Oracle

Exam 1z0-895

Java EE 6 Enterprise JavaBeans Developer Certified Expert Exam

Version: 14.0

[Total Questions: 90]

Question No : 1

A developer needs to deliver a large-scale enterprise application that connects developer chooses an EJB 3.1-compliant application server, which three are true about the EJB business component tier? (Choose three.)

A. Load-balancing is NOT a guarantee for all EJB 3.1 containers.
B. Clustering is guaranteed to be supported by the EJB 3.1 container.
C. Thread pooling can be optimized by the Bean Provider programmatically.
D. Bean Providers are NOT required to write code for transaction demarcation.
E. Support for server fail-over is guaranteed for an EJB 3.1-compliant application server.
F. EJB 3.1 compliant components are guaranteed to work within any Java EE 6 application server

Answer: A,C,F

Explanation: The EJB tier hosts the business logic of a J2EE application and provides system-level services to the business componentsproblems include state maintenance, transaction management, and availability to local and remote clients.

The EJB 3.1 specification does not address "high-end" features like clustering (not B), load-balancing (A) and fail-over (not E).

F: The target platform for EJB is Java EE.

Question No : 2

A developer examines a list of potential enterprise applications and selects the most appropriate technologies to use for each application.

For which two applications is EJB an appropriate solution? (Choose two.)

A. To render a GUI for mobile clients.
B. As a container for web-tier components including JSP.
C. As a Web service endpoint accessed by non-Java clients.
D. To receive and respond to HTTP Post requests directly from a web browser.
E. As an online shopping cart which can persist across multiple sessions with a single client.

Answer: C,E

Question No : 3

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.

Answer: B,D

Explanation: 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 typicallycontains the business logic.

Question No : 4

Assume you would like to receive notification from the container as a stateless session bean transitions to and from the ready state.

Which of the following life cycle back annotations would you use? (Choose one.)

A. @PostConstruct, @PostDestroy
B. @PostConstruct, @PreDestroy
C. @PreConstruct, @PostDestroy
D. @PostConstruct, @PostDestroy, @Remove
E. @PostConstruct, @PreDestroy, @Remove

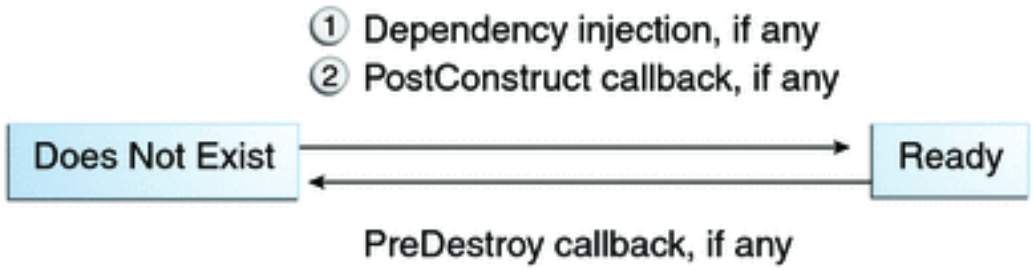
Answer: B

Explanation: The Lifecycle of a Stateless Session Bean

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. The bean's instance is then ready for garbage collection.

Lifecycle of a Stateless Session Bean:



Note: An enterprise bean goes through various stages during its lifetime, or lifecycle. Each type of enterprise bean (stateful session, stateless session, singleton session, or message-driven) has a different lifecycle.

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

Question No : 5

Which API must an EJB 3.1 container make available to enterprise beans at runtime? (Choose one)

A. The JXTA 1.1 API
B. The MIDP 2.0 API
C. The Java SE 6 JNDI API
D. The Java SE 5 JDBC API

Answer: C,D

Question No : 6

A developer wants to write a stateful session bean using the following interface as local business interface:

```
1. package acme;
2. public interface Bar {
3. public void bar ();
4. }
```

Assuming there is not an ejb-jar.xml file, which code can be inserted into Lines 4-6 below to define the bean with the ejb name of BarBean?

```
1. package acme;
2. import javax.ejb.*;
3. import java.io.*;
4.
5.
6.
7. }
```

A. @Stateful
public class BarEJB implements Bar {
public void bar () {}
B. @Stateful (name = “Bar”)
public class Barbean implements Bar {
public void bar () {}
C. @Stateful
public class BarBean implements Serializable, Bar {
public void bar () {}
D. @Stateful (name = “bar”)
public class BarBean implements Serializable, Bar {
public void bar () throws java.rmi.RemoteException {}

Answer: C

Question No : 7

A developer creates a stateful session bean that is used by many concurrent clients. The clients are written by other development team; and it is assumed that these clients might not remove the bean when ending their session. The number of concurrent sessions will be greater than the defined bean cache size.

The developer must consider that the state of the session bean can be influenced by either passivation or timeout.

Which three actions should the developer take to make the bean behave correctly in passivation and timeout situations? (Choose three.)

A. Release references to resources in a @Remove annotated method.
B. Re-establishreferences to resources in an @Init annotated method.
C. Release references to resources in a @PreDestroy annotated method.
D. Release references to resources in a@PrePassivate annotated method.
E. Re-establish references to resources in a @PostActivate annotated method.

Answer: C,D,E

Question No : 8

A stateful session bean contains a number of instance variables. The types of instance variables A and B are serializable. Instance variable B is a complex type which is populated by many business calls, and can, therefore, not be refilled by the client without starting all over. A helper instance variable C is defined as having a Serializable type, and can hold all the information which is in variable B. for example, B is of type XML-DOM tree and C of Type String.

Which two solutions, when combined, maintain the state of the session bean over a passivation and activation by the container? (Choose two.)

A. The value of helper variable C is used to create the value of Instance variable B in the beans no-arg constructor.
B. The value of helper variable C is used to create the value of instance variable B in a @postcreate annotated method.
C. The value of helper variable C is used to create the value of instance variable B in a @postActivate annotated method.
D. Instance variable A must be made null and instance variable B must be converted to a Serializable type and assigned to another instance variable in a @preDestroy annotated method.
E. Instance variable A must be defined transient. Instance variable B must be converted to

a Serializable type, set to null, and assigned to the instance variable C in a @PrePassivate annotated method.

Answer: C,E

Question No : 9

A developer writes a stateful session bean FooBean with one remote business interface Foo. Foo defines an integer / setter method pair implemented as:

10. private int value;

11. public void setValue (int i) {value = i; }

12. public int getValue () {return value; }

A session bean ClientBean has a business method doSomething and an ejb-ref with ejb-ref-name “fooRef” that is mapped to FooBean’s Foo interface.

11. @Resource private SessionContext SessionCtx;

12. public void doSomething () {

13. Foo foo1 = (Foo) sessionCtx.lookup("fooRef");

14. Foo foo2 = (Foo) sessionCtx.lookup("fooRef");

15. foo1.setvalue(1);

Which statement is true after the code at line 15 completes?

A. Foo1.getValue () = = 0 and foo2.getValue() = = 0
B. Foo1.getValue () = = 0 and foo2.getValue() = = 1
C. Foo1.getValue () = = 1 and foo2.getValue() = = 0
D. Foo1.getValue () = = 1 and foo2.getValue() = = 1

Answer: D

Explanation: Foo1 and Foo2 references the same object.

Question No : 10

A developer writes a stateless session bean FooBean with one remote business interface FooRemote containing one business method foo. Method foo takes a single parameter of application-defined type MyData.

11. public class MyData implements java.io.Serialization {

12. int a;

13. }

Methods foo is implemented with the FooBean class as:

11. public void foo (MyData data) {

12. data.a = 2;

13. }

Another session bean within the same application has a reference to FooRemote in variable fooRef and calls method foo with the following code:

11. MyData data = new MyData();

12. data.a = 1;

13. Fooref.foo(data);

14. System.out.printIn(data.a);

What is the value of data.a when control reaches Line 14 of the client?

A. 0
B. 1
C. 2

Answer: B

Question No : 11

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.

Answer: A,C

Explanation: * 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.

C:A stateless session bean is relatively easy to develop and also very efficient. 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 No : 12

A developer wants to release resources within a stateless session bean class. The cleanup method should be executed by the container before an instance of the class is removed. The deployment descriptor is NOT used.

Which three statements are correct? (Choose three.)

A. The cleanup method may declare checked exceptions.
B. The cleanup method must have no arguments and return void.
C. The cleanup method is executed in an unspecified transaction and security context.
D. The developer should mark the cleanup method with the @PreDestroy annotation.
E. The developer should mark the cleanup method with the @PostDestroy annotation.
F. The cleanup method is executed in the transaction and security context of the last business method Invocation.

Answer: B,C,D

Question No : 13

A developer creates a stateless session bean. This session bean needs data from a remote system. Reading this data takes a long time. Assume that the data will NOT change during the life time of the bean and that the information to connect to the remote system is defined in JNDI.

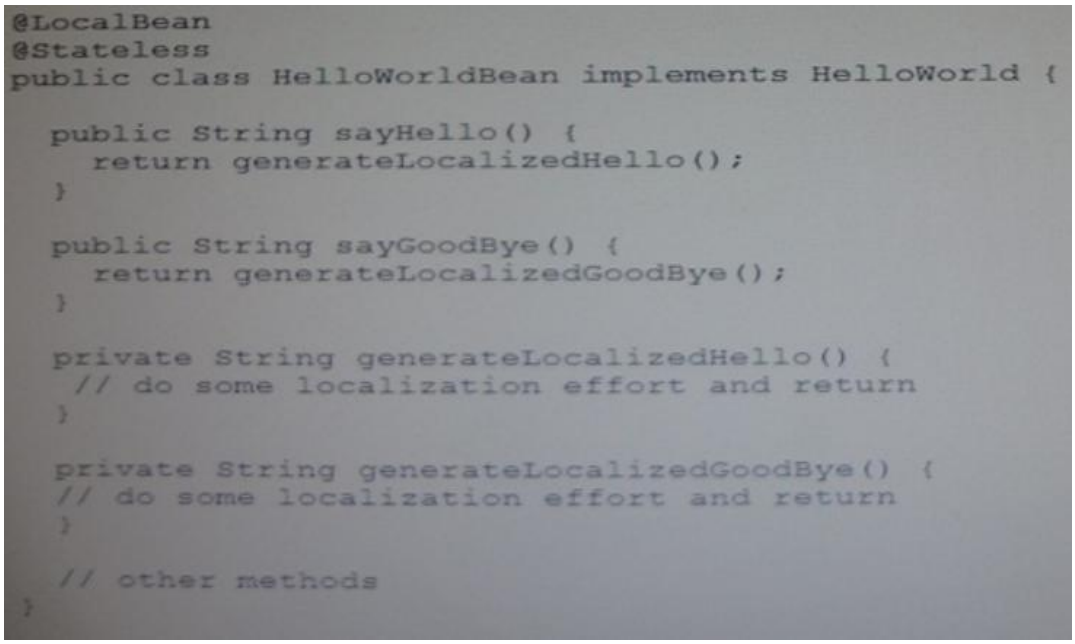
Which statement describes how to manage the data correctly?

A. Read the data in the bean's constructor.
B. The data can only be read in the bean's business methods.
C. Read the data in a method which is annotated with @PrePassivate.
D. Read the data in a method which is annotated with @PostActivate.
E. Read the data in a method which is annotated with @PostConstruct.

Answer: E

Question No : 14

Suppose an EJB named HelloWorldBean is deployed as a standalone ejb-jar. Assuming the HelloWorldBean is implemented as follows:



```
@LocalBean
@Stateless
public class HelloWorldBean implements HelloWorld {

  public String sayHello() {
    return generateLocalizedHello();
  }

  public String sayGoodBye() {
    return generateLocalizedGoodBye();
  }

  private String generateLocalizedHello() {
   // do some localization effort and return
  }

  private String generateLocalizedGoodBye() {
  // do some localization effort and return
  }

  // other methods
}
```

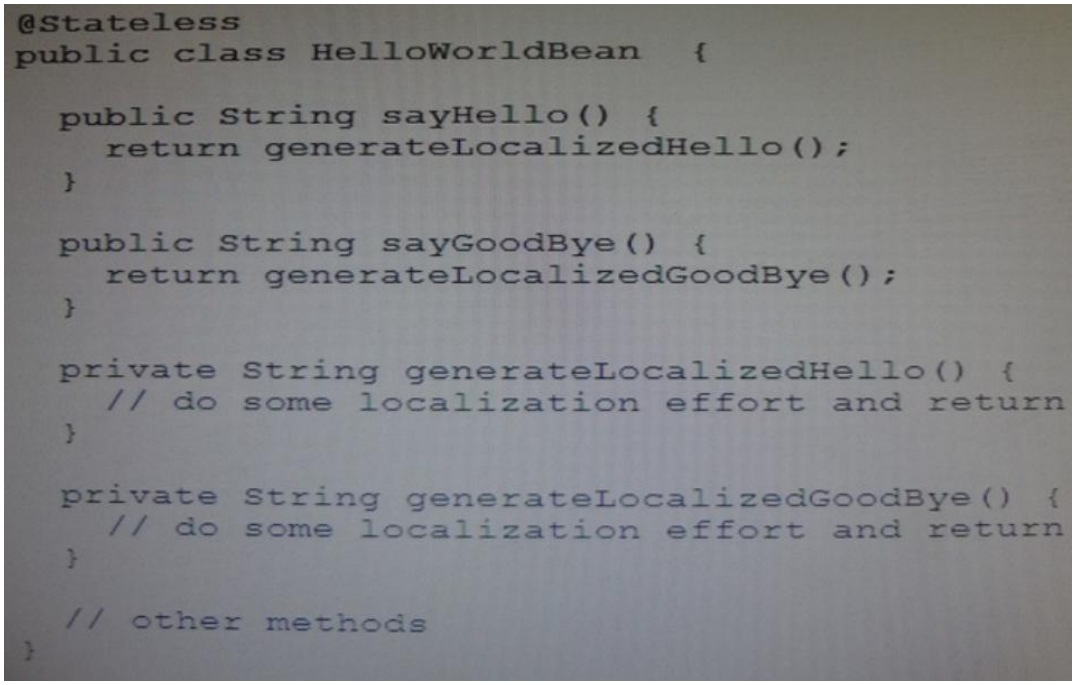
Which HelloWorldBean methods are accessible by another EJB within the same ejb-jar?

A. All of the methods declared in HelloWorldBean
B. All of the methods declared in HelloWorld and HelloWorldBean
C. All of the public methods declared in HelloWorldBean
D. All of the public methods declared in HelloWorld andall of the methods declared in HelloWorldBean

Answer: C

Question No : 15

Given the following stateless session bean:



```
@Stateless
public class HelloWorldBean  {

  public String sayHello() {
    return generateLocalizedHello();
  }

  public String sayGoodBye() {
    return generateLocalizedGoodBye();
  }

  private String generateLocalizedHello() {
    // do some localization effort and return
  }

  private String generateLocalizedGoodBye() {
    // do some localization effort and return
  }

  // other methods
}
```

How would you change the EJB to prevent multiple clients from simultaneously accessing the sayHello method of a single bean instance?

A. Convert sayHello into a synchronized method
B. Execute the call to generateLocalizedHello in a synchronized block
C. Convert generateLocalizehello into a synchronized method
D. Convert HelloWordBean into a singleton bean
E. No changes are needed

Answer: A

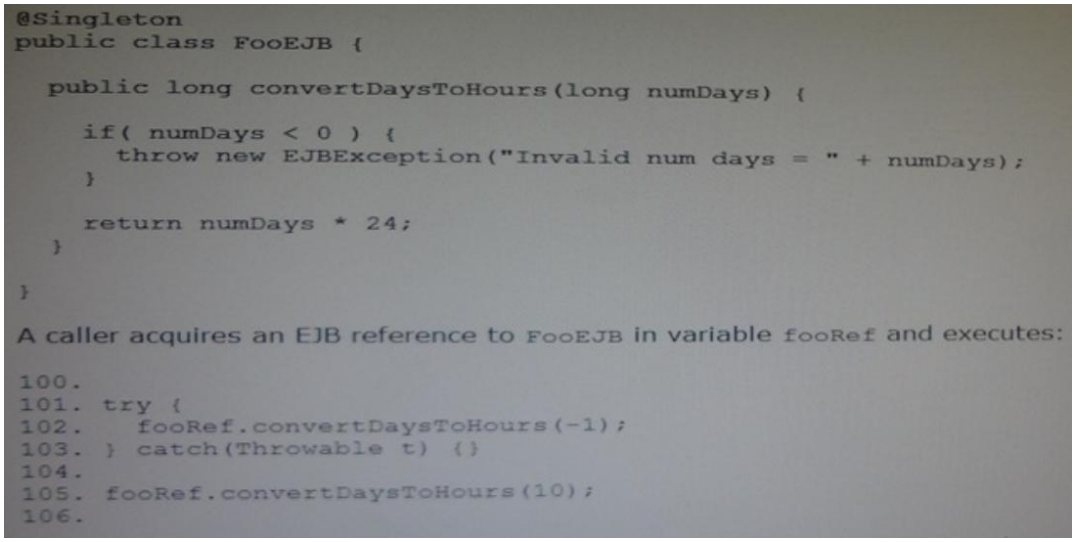
Explanation: * It is not possible for two invocations of synchronized methods on the same object to interleave. When one thread is executing a synchronized method for an object, all other threads that invoke synchronized methods for the same object block (suspend execution) until the first thread is done with the object.

* When a synchronized method exits, it automatically establishes a happens-before relationship with any subsequent invocation of a synchronized method for the same object. This guarantees that changes to the state of the object are visible to all threads.

Reference: The Java Tutorial, Synchronized Methods

Question No : 16

Given singleton bean FooEJB:



```
@Singleton
public class FooEJB {

  public long convertDaysToHours(long numDays) {

    if( numDays < 0 ) {
      throw new EJBException("Invalid num days = " + numDays);
    }

    return numDays * 24;
  }

}
```

A caller acquires an EJB reference to `FooEJB` in variable `fooRef` and executes:

```
100.
101. try {
102.    fooRef.convertDaysToHours(-1);
103. } catch(Throwable t) {}
104.
105. fooRef.convertDaysToHours(10);
106.
```

How many distinct FooEJB bean instances will be used to process the code on the lines 101-105?

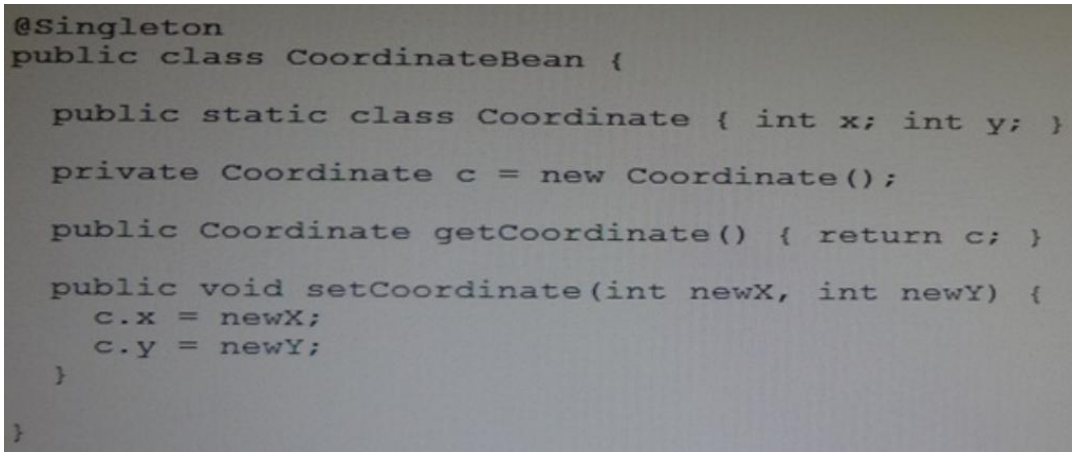
A. 0
B. 1
C. 2

Answer: B

Explanation: Java has several design patterns Singleton Pattern being the most commonly used. Java Singleton pattern belongs to the family of design patterns, that govern the instantiation process. This design pattern proposes that at any time there can only be one instance of a singleton (object) created by the JVM.

Question No : 17

A developer writes a Singleton bean that holds state for a single coordinate:



```
@Singleton
public class CoordinateBean {

  public static class Coordinate { int x; int y; }

  private Coordinate c = new Coordinate();

  public Coordinate getCoordinate() { return c; }

  public void setCoordinate(int newX, int newY) {
    c.x = newX;
    c.y = newY;
  }

}
```

An update thread acquires an EJB reference to CoordinateBean and alternates between invoking SetCoordinate (0, 0) and SetCoordinate (1, 1) in a loop.

At the same time, ten reader threads each acquire an EJB reference to CoordinateBean and invoke getCoordinate () in a loop.

Which represents the set of all possible coordinate values [X, Y] returned to the reader threads?

A. [0, 0]
B. [1, 1]
C. [0, 0], [1, 1]
D. [0, 0], [0, 1], [1, 0], [1, 1]

Answer: B

Explanation: The value could be either [0,0] or [1,1].

Note:
Java has several design patterns Singleton Pattern being the most commonly used. **Java Singleton pattern**belongs to the family of design patterns, that govern the instantiation process. This design pattern proposes that at any time there can only be one instance of a singleton (object) created by the JVM.

Question No : 18

Assume a client will be accessing a Singleton bean.

Which client views is a Singleton bean capable of exposing? (Choose two)

A. Web Service
B. Message listener
C. EJB 2.x Remote Home
D. EJB 3.x local business
E. Java Persistence API entity

Answer: A,B

Explanation: Singleton session beans are appropriate in the following circumstances.
* State needs to be shared across the application.
* A single enterprise bean needs to be accessed by multiple threads concurrently.
* The application needs an enterprise bean to perform tasks upon application startup and shutdown.
* The bean implements a web service. (A)

B: An interceptor method you define in a separate interceptor class takes an invocation context as argument: using this context, your interceptor method implementation can access details of the original session bean business method or message-driven bean message listener method invocation.
Singleton Interceptors
If your interceptors are stateless, you can use an OC4J optimization extension to the EJB 3.0 specification that allows you to specify singleton interceptors. When you configure a session bean or message-driven bean to use singleton interceptors and you associate the bean with an interceptor class, OC4J creates a single instance of the interceptor class that all bean instances share. This can reduce memory requirements and life cycle overhead.

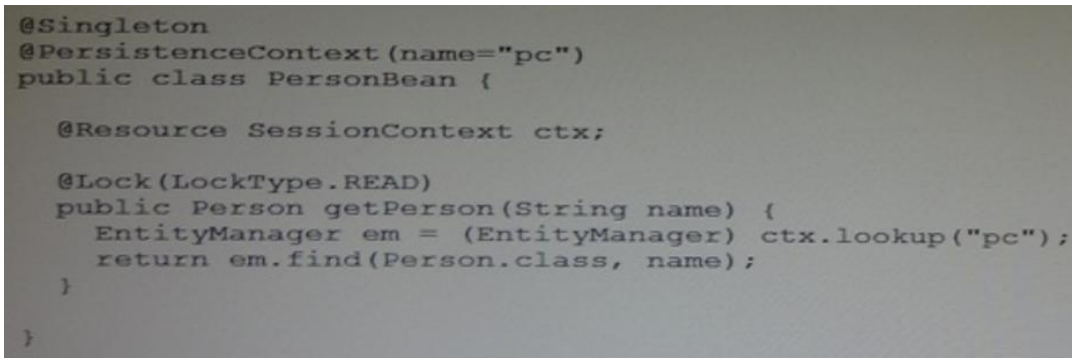
Note:
Singleton session beans offer similar functionality to stateless session beans but differ from them in that there is only one singleton session bean per application, as opposed to a pool of stateless session beans, any of which may respond to a client request. Like stateless session beans, singleton session beans can implement web service endpoints.

Reference: The Java EE 6 Tutorial, What Is a Session Bean?

Reference: Oracle Containers for J2EE Enterprise JavaBeans Developer's Guide, How do you use an Enterprise Bean in Your Application

Question No : 19

A developer writes a Singleton bean that uses the java Persistence API within a business method:



```
@Singleton
@PersistenceContext(name="pc")
public class PersonBean {

  @Resource SessionContext ctx;

  @Lock(LockType.READ)
  public Person getPerson(String name) {
    EntityManager em = (EntityManager) ctx.lookup("pc");
    return em.find(Person.class, name);
  }

}
```

Two different concurrently executing caller threads acquire an EJB reference to PersonBean and each invoke the getPerson () method one time. How many distinct transaction are used to process the caller invocations?

A. 0
B. 1
C. 2

Answer: B

Explanation: Only one transaction is required. LockType READ allows simultaneous access to singleton beans.

Note: READ

public static final LockType READ
For read-only operations. Allows simultaneous access to methods designated as READ, as long as no WRITE lock is held.

Reference: javax.ejb, Enum LockType

Question No : 20

Given the following client-side code that makes use of the session bean Foo:

10. @EJB Foo bean1;

12. @EJB Foo bean2;

// more code here

20. boolean test1 = beanl.equals(bean1);

21. boolean test2 = beanl.equals(bean2) ;

Which three statements are true? (Choose three)

A. If Foo is stateful, test1 is true, and test2 is true.
B. If Foo is stateful, test1 is true, and test2 is false.
C. If Foo is stateless, test1 is true, and test2 is true.
D. If Foo is stateless, test1 is true, and test2 is false.
E. If Foo is singleton, test1 is true, and test2 is true.
F. If Foo is singleton, test1 is true, and test2 is false.

Answer: B,C

Question No : 21

A developer writes a stateful session bean FooBean with two local business interfaces Foo and bar. The developer wants to write a business method called getBar for interface Foo that returns a Bar reference to the same session bean identity on which the client onvokes getBar.

Which code, when inserted on line 12 below implements the getBar method with the wanted behavior?

10. @Resource SessionContext sessionCtx;

11. public Bar getbar () {

12.

13. }

A. Return (bar) this;
B. Return (bar) new FooBarBean();
C. Return (bar) sessionCtx.lookup(“FooBarBean”)
D. Return (bar) sessionCtx.getBusinessObject(Bar.class);

E. Return (bar) session Ctx.lookup("java: comp/env/ejb/FooBarBean");

Answer: D

Question No : 22

Given the following code in an EJB session bean:

```
10.  @Resource(name="jdbc/employeeDB")
11.  private DataSource dataSource;
12.
13.  public void lookupEmployee(String id) {
14.    InitialContext ic = new InitialContext();
15.    // insert code here
16.    DataSource ds = (DataSource) obj;
17.  }
```

Which code, inserted at Line 15, portably looks up the injected resource?

A. Object obj = ic.lookup ("employeeDB");
B. Object obj = ic.lookup ("dataSource");
C. Object obj = ic.lookup ("jdbc/employeeDB");
D. Object obj = ic.lookup ("java:comp/env/employeeDB");
E. Object obj= ic.lookup ("java:cmp/env/jdbc/employeeDB);

Answer: E

Question No : 23

A developer is writing client code to access a session bean deployed to a server instance.

The client can access the session bean under which of these circumstances? (Choose three)

A. The client is deployed in the same JVM as the session bean and the session bean has a local interface.
B. The client is deployed in the same JVM as the session bean and the session bean exposes a no-interface view.
C. The client is deployed in a different JVM from the session bean and the session bean has a local interface.

D. The client is deployed in a different JVM from the session bean and the session bean has a remote interface.
E. The client is deployed in a different JVM from the session bean and the session bean has a no-interface implementation.

Answer: A,B,D

Explanation: D: If your architecture has a requirement whereby the client application (web application or rich client) has to run on a different JavaVirtual Machine (JVM) from the one that is used to run the session beans in an EJB container, then you need to use the remote interface.

Question No : 24

A developer writes three interceptor classes: AInt, BInt, and CInt. Each interceptor class defines an AroundInvoke method called interceptor. In the ejb-jar.xml descriptor, CInt is declared as the default interceptor.

FooBean is a stateless session bean with a local business interface Foo that declares a method Foo ():

```
10. @Stateless
11. @Interceptors(AInt.class)
12. public class FooBean Implements Foo {
13.
14.  @Interceptors (BInt.class)
15. @ExcludeClassInterceptors
16. public void foo () {}
17. }
```

What is the interceptor order when the business method foo () is invoked?

A. BInt
B. CInt, BInt
C. CInt, AInt, BInt

D. BInt, AInt, CInt

Answer: B

Explanation: The default Intercepter, CInt, comes first. The class intercepter AInt is excluded by @ExcludeClassInterceptors, so the Method Intercepter BInt would be next in order.

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: Annotation Type ExcludeClassInterceptors
Used to exclude class-level interceptors for a business method or timeout method of a target class.

Reference: EJB Interceptors

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

Question No : 25

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 <interceptor-binding> section:

<interceptor-binding>

<ejb-name>FooBean</ejb-name>

<interceptor-order>

<interceptor.class>com.acme.AInt</interceptor-class>

</interceptor-order>

</interceptor.binding>

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

Answer: B

Explanation: 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 No : 26

How many interceptor classes can be applied to a single stateful session bean?

A. a maximum of one
B. any number may be applied
C. one for each business method
D. one for each business interface

Answer: B

Explanation: The @Interceptors annotation can take an array of classes, so you can bind more than one class-level interceptor this way, e.g.

```
@Stateless
@Interceptors ({TracingInterceptor.class, SomeInterceptor.class})
public class EmailSystemBean
{

}
```

Reference: EJB Interceptors

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

Question No : 27

A developer writes an interceptor class called FooInterceptor containing the following AroundInvoke method:

11. @AroundInvoke

12. public Object intercept (InvocationContext ctx) {

13. return “intercepted”;

14. }

FooInterceptor is applied to a business method in a stateless session bean:

11. @Interceptors (FooInterceptor.class)

12. public String testzero(int i) {

13. return (i = = 0) ? “zero”: “not zero”

14. }

Which describes the result when a client invokes the testzero method with a value of 1?

A. The interceptor method is NEVER invoked.
B. The clientreceives a return value of “zero”.
C. The client receives a return value of “not zero”.
D. The client receives a return value of “intercepted”.

Answer: D

Question No : 28

A bean developer wants to write a stateless session bean class that implements the following remote business interface:

@Remote

Public interface Foo {

Void bar () throws Exception;

Which bean class method is valid?

A. @Asynchronous
public void bar () throws Exception { . . . }

B. @Asynchronous
Future <void> bar () { . . .}
C. void bar () throws Exception { . . . }
D. public void bar () { . . . }

Answer: B

Explanation: with EJB 3.1, you can use a simple session EJB with the @Asynchronous annotation on the method which must be called asynchronously.

```
@Stateless
@Remote(HelloEjbAsynchronousRemote.class)
public class HelloEjbAsynchronous implements HelloEjbAsynchronousRemote {

@Asynchronous
@Override
public Future<String> ejbAsynchronousSayHello(String name){
```

If your method has a return value, your method has to return an AsyncResult object which is an implementation of Future.

Question No : 29

Which is true about caller security principal propagation for asynchronous EJB method Invocations?

A. Caller security principal propagates along with an asynchronous EJB method invocation.
B. Caller security principal does not propagate along with an asynchronous EJB method invocation.
C. Caller security principal propagates along with an asynchronous EJB method invocation only if the target bean has at least one protected method.
D. Caller security principal propagates along with an asynchronous EJB method invocation only if the calling bean has at least one protected method.

Answer: D

Explanation: One important caveat of asynchronous method invocation is that transactions are not propagated to the asynchronous method—a new transaction will be started for the asynchronous method. However, unlike transactions, the security principal will be propagated.

Declarative security is based only on the principal and the method being invoked, whereas programmatic security can take state into consideration.

Question No : 30

A bean developer writes a stateless session bean FooEJB with the following asynchronous business method:

```
@Asynchronous
public Future<Integer> fooAsync () {
System.out.printIn (“begin”);
int i = 1;
System.out.print(“end”);
Return new AsyncResult<Integer> (i);
}
```

Given the following code, where fooRef is an EJB reference to FooEJB:

```
Future<Integer> fooFuture = fooref.fooAsync();
fooFuture.cancel (true);
```

Which two represents possible system output after all processing has completed? (Choose two)

A. Begin end
B. Begin
C. End
D. 1
E. <no output>

Answer: D,E

Explanation: Either it will run and return 1, or it will be cancelled and produce no output.

Note:EJB 3.1 can support a return type of java.util.concurrent.Future<V>, where V