alpha and beta
- E. alpha and gamma
- F. alpha, beta and gamma

Your answer is incorrect.

Answer: F

Explanation:

As per the EJB 3.1 Specification (subsection 17.3.1), 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 used 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.

The correct answer is: alpha, beta and gamma

Submit your Feedback/Queries to our Experts

QUESTION 24

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

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

Please select :

- A. Portability: The Java EE platform must support the Write Once, Run Anywhere application property
- B. Isolation: The Java EE platform should be able to perform authentication and access control according to instructions established by the deployer, and managed by the system administrator
- C. Abstraction: Security requirements are logically specified using deployment descriptors, specifying how security roles and access control are to be mapped into the operational environment
- D. Security technology: The Java EE platform must support the Kerberos authentication protocol
- E. Independence: Required security behaviors and deployment contracts should be implementable using a variety of popular security technologies

Your answer is incorrect.

Answer: D

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 specification does not mandate a specific security technology, such as Kerberos, PK, NIS+, or NTLM.

The correct answer is: Security technology: The Java EE platform must support the Kerberos authentication protocol

Submit your Feedback/Queries to our Experts

QUESTION 25

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Which interface can be used as the entry point for methods within an EJB enterprise bean to programmatically access the security context of their caller?

Please select :

- A. Context
- B. SecurityContext
- C. EJBContext
- D. Principal

Your answer is incorrect.

Answer: C

Context is an interface representing a naming context in a Java SE environment. It has nothing to do with EJB security. Hence, option A is incorrect.

There is a SecurityContext interface defined for the Java EE platform, but it is used for client access to web services. Thus, option B is incorrect.

Principal is an interface representing the abstract notion of a principal. It is used to get details about a particular user. An instance of this type, however, cannot be retrieved without invoking method EJBContext.getCallerPrincipal. Option D is incorrect, therefore.

Note that the only way to access the security context of a caller is to use one of two methods defined in the EJBContext interface: getCallerPrincipal() and isCallerInRole(String roleName).

The correct answer is: EJBContext

Submit your Feedback/Queries to our Experts

QUESTION 26

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Given an enterprise bean declaration:

```
@Stateless  
@RunAs("whizlabs")  
public class WhizlabsBean {  
    @Resource  
    SessionContext context;  
    public void doSomething() {  
        callerPrincipal = context.getCallerPrincipal();  
    }  
}
```

```

    context.getCallerPrincipal();
    // do something
}
// other declarations
}

```

What happens if the context.getCallerPrincipal method is invoked by an unauthenticated client, provided there is no security constraints set on this method and no deployment descriptor exists?

Please select :

- A. It returns a null value
- B. It returns an identity set by the container
- C. It returns an identity assigned to the whizlabs role
- D. It throws an IllegalStateException

Your answer is incorrect.

Answer: B

Explanation:

As per the EJB 3.1 Specification (subsection 17.2.5.1), if the security identity has not been established, getCallerPrincipal returns the container's representation of the unauthenticated identity. Note that getCallerPrincipal returns the principal that represents the caller of the enterprise bean, not the principal that corresponds to the run-as security identity for the bean, if any.

The correct answer is: It returns an identity set by the container

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

QUESTION 27

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Which of the following is NOT a requirement for a session bean class?

Please select :

- A. It must be a top-level class with public access modifier
- B. It must not be final nor abstract
- C. It must have a public no-argument constructor
- D. It must not define the finalize method
- E. It must implement all business interfaces of the bean
- F. None of the above

Your answer is incorrect.

Answer: E

Explanation:

The following are the requirements for a session bean class, taken from the EJB 3.1 Specification (subsection 4.9.2):

- The class must be defined as public, must not be final, and must not be abstract. The class must be a top-level class.
- The class must have a public constructor that takes no parameters. The container uses this constructor to create instances of the session bean class.
- The class must not define the finalize() method.
- The class must implement the bean's business interface(s) or the methods of the bean's business interface(s), if any.
- The class must implement the business methods of the bean's EJB 2.1 client view interfaces, if any.

Notice the requirement on implementing business interfaces. A bean class usually implements all of its business interfaces, but this is not a requirement. The class just needs to provide implementation for all methods defined by those business interfaces, providing the signatures are the same. This means that the return type of a bean class method may be unrelated to the return type of the corresponding method defined by an interface.

The correct answer is: It must implement all business interfaces of the bean

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

QUESTION 28

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Given a bean class declaration:

```

@Stateless
public class WhizlabsBean implements Whizlabs1, Whizlabs2 { ... }

```

Neither Whizlabs1 nor Whizlabs2 is annotated as @Local or @Remote, and there is no associated deployment descriptor. Which of the following statements is correct about client views of the above bean?

Please select :

- A. The bean does not expose any client view
- B. The bean exposes a no-interface view
- C. The bean exposes both implemented interfaces as local business interface client views
- D. The bean exposes both implemented interfaces as remote business interface client views

Your answer is incorrect.

Answer: A

Explanation:

Explanation:

A bean class is permitted to have more than one interface. If a bean class has more than one interface, excluding Serializable, Externalizable and any of the interfaces defined by the javax.ejb package, any business interface of the bean class must be explicitly designated as a business interface of the bean by means of the @Local or @Remote annotation on the bean class or interface or in the deployment descriptor.

Since WhizlabsBean implements two interfaces, and none of them is specified as a local or remote business interface, Whizlabs1 and Whizlabs2 are just normal interfaces.

The correct answer is: The bean does not expose any client view

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

QUESTION 29

NOT ANSWERED

[Ask our Experts](#)

MARK FOR REVIEW

Given two bean class declarations:

```
@Stateless  
public class BeanA implements Whizlabs { ... }
```

```
@Stateful  
@LocalBean  
public class BeanB implements Whizlabs { ... }
```

The Whizlabs interface is not annotated with either @Local or @Remote, and there is no associated deployment descriptor. Which of the above beans expose a client view through a business interface?

Please select :

- A. BeanA only
- B. BeanB only
- C. Both BeanA and BeanB
- D. Neither BeanA nor BeanB

Your answer is incorrect.

Answer: A

Explanation:

In case of BeanA, the bean class implements an interface, and does not expose any other client view. This interface is considered a local business interface and the bean expose only a local client view through this interface.

In case of BeanB, the bean class also implements an interface, but the @LocalBean annotation designates that a session bean exposes a no-interface view. As a result, BeanB exposes only a no-interface client view.

The correct answer is: BeanA only

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

QUESTION 30

NOT ANSWERED

[Ask our Experts](#)

MARK FOR REVIEW

Which type of session beans can have business methods decorated with the @AccessTimeout annotation?

Please select :

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



Your answer is incorrect.

Answer: C

Explanation:

The Java EE 6 API documentation declares that the @AccessTimeout annotation specifies the amount of time in a given time unit that a concurrent access attempt should block before timing out. This annotation may be applied to a stateful session bean or to a singleton session bean that uses container managed concurrency.

References:

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

The correct answer is: Singleton and stateful

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

QUESTION 31

NOT ANSWERED

[Ask our Experts](#)

MARK FOR REVIEW

Given a local client that invokes a business method of a session bean with container-managed transaction demarcation, and receive an application exception as a result of the invocation. The client is another enterprise bean with bean-managed transaction demarcation. Which of the following is a correct way for the client to check whether the transaction has been marked for rollback?

Please select :

- A. Use the EJBContext.getRollbackOnly method
- B. Use the UserTransaction.getRollbackOnly method
- C. Use the UserTransaction.getStatus method
- D. None of the above

D. None of the above

Your answer is incorrect.

Answer: C

Explanation:

The EJBContext.getRollbackOnly method is used for bean with container-managed transaction demarcation only, thus, option A is incorrect. Note that the transaction demarcation of the target bean has nothing to do over here.

The UserTransaction interface does not define a method named getRollbackOnly, hence option B is incorrect. The method given in option C is the correct way to implement the task.

The correct answer is: Use the UserTransaction.getStatus method

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

QUESTION 32

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Which of the following methods can be used in an enterprise bean with container-managed transaction demarcation to commit the transaction associated with the current thread?

Please select :

- A. java.sql.Connection.commit
- B. javax.jms.Session.commit
- C. javax.transaction.UserTransaction.commit
- D. javax.persistence.EntityTransaction.commit
- E. None of the above

Your answer is incorrect.

Answer: E

Explanation:

As per the EJB 3.1 Specification (subsection 13.3.4), an enterprise bean with container-managed transaction demarcation must not use any resource-manager specific transaction management methods that would interfere with the container's demarcation of transaction boundaries. For example, the enterprise bean methods must not use the following methods of the java.sql.Connection interface: commit, setAutoCommit, and rollback; or the following methods of the javax.jms.Session interface: commit and rollback.

The correct answer is: None of the above

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

QUESTION 33

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Which THREE of the following transaction attributes can be used on timeout callback methods of enterprise 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, C and E

Explanation:

As per the EJB 3.1 Specification (subsection 13.3.7), for an enterprise bean's timeout callback methods, only the REQUIRED, REQUIRES_NEW and NOT_SUPPORTED transaction attributes may be used.

The correct answers are: Required, RequiresNew, NotSupported

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

QUESTION 34

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Given two class declarations:

```
@TransactionAttribute(NEVER)
public class SuperClass {
    public void methodA () { ... }
    public void methodB () { ... }
    // other declarations
}
@Stateless
public class SubClass extends SuperClass {
    public void methodA () { ... }
    // other declarations
```

}

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

Please select :

- A. methodA: Never
methodB: Never
- B. methodA: Required
methodB: Never
- C. methodA: Required
methodB: Required
- D. methodA: Never
methodB: Required

Your answer is incorrect.

Answer: B

Explanation:

Both methodA and methodB are declared in SuperClass with transaction attribute Never, but methodA is overridden in SubClass with attribute Required, the default value when no attribute is explicitly specified.

The correct answer is: methodA: Required

methodB: Never

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

[QUESTION 35](#)

[NOT ANSWERED](#)

[▼ MARK FOR REVIEW](#)

[Ask our Experts](#)

Which of the following is NOT a valid fragment in the deployment descriptor?

Please select :

- A. <container-transaction>
<method>
<ejb-name>Bean1</ejb-name>
<method-name>methodA</method-name>
</method>
<method>
<ejb-name>Bean2</ejb-name>
<method-name>methodB</method-name>
</method>
<trans-attribute>Required</trans-attribute>
</container-transaction>
- B. <container-transaction>
<method>
<ejb-name>Bean1</ejb-name>
<method-name>*</method-name>
</method>
<method>
<ejb-name>Bean1</ejb-name>
<method-name>methodA</method-name>
</method>
<trans-attribute>Required</trans-attribute>
</container-transaction>
- C. <container-transaction>
<method>
<ejb-name>Bean1</ejb-name>
<method-name>methodA</method-name>
</method>
</container-transaction>
<container-transaction>
<method>
<ejb-name>Bean2</ejb-name>
<method-name>methodB</method-name>
</method>
<trans-attribute>Required</trans-attribute>
</container-transaction>
- D. None of the above

Your answer is incorrect.

Answer: A

Explanation:

The fragment in option A is invalid as the EJB 3.1 Specification (subsection 13.3.7.2.1) requires that all the methods specified in a single container-transaction element be methods of the same enterprise bean.

The correct answer is: <container-transaction>

```
<method>
<ejb-name>Bean1</ejb-name>
<method-name>methodA</method-name>
</method>
<method>
```

```

<ejb-name>Bean2</ejb-name>
<method-name>methodB</method-name>
</method>
<trans-attribute>Required</trans-attribute>
</container-transaction>

```

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

QUESTION 36

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Given a method invocation chain among three methods in stateless session beans with container-managed transaction demarcation:
methodA (T1) --> methodB (No) --> 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. Never and RequiresNew
- B. NotSupported and Required
- C. NotSupported and Supports
- D. Never and Mandatory

Your answer is incorrect.

Answer: B

Explanation:

If the transaction attribute of methodB is Never, a RemoteException will be thrown. Thus, options A and D are incorrect.

If the transaction attribute of methodC is Supports, it will be executed outside a transaction. So option C is incorrect as well.

The following are descriptions of the NotSupported and Required transaction attributes:

- NotSupported: If the client is running within a transaction and invokes the enterprise bean's method, the container suspends the client's transaction before invoking the method. After the method has completed, the container resumes the client's transaction. If the client is not associated with a transaction, the container does not start a new transaction before running the method.
- Required: If the client is running within a transaction and invokes the enterprise bean's method, the method executes within the client's transaction. If the client is not associated with a transaction, the container starts a new transaction before running the method.

References:

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

The correct answer is: NotSupported and Required

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

QUESTION 37

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Given four methods in stateless session beans with container-managed transaction demarcation. The transaction attributes of these methods are shown below:
methodA: NotSupported
methodB: Supports
methodC: RequiresNew
methodD: Required

Which of the following invocation chains is correct if the client calling methodA runs within transaction T1?

Note: the arrow (--) represents a method invocation; while T1, T2, T3 denotes transaction 1, transaction 2, transaction 3 and No means the associated method runs outside a transaction.

Please select :

- A. methodA (No) --> methodB (T1) --> methodC (T2) --> methodD (T3)
- B. methodA (T1) --> methodB (T1) --> methodC (T2) --> methodD (T3)
- C. methodA (No) --> methodB (T2) --> methodC (T3) --> methodD (T3)
- D. methodA (No) --> methodB (No) --> methodC (T2) --> methodD (T2)

Your answer is incorrect.

Answer: D

Explanation:

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

The transaction attribute of methodB is Supports, so it must not run within a transaction if the caller does not. As a result, options A and C are incorrect.

References:

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

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

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

QUESTION 38

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Which TWO of the following are EJB components?

Please select :

- A. Entity beans
- B. Managed beans
- C. JMS beans
- D. Session beans

Your answer is incorrect.

Answer: A and D

Explanation:

Entity beans represent the persistent data stored in the database, deprecated as of EJB 3.0. However, these beans are still supported in EJB 3.1, they are thus EJB components.

Session beans are managed by the EJB container, encapsulating business logic that can be invoked programmatically by clients over local, remote, or web service client views.

Managed beans represent a generalization of the managed beans specified by JavaServer Faces technology. These beans are not managed by EJB containers. So, option B is incorrect.

There are no such things like JMS beans. Therefore, option C is incorrect as well.

References:

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

The correct answers are: Entity beans, Session beans

Submit your Feedback/Queries to our Experts

QUESTION 39

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Which of the following statements is correct about EJB Lite?

Please select :

- A. EJB Lite is an additional set of EJB core features; it is recommended, but not required, that Java EE containers implement this supplementary feature set
- B. EJB Lite is a feature set that is newly added to the EJB 3.1 specification; it does not exist in the EJB 3.0 specification
- C. EJB Lite applications can be deployed on any Java EE product that implements Enterprise JavaBeans technology
- D. EJB Lite applications are operable in EJB Lite compliant containers, but may not be in full EJB API compliant ones

Your answer is incorrect.

Answer: C

Explanation:

As per the EJB 3.1 Specification (subsection 21.1), EJB Lite is a proper subset of the full EJB 3.1 API that includes a small, powerful selection of EJB features suitable for writing portable transactional business logic. The definition of EJB 3.1 Lite gives vendors an option to implement only a portable subset of the EJB API within their product. An EJB 3.1 Lite application is merely an EJB application whose EJB API usage falls within the EJB Lite subset. There are no special APIs defined only for EJB 3.1 Lite. Therefore, any EJB 3.1 Lite application can be deployed on any Java EE product that implements Enterprise JavaBeans technology, whether that product supports EJB 3.1 Lite or the full EJB API.

The correct answer is: EJB Lite applications can be deployed on any Java EE product that implements Enterprise JavaBeans technology

Submit your Feedback/Queries to our Experts

QUESTION 40

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

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

Please select :

- A. Must not use the AWT functionality to output information to a display, or to input from a keyboard
- B. Must not listen or accept connections on a socket
- C. Must not attempt to use the Reflection API to access information that the security rules of the Java programming language make unavailable
- D. None of the above

Your answer is incorrect.

Answer: D

Explanation:

The EJB 3.1 Specification (subsection 21.2.2) declares:

- An enterprise bean must not use the AWT functionality to attempt to output information to a display, or to input information from a keyboard. The reason is that most servers do not allow direct interaction between an application program and a keyboard/display attached to the server system.
- An enterprise bean must not attempt to listen on a socket, accept connections on a socket, or use a socket for multicast since allowing the instance to become a network server would conflict with the basic function of the enterprise bean - to serve the EJB clients.
- The enterprise bean must not attempt to query a class to obtain information about the declared members that are not otherwise accessible to the enterprise bean because of the security rules of the Java language. The enterprise bean must not attempt to use the Reflection API to access information that the security rules of the Java programming language make unavailable. Allowing the enterprise bean to access information about other classes and to access the classes in a manner that is normally disallowed by the Java programming language could compromise

security.

The correct answer is: None of the above

Submit your Feedback/Queries to our Experts

QUESTION 41

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Which of the following APIs is NOT defined for the Java EE platform?

Please select :

- A. JCA
- B. JAX-RS
- C. JavaMail
- D. JNDI
- E. JPA

Your answer is incorrect.

Answer: D

Explanation:

The Java Naming and Directory Interface API is included in the Java SE platform. All the others are Java EE APIs and must be provided by EJB containers.

References:

<http://docs.oracle.com/javaee/6/tutorial/doc/bnacj.html>

The correct answer is: JNDI

Submit your Feedback/Queries to our Experts

QUESTION 42

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Which of the following methods defined in the java.lang.System class can be invoked by an enterprise bean method?

Please select :

- A. exit(int status)
- B. setErr(PrintStream err)
- C. setProperty(String key, String value)
- D. setSecurityManager(SecurityManager sm)

Your answer is incorrect.

Answer: C

Explanation:

As per the Java documentation, an enterprise bean must not attempt to set security manager; create a new security manager; stop the JVM; or change the input, output, and error streams. Therefore, options A, B and D are incorrect.

The correct answer is: setProperty(String key, String value)

Submit your Feedback/Queries to our Experts

QUESTION 43

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Which THREE of the following objects must all EJB containers make available to enterprise bean instances through dependency injection and JNDI lookup?

Please select :

- A. UserTransaction objects
- B. EJBContext objects
- C. TimerService objects

- D. The local business interfaces of other enterprise beans
- E. The remote business interfaces of other enterprise beans

Your answer is incorrect.

Answer: A, B and D

Explanation:

The following is an extract taken from the EJB 3.1 Specification (subsection 21.3.2):

All EJB containers must make available at least the following objects in the namespace:

- The local business interfaces of other enterprise beans
- References to the no-interface view of other enterprise beans
- UserTransaction objects
- EJBContext objects
- The resource factories used by the enterprise beans
- The entity managers and entity manager factories used by the enterprise beans

An EJB 3.1 container within an implementation of the Full Java EE Platform must make available the following objects in the namespace:

- The remote business interfaces of other enterprise beans
- The web service interfaces used by the enterprise beans
- The home interfaces of other enterprise beans
- ORB objects
- TimerService objects

The correct answers are: UserTransaction objects, EJBContext objects, The local business interfaces of other enterprise beans

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

QUESTION 44

NOT ANSWERED

 **MARK FOR REVIEW**

[Ask our Experts](#)

When packaging an enterprise bean in an EJB-JAR file, which class files of the following related types must be put into the same JAR file as this bean?

Please select :

- A. Its business interfaces
- B. Its web service endpoint interfaces
- C. Its component interfaces
- D. Its interceptor classes
- E. None of the above

Your answer is incorrect.

Answer: E

Explanation:

The EJB-JAR file or WAR file must contain, either by inclusion or by reference, the class files for all the classes and interfaces listed above. Thus option E is the correct answer.

Note: As per the Java EJB 3.1 Specification (subsection 20.3), a JAR file contains a second file "by reference" if the second file is named in the Class-Path attribute in the Manifest file of the referencing JAR file or is contained (either by inclusion or by reference) in another JAR file that is named in the Class-Path attribute in the Manifest file of the referencing JAR file.

The correct answer is: None of the above

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

QUESTION 45

NOT ANSWERED

 **MARK FOR REVIEW**

[Ask our Experts](#)

Given two EAR files deployed in a Java EE container:

app1.ear:

META-INF/application.xml

ejb_client.jar

ejb1.jar

ejb2.jar

app2.ear:

META-INF/application.xml

ejb3.jar

The ejb_client.jar file contains client views (business interfaces or home and component interfaces) of enterprise beans enclosed within ejb1.jar, ejb2.jar and ejb3.jar. The Manifest file of ejb2.jar and ejb3.jar contains the entry: Class-Path: ejb_client.jar. Which beans can use those client views?

Please select :

- A. Beans in ejb2.jar only
- B. Beans in ejb1.jar and ejb2.jar
- C. Beans in ejb2.jar and ejb3.jar
- D. Beans in ejb1.jar, ejb2.jar and ejb3.jar

Your answer is incorrect.

Answer: A

Explanation:

As per the EJB 3.1 Specification (subsection 20.6), the EJB specification does not specify whether an EJB-JAR file or WAR file should include by copy or by reference the classes that are in an EJB-CLIENT JAR file, but they must be included either one way or the other. If the by-copy approach is used, the producer simply includes all the class files in the EJB-CLIENT JAR file also in the EJB-JAR file or WAR file. If the by-reference approach is used, the EJB-JAR file or WAR file producer does not duplicate the content of the EJB-CLIENT JAR file in the EJB-JAR file or WAR file, but instead uses a Manifest Class-Path entry in the EJB-JAR file or WAR file to specify that the EJB-JAR file or WAR file depends on the EJB-CLIENT JAR at runtime. Therefore, the client views are not visible in ejb1.jar

Enterprise beans in ejb3.jar do not have access to the client views packaged in ejb_client.jar since those are included in another EAR file.

The correct answer is: Beans in ejb2.jar only

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

QUESTION 46

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Which TWO of the following are suitable locations to place the EJB deployment descriptor?

Please select :

- A. The META-INF directory of an EAR file
- B. The WEB-INF directory of a WAR file
- C. The META-INF directory of a WAR file
- D. The META-INF directory of a JAR file

Your answer is incorrect.

Answer: B and D

Explanation:

An enterprise bean can be packaged within an EJB-JAR or a WAR file, of which the META-INF or WEB-INF directory, respectively, is the location to put the EJB deployment descriptor in.

The descriptor cannot be placed directly in an EAR file, thus option A is incorrect. Meanwhile, META-INF is not a standard directory of a WAR file, option C is also incorrect, then.

The correct answers are: The WEB-INF directory of a WAR file, The META-INF directory of a JAR file

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

QUESTION 47

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Which of the following statements is correct about packaging an enterprise bean?

Please select :

- A. The enterprise bean class file must be packaged in an EJB-JAR or in a WAR file
- B. In order to be usable within a web application, the enterprise bean class file must be put in the WEB-INF/classes directory of the WAR file
- C. The outermost archive file of the enterprise bean must be an EAR file
- D. None of the above

Your answer is incorrect.

Answer: A

Explanation:

As per the Java documentation, an EJB-JAR file is a Java EE module specifically designed for the exclusive packaging of enterprise beans. Enterprise beans can also be packaged within a web application module (WAR). As such, option A is the correct answer.

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. Therefore, option B is incorrect.

Both EJB-JAR and WAR files can be deployed without being wrapped inside an EAR file. Hence, option C is incorrect.

References:

<http://docs.oracle.com/javaee/6/tutorial/doc/gipio.html>

The correct answer is: The enterprise bean class file must be packaged in an EJB-JAR or in a WAR file

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

QUESTION 48

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Which of the following statements can be used to create an embeddable EJB container?

Please select :

- A. EJBContainer container = new EJBContainer();
- B. EJBContainer container = EJBContainer.createEJBContainer();
- C. EJBContainer container = EJBContainerProvider.createEJBContainer();
- D. FIRContainer container = FIRContainerFactory.newFIRContainer();

Your answer is incorrect.

Answer: B

Explanation:

EJBContainer is an abstract class and cannot be instantiated using the new operator. Hence, option A is incorrect.

Since the EJBContainerProvider.createEJBContainer has no overloaded flavor that takes no argument, option C is incorrect.

There is no such a class like EJBContainerFactory, thus option D is incorrect.

References:

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

The correct answer is: EJBContainer container = EJBContainer.createEJBContainer();

Submit your Feedback/Queries to our Experts

QUESTION 49

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Given the declarations of several exception classes:

```
@ApplicationException(rollback = true)
public class ExceptionA extends RuntimeException { ... }

public class ExceptionB extends ExceptionA { ... }

@ApplicationException(inherited = false, rollback = false)
public class ExceptionC extends ExceptionB { ... }

public class ExceptionD extends ExceptionC { ... }
```

Which of the exceptions shown above are application exceptions that does not cause transaction rollback, provided there is no associated deployment descriptor?

Please select :

- A. ExceptionC only
- B. ExceptionB and ExceptionC
- C. ExceptionC and ExceptionD
- D. ExceptionB, ExceptionC and ExceptionD

Your answer is incorrect.

Answer: A

Explanation:

ExceptionB is an application exception causing transaction rollback as inherited from ExceptionA.

ExceptionD is not an application exception since it is not decorated with the @ApplicationException annotation and the inherited element of this annotation on the superclass, ExceptionC, is set to false.

ExceptionA and ExceptionC are both application exceptions, but only ExceptionA causes transaction rollback, as explicitly specified.

The correct answer is: ExceptionC only

Submit your Feedback/Queries to our Experts

QUESTION 50

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Given a session bean (with container-managed transaction demarcation) business method running within a local client's transaction.

At some point the bean method throws an application exception (not specified as causing rollback). Which of the following actions the container will take?

Please select :

- A. Marks the transaction for rollback
- B. Re-throws the exception to the caller
- C. Throws EJBTransactionRolledbackException to the caller
- D. Discards the bean instance

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 re-throws the exception, and marks the transaction for rollback only if the exception is specified as causing rollback.

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

The correct answer is: Re-throws the exception to the caller

Submit your Feedback/Queries to our Experts

QUESTION 51

NOT ANSWERED

MARK FOR REVIEW

Ask our Experts

Given a session bean business method:

```
public void myMethod() {
    // method body
```

}

Which of the following exceptions can be thrown from within the above method?

Please select :

- A. CreateException
- B. EJBException
- C. FinderException
- D. RemoveException

Your answer is incorrect.

Answer: B

Explanation:

CreateException, RemoveException and FinderException are standard application exceptions for entities. These exceptions are all checked exceptions and must be listed in the throws clause of a method if it is expected to be thrown from the method body.

EJBException is a subclass of RuntimeException, hence it does not need to be specified in the method declaration.

The correct answer is: EJBException

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

QUESTION 52

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Which TWO of the following are system exceptions?

Please select :

- A. javax.ejb.CreateException
- B. java.io.IOException
- C. javax.ejb.EJBTransactionRequiredException
- D. javax.transaction.TransactionRequiredException

Your answer is incorrect.

Answer: C and D

javax.ejb.EJBTransactionRequiredException is a subclass of RuntimeException, while javax.transaction.TransactionRequiredException is a subclass of RemoteException. Thus, they are both system exceptions.

Note: System exceptions are RuntimeException and RemoteException and their subclasses.

The correct answers are: javax.ejb.EJBTransactionRequiredException, javax.transaction.TransactionRequiredException

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

QUESTION 53

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Which of the following cannot be annotated as @ApplicationException?

Please select :

- A. Subclasses of RuntimeException
- B. Subclasses of RemoteException
- C. Subclasses of IOException
- D. Subclasses of FinderException

Your answer is incorrect.

Answer: B

Explanation:

As per the EJB 3.1 Specification (subsection 14.1.1), an application exception may be a subclass (direct or indirect) of java.lang.Exception (i.e., a "checked exception"), or an application exception class may be defined as a subclass of the java.lang.RuntimeException (an "unchecked exception"). An application exception may not be a subclass of the java.rmi.RemoteException. The java.rmi.RemoteException and its subclasses are reserved for system exceptions. The javax.ejb.CreateException, javax.ejb.RemoveException, javax.ejb.FinderException, and subclasses thereof are considered to be application exceptions.

The correct answer is: Subclasses of RemoteException

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

QUESTION 54

NOT ANSWERED

MARK FOR REVIEW

[Ask our Experts](#)

Which of the following statements is correct about singleton session beans?

Please select :

- A. An application has only one instance of a singleton bean class no matter how many JVMs over which the container is distributed
- B. The state of a singleton instance must survive container shutdown or crash
- C. Singleton session beans support concurrent access
- D. A singleton session bean class may implement the SessionSynchronization interface to be notified of transaction boundaries

Your answer is incorrect.

Your answer is incorrect.

Answer: C

Explanation:

The following is an excerpt from the EJB 3.1 Specification (subsection 4.8):

A Singleton session bean is a session bean component that is instantiated once per application. In cases where the container is distributed over many virtual machines, each application will have one bean instance of the Singleton for each JVM.

Once instantiated, a Singleton session bean instance lives for the duration of the application in which it is created. It maintains its state between client invocations but that state is not required to survive container shutdown or crash.

A Singleton session bean is intended to be shared and supports concurrent access.

A Singleton session bean must not implement the javax.ejb.SessionSynchronization interface or use the session synchronization annotations.

The correct answer is: Singleton session beans support concurrent access

Submit your Feedback/Queries to our Experts

QUESTION 55

NOT ANSWERED

Ask our Experts

Which of the following annotations should be specified on a singleton session bean, so as to instruct the container to initialize the bean instance during the application startup sequence?

Please select :

- A. @Start
- B. @Startup
- C. @Startup(true)
- D. @Begin(true)
- E. None of the above is needed, as a singleton bean is eagerly initialized by default

Your answer is incorrect.

Answer: B

Explanation:

The @Startup annotation marks a singleton bean for eager initialization during the application startup sequence. This annotation contains no elements.

References:

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

The correct answer is:@Startup

Submit your Feedback/Queries to our Experts

QUESTION 56

NOT ANSWERED

Ask our Experts

Which of the following methods of a singleton session bean causes the bean instance to be discarded when throwing a system exception?

Please select :

- A. A business method
- B. A @PostConstruct method
- C. A @PreDestroy method
- D. A timeout method
- E. None of the above

Your answer is incorrect.

Answer: B

Explanation:

As per the EJB 3.1 Specification (subsection 4.8.4), errors occurring during Singleton initialization are considered fatal and must result in the discarding of the Singleton instance. Possible initialization errors include injection failure, or a system exception thrown from a PostConstruct method.

Once the Singleton bean instance has successfully initialized, the same bean instance must remain active until application shutdown. Unlike instances of other component types, system exceptions thrown from business methods or callbacks of a Singleton do not result in the destruction of the Singleton instance.

The correct answer is: A @PostConstruct method

Submit your Feedback/Queries to our Experts

QUESTION 57

NOT ANSWERED

Ask our Experts

In which of the following cases an enterprise bean should NOT throw a system exception?

A. Failure to obtain a database connection

Please select :

- A. Failure to obtain a database connection
- B. A JNDI exception occurs
- C. An application exception occurs

- C. A JVM error occurs
- D. An unexpected unchecked exception occurs
- E. The input arguments to a business method are invalid

Your answer is incorrect.

Answer: E

Explanation:

A system exception should be thrown if an exception or error is unexpected, or the exception is expected but the EJB Provider does not know how to recover from it. The situations in options A, B, C and D are out of the client's control, thus system exceptions should be thrown.

When input arguments to a business method are invalid, the client should be given a chance to recover the transaction. Therefore, an application exception should be thrown and option E is the correct answer.

The correct answer is: The input arguments to a business method are invalid

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

QUESTION 58

NOT ANSWERED

 **MARK FOR REVIEW**

[Ask our Experts](#)

Which of the following statements is correct about Timer objects?

Please select :

- A. They can be created for session beans and message-driven beans
- B. They can be created for session beans and entity beans
- C. They cannot be created for stateful session beans
- D. They cannot be created for singleton session beans
- E. They can be created for session bean only

Your answer is incorrect.

Answer: C

Explanation:

As per the EJB 3.1 Specification (subsection 18.2), timers can be created for stateless session beans, singleton session beans, message-driven beans, and 2.1 entity beans. Timers cannot be created for stateful session beans.

The correct answer is: They cannot be created for stateful session beans

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

QUESTION 59

NOT ANSWERED

 **MARK FOR REVIEW**

[Ask our Experts](#)

Which of the following statements is correct about the timeout callbacks for a particular enterprise bean when there is a timer created for this bean using the TimerService API?

Please select :

- A. There must be exactly one method annotated as @Timeout in the bean
- B. If the bean implements the TimedObject interface, @Timeout must not be specified on any method of the bean
- C. The timeout method may be specified on the bean class or on a superclass
- D. None of the above

Your answer is incorrect.

Answer: C

Explanation:

As per the EJB 3.1 Specification (subsection 18.2.5.1), 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 javax.ejb.TimedObject interface. This interface has a single method, ejbTimeout. If the bean implements the TimedObject interface, the Timeout annotation or timeout-method deployment descriptor element can only be used to specify the ejbTimeout method. A bean can have at most one timeout method for handling programmatic timers. This method may be specified on the bean class or on a superclass.

The correct answer is: The timeout method may be specified on the bean class or on a superclass

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

QUESTION 60

NOT ANSWERED

 **MARK FOR REVIEW**

[Ask our Experts](#)

Given two class declarations:

```
@Stateless
public class Whizlabs implements TimedObject {
    @Resource
    private TimerService timerService;
    public void createTimer() {
        Timer timer = timerService.createTimer(1000, new Info());
    }
    public void ejbTimeout(Timer timer) {
        System.out.println("Timeout");
    }
}
```

```
        }  
    }  
    public class Info {  
        private String description;  
        // getter and setter  
    }
```

What happens when the createTimer is invoked, given no timeout-related elements are existent in the deployment descriptor?

Please select :

- A. The string "Timeout" gets printed on the console after 1 second
- B. The string "Timeout" gets printed on the console after 1000 seconds
- C. An error is produced as there is no method annotated with @Timeout
- D. An error is produced as the Info class is not serializable

Your answer is incorrect.

Answer: D

Explanation:

The second argument to the TimerService.createTimer is application information to be delivered along with the timer expiration notification. This information must be represented by a serializable object, thus option D is correct.

Note that the given session bean implements the TimedObject interface, so the @Timeout annotation or timeout-method deployment descriptor element is unnecessary.

The correct answer is: An error is produced as the Info class is not serializable

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.