# 1Z0-895<sup>Q&As</sup>

Java EE 6 Enterprise JavaBeans Developer Certified Expert

## Pass Oracle 1Z0-895 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

**https://www.passapply.com/1Z0-895.html**

### 100% Passing Guarantee
### 100% Money Back Assurance

Following Questions and Answers are all new published by Oracle
Official Exam Center

⚙ **Instant Download** After Purchase

⚙ **100% Money Back** Guarantee

⚙ **365 Days** Free Update

⚙ **800,000+** Satisfied Customers

**QUESTION 1**

Which two statements are true? (Choose two.)

A. Typically, remotely accessible objects should be coarse-grained.

B. If a client accesses an enterprise bean locally such access must be mediated by the EJB container.

C. A given enterprise bean\\\'s transaction information is immutable because it is deployed across various containers.

D. If a container provides services NOT required by the EJB specification, then that container is NOT considered to be an EJB container.

E. An enterprise bean\\\'s transaction Information can be accessed by external tools only if the information is contained in an XML deployment descriptor.

Correct Answer: BD

D: An EJB container is nothing but the program that runs on the server and implements the EJB specifications. EJB container provides special type of the environment suitable for running the enterprise components. Enterprise beans are used in distributed applications that typically contains the business logic.

Incorrect answers:

A: Remotely accessible objects should be fine-grained

**QUESTION 2**

A developer creates a stateless session bean, EmployeeServiceBean, and its interface, EmployeeService. The session bean uses two annotated entity classes, Employee.class and Department.class.

Which two package options can the developer use when creating a deployable EAR? The proposed directory structure is listed for each option. (Choose two)

A. Emp.ear emp-ejb.jar META-INF/persistence.xml EmployeeService.class EmployeeServiceBean.class lib/emp-classes.jar Employee.class Department.class

B. Emp.ear META-INF/orm.xml EmployeeService.class EmployeeServiceBean.class Employee.class Department.class

C. Emp.ear emp-ejb.jar META-INF/persistence.cml Employee.class Department.class EmployeeService.class EmployeeServiceBean.class

D. Emp.ear emp-ejb.jar persistence.xml Employee.class Department.class EmployeeService.class EmployeeServiceBean.class

Correct Answer: AC

**QUESTION 3**

An enterprise bean has security permissions set up using declarative security features. Under which two conditions can a client be guaranteed to have permission to invoke a business method on the enterprise bean? (Choose two.)

A. The Application Assembler has marked the enterprise bean method as unchecked.

B. The client\\'s principal has been assigned a security role with permission to invoke the method.

C. The Application Assembler has set the security-identity deployment descriptor to run-as.

D. The Application Assembler has mapped all security role references using the role-link element.

Correct Answer: AB

**QUESTION 4**

Given Singleton bean FooEJB:



FooEJB is packaged as the only bean in an ejb-jar and deployed to a server instance. Which represents the output generated from FooEJB after the deployment has completed?

A. Init

B. foo

C. Init foo

D.

E. a or d

Correct Answer: A

Note:

* The PostConstruct annotation is used on a method that needs to be executed after dependency injection is done to perform any initialization. This method MUST be invoked before the class is put into service. This annotation MUST be supported on all classes that support dependency injection. The method annotated with PostConstruct MUST be invoked even if the class does not request any resources to be injected. Only one method can be annotated with this annotation. The method on which the PostConstruct annotation is applied MUST fulfill all of the following criteria - - The method MUST NOT have any parameters except in the case of EJB interceptors in which case it takes an InvocationC ontext object as defined by the EJB specification. - The return type of the method MUST be void. - The method MUST NOT throw a checked exception.

- The method on which PostConstruct is applied MAY be public, protected, package private or private. - The method MUST NOT be static except for the application client. - The method MAY be final. - If the method throws an unchecked exception the class MUST NOT be put into service except in the case of EJBs where the EJB can handle exceptions and even recover from them.

* PostConstruct methods are invoked on the newly constructed instance, after any dependency injection has been performed by the container and before the first business method is invoked on the bean.

**QUESTION 5**

MyMsg is a JMS message-driven bean with container-managed transaction demarcation. FooBean is an EJB 3.x stateless session bean that sends message to the JMS destination with MyMsgBean is associated.

MyMsgBean\\'s message listener method has transaction attribute REQUIRED, and is defined as follows:

10.

 public class MyMsgBean implements javax.jms.messageListener {

11.

 public void onMessage(javax.jms.Message message) {

12.

 / / do some work not shown here

13.

 thrown new RuntimeException("unexpected error . . . ");

14.

 }

Which statement is true about the result of message processing?

A. FooBean receives javax.ejb.EJBException.

B. The container discards the MyMsgBean bean instance.

C. FooBean receives the original RuntimeException thrown from the message listener method.

D. The container does NOT roll back the transaction, and FooBean can continue the transaction.

Correct Answer: B

Reference: Enum TransactionAttributeType

**QUESTION 6**

An ejb-jar also contains three interceptor classes: AInt, BInt, CInt. Each interceptor class defines an AroundInvoke method called intercept.

The ejb-jar also contains a stateless session bean FooBean with a local business interface Foo that declares a method foo ():

10.

 @Stateless

11.

 @Intercaptors ({CInt.class, BInt.class})

12.

 public class FooBean implements Foo {

13.

14.

 public void foo () {}

15.

16.

 }

The ejb-jar contains a META-INF/ejb-jar.xml file with an section:


FooBean


com.acme.AInt



What is the interceptor order when the business methodfoo() is invoked?

A. AInt

B. AInt, CInt, BInt

C. CInt, BInt, AInt

D. AInt, BInt, CInt

Correct Answer: B

With the interceptor-order clauses Aint will be first in the order of interceptors. Within each group (default, class, method) the order of the interceptors are from left to right as defined in the @Interceptors annotation, and then the xml interceptors. In this scenario, with the @Intercaptors ({CInt.class, BInt.class}) line, the ordering continues with CInt and BInt. Note 1: By default the ordering of interceptors when invoking a method are

* External interceptors ** Default interceptors, if present ** Class interceptors, if present ** Method interceptors, if present Bean class interceptor method *

Note 2: You can override the default sort order of the external interceptors by specifiying an interceptor- binding with an interceptor-order specifying the order of the interceptors

Reference: EJB Interceptors

http://docs.jboss.org/ejb3/app-server/tutorial/interceptor/interceptor.html

---

**QUESTION 7**

A stateless session bean FooBean implements an asynchronous business method foo() on its bean class:

@Asynchronous

public void foo() ( ... )

The asynchronous business method is exposed through a remote business interface FooRemote. A caller acquires an EJB reference to this bean and invokes it as follows:

100. fooRemoteRef.foo();

Which exception can result from the invocation on line 100?

A. java.rmi.RemoteException

B. java.util.concurrent.ExecutionException

C. javax.ejb.EJBException

D. java.lang.IllegalArgumentException

Correct Answer: A

Note:

*

 RemoteRef represents the handle for a remote object. A RemoteStub uses a remote reference to carry out a remote method invocation to a remote object.

*

 invoke

public Object invoke(Remote obj,

Method method,

Object[] params,

long opnum)

throws Exception

Invoke a method. This form of delegating method invocation to the reference allows the reference to take care of setting up the connection to the remote host, marshaling some representation for the method and parameters, then

communicating the method invocation to the remote host. This method either returns the result of a method invocation on the remote object which resides on the remote host or throws a RemoteException if the call failed or an application-level

exception if the remote invocation throws an exception.

Parameters:

**QUESTION 8**

Given the stateful session bean:

```
10.  @Stateful
11.  public class VideoBean implements video {
12.      public void methodA() {}
13.
14.      @TransactionAttribute(TransactionAttributeType.SUPPORTS)
15.      public void methodB() {}
16.
17.      public void methodC() {
18.      @TransactionAttribute(TransactionAttributeType.REQUIRED)
19.
20.      public void methodD() {}
21.  }
```

Assuming no other transaction-related metadata, which is true?

A. methodB and methodC have transaction attribute SUPPORTS, which methodD has transaction attribute REQUIRED.

B. methodA and methodC have transaction attribute REQUIRES_NEW, while methodB has transaction attribute SUPPORTS.

C. methodC, methodD, and methodA have transaction attribute REQUIRED, and methodB has transaction attribute SUPPORTS.

D. methodB has transaction attribute SUPPORTS, methodD has transaction attribute REQUIRED, and methodA and methodC have transaction attribute REQUIRES_NEW.

Correct Answer: C

**QUESTION 9**

Which two statements are correct about stateless session beans? (Choose two.)

A. The bean class may declare instance variables.

B. The lifetime of the bean instance is controlled by the client.

C. The container may use the same bean instance to handle multiple business method invocations at the same time.

D. The container may use the same bean instance to handle business method invocations requested by different clients, but not concurrently.

Correct Answer: AC

* A: Stateless session beans are EJB\\'s version of the traditional transaction processing applications, which are executed using a procedure call. The procedure executes from beginning to end and then returns the result. Once the procedure is done, nothing about the data that was manipulated or the details of the request are remembered. There is no state.

These restrictions don\\'t mean that a stateless session bean can\\'t have instance variables and therefore some kind of internal state. There\\'s nothing that prevents you from keeping a variable that tracks the number of times a bean has been called or that tracks data for debugging. An instance variable can even hold a reference to a live resource like a URL connection for writing debugging data, verifying credit cards, or anything else that might be useful.

A stateless session bean is relatively easy to develop and also very efficient.

C:

Stateless session beans require few server resources because they are neither persistent nor dedicated to one client. Because they aren\\'t dedicated to one client, many EJB objects can use just a few instances of a stateless bean. A

stateless session bean does not maintain conversational state relative to the EJB object it is servicing, so it can be swapped freely between EJB objects. As soon as a stateless instance services a method invocation, it can be swapped to

another EJB object immediately. Because there is no conversational state, a stateless session bean doesn\\'t require passivation or activation, further reducing the overhead of swapping. In short, they are lightweight and fast!

* The Lifecycle of a Stateless Session Bean Because a stateless session bean is never passivated, its lifecycle has only two stages: nonexistent and ready for the invocation of business methods. The EJB container typically creates and maintains a pool of stateless session beans, beginning the stateless session bean\\'s lifecycle. The container performs any dependency injection and then invokes the method annotated @PostConstruct, if it exists. The bean is now ready to have its business methods invoked by a client.

At the end of the lifecycle, the EJB container calls the method annotated @PreDestroy, if it exists (not B). The bean\\'s instance is then ready for garbage collection.

**QUESTION 10**

Which statement is true about both stateful session beans and stateless session beans? (Select Two)

A. Bean instance are NOT required to survive container crashes.

B. Any bean instance must be able to handle concurrent invocations from different threads.

C. A bean with bean-managed transactions must commit or roll back any transaction before returning from a business method.

D. The container passivates and actives them using methods annotated with @PrePassivate and @PostActivate annotations.

Correct Answer: AC

Note:

* Session beans can either be stateful or stateless. With stateful beans, the EJB container saves internal bean data during and in between method calls on the client\\'s behalf. With stateless beans, the clients may call any available instance of an instantiated bean for as long as the EJB container has the ability to pool stateless beans. This enables the number of instantiations of a bean to be reduced, thereby reducing required resources.

Incorrect:

B: Stateful session beans maintain state both within and between transactions. Each stateful session bean is therefore associated with a specific client.

D: @PrePassivate(javax.ejb.PrePassivate) :

If a stateful session bean instance is idle for too long, the container might passivate it and store its state to a cache.

The method tagged by this annotation is called before the container passivates the bean instance. This annotation is only applicable to stateful session beans.

---

**QUESTION 11**

Given this code snippet from a JMS message driven bean class X: When this bean class handles a message, which is correct?

```
11.    public X() { System.out.print("1 "); }
12.    public void onMessage(Message m)
          throws java.rmi.RemoteException {
13.       try {
14.          TextMessage tm = (TextMessage) m;
15.          String text = tm.getText();
16.          System.out.print("2 ");
17.       } catch (JMSException e) {
18.          throw new java.rmi.RemoteException();
19.       }
20.    }
```

A. After a message delivery the result is 1.

B. After a message delivery the result is 2.

C. After a message delivery the result is 12.

D. After a message delivery an exception is thrown.

E. After a message delivery the result is unpredictable.

F. This is NOT a compliant JMS message-driven bean.

Correct Answer: F

---

**QUESTION 12**

Given the following stateless session bean implementation classes:

```
10.  @TransactionAttribute(TransactionAttributeType.MANDATORY)
11.  public class MySuper {
12.     public void methodA() {}
13.     public void methodB() {}
14.  }

10.  @Stateless
11.  public class MyBean extends MySuper implements MyInt {
12.     public void methodA() {}
13.
14.     @TransactionAttribute(TransactionAttributeType.REQUIRES_NEW)
15.     public void methodC() {}
16.  }

10.  @Remote()
11.  public interface MyI
12.     public void method    ;
13.     public void meth   ();
14.     public void m    C();
15.  }
```

Assuming no other transaction-related metadata, what are the transaction attributes of methodA, methodB, and methodC respectively?

A. MANDATORY, MANDATORY , and MANDATORY

B. REQUIRED, MANDATORY, and REQUIRES_NEW

C. MANDATORY, MANDATORY , and REQUIRES_NEW

D. REQUIRED, REQUIRES_NEW, and REQUIRES_NEW

Correct Answer: C

Note:

*

 You can only bring out the effects of transaction attributes only when you call the method through a session bean instance and NOT through a direct method call. Even if your methods are within the same bean, you need to get the local

instance of the same bean and call through its local interface instead of a direct method invoke.

*

 The enum TransactionAttributeType is used with the TransactionAttribute annotation to specify whether the methods of a session bean or message driven bean are called with a valid transaction context.

*

 MANDATORY

If a client invokes the enterprise bean\\\'s method while the client is associated with a transaction context, the container invokes the enterprise bean\\\'s method in the client\\\'s transaction context.

*

REQUIRES_NEW

The container must invoke an enterprise bean method whose transaction attribute is set to REQUIRES_NEW with a new transaction context.

Reference: TransactionAttributeType.MANDATORY

Latest 1Z0-895 Dumps         1Z0-895 PDF Dumps         1Z0-895 Practice Test

To Read the Whole Q&As, please purchase the Complete Version from Our website.

# Try our product !

100% Guaranteed Success
100% Money Back Guarantee
365 Days Free Update
Instant Download After Purchase
24x7 Customer Support
Average 99.9% Success Rate
More than 800,000 Satisfied Customers Worldwide
Multi-Platform capabilities - Windows, Mac, Android, iPhone, iPod, iPad, Kindle

We provide exam PDF and VCE of Cisco, Microsoft, IBM, CompTIA, Oracle and other IT Certifications.
You can view Vendor list of All Certification Exams offered:

https://www.passapply.com/allproducts

## Need Help

Please provide as much detail as possible so we can best assist you.
To update a previously submitted ticket:

Any charges made through this site will appear as Global Simulators Limited.
All trademarks are the property of their respective owners.
Copyright © passapply, All Rights Reserved.