1. What does the EJB specification architecture define?

|  |  |
| --- | --- |
| A. | Transactional components |
| B. | Distributed object components |
| C. | Server-side components |
| D. | **All of the above** |

2. What executes EJB components?

|  |  |
| --- | --- |
| A. | A web server |
| B. | An application server |
| C. | **An EJB container** |
| D. | A database server |

3. What do enterprise beans use to communicate with the EJB container to get runtime context information?

|  |  |
| --- | --- |
| A. | **The javax.ejb.EJBContext provided by the container** |
| B. | A JNDI ENC context |
| C. | A javax.ejb.EJBHome object provided by the container |
| D. | A javax.ejb.EJBMetaData object provided by the container |

4. Through what interface does an application create, find, and remove enterprise beans?

|  |  |
| --- | --- |
| A. | java.rmi.Remote |
| B. | **javax.ejb.EJBHome** |
| C. | javax.ejb.EJBObject |
| D. | javax.ejb.EntityBean |

5. What type of enterprise bean is used to embody business objects?

|  |  |
| --- | --- |
| A. | javax.ejb.EnterpriseBean |
| B. | java.rmi.Remote |
| C. | javax.ejb.SessionBean |
| D. | **javax.ejb.EntityBean** |

6. What type of enterprise bean is used to embody application processing state information?

|  |  |
| --- | --- |
| A. | javax.ejb.EnterpriseBean |
| B. | javax.rmi.Remote |
| C. | **javax.ejb.SessionBean** |
| D. | javax.ejb.EntityBean |

7. What interface must the enterprise bean implement so that an application can invoke its operations?

|  |  |
| --- | --- |
| A. | javax.ejb.EntityBean |
| B. | javax.ejb.EJBHome |
| C. | **javax.ejb.EJBObject** |
| D. | javax.rmi.Remote |

8. At what point, precisely, in the life-cycle is a container-managed entity bean considered created?

|  |  |
| --- | --- |
| A. | Immediately prior to the execution of its ejbCreate() method |
| B. | **Immediately after the execution of its ejbCreate() method** |
| C. | After the CMP bean's data has been committed to the underlying persistent datastore |
| D. | During the execution of its ejbPostCreate() method |

9. What distinguishes a bean-managed persistent (BMP) enterprise bean from a container-managed persistent (CMP) enterprise bean?

|  |  |
| --- | --- |
| A. | A BMP bean must implement the ejbLoad() and ejbStore() methods |
| B. | A BMP bean can implement persistence to custom datastores such as legacy systems |
| C. | A BMP bean is responsible for managing its own persistence to a persistent datastore |
| D. | **All of the above** |

10. What is a deployment descriptor?

|  |  |
| --- | --- |
| **A.** | **An XML file format used by the container to learn about the attributes of a bean, such as transactional characteristics and access control** |
| B. | A method for transporting enterprise beans back and forth between systems |
| C. | An XML file used by enterprise bean clients to learn about the attributes of a bean, such as access control and transactional characteristics. |
| D. | A format for bundling enterprise beans for delivery to customers |

1. A local home or component interface for an EJB can only be used by:

|  |  |
| --- | --- |
| A. | Another EJB |
| B. | A web-tier client |
| C. | A business logic-tier client |
| D. | **A client located in the same JavaTM Virtual Machine (JVM)**[**1**](http://java.sun.com/developer/Quizzes/ejb/professionalejb.html#jvm) |

2. Local interfaces have declarative:

|  |  |
| --- | --- |
| A. | **Transactions and security** |
| B. | Transactions only |
| C. | Security only |
| D. | Neither, for performance reasons |

3. An entity in a unidirectional relationship that is the target of a role with a cmr field:

|  |  |
| --- | --- |
| A. | **Must have local home and component interfaces** |
| B. | Must have remote home and compnent interfaces |
| C. | May have either local or remote interfaces, but not both |
| D. | May have any combination of local or remote interfaces |

4. An entity in a unidirectional relationship that is the source of a role with a cmr field:

|  |  |
| --- | --- |
| A. | Must have local home and component interfaces |
| B. | Must have remote home and compnent interfaces |
| C. | May have either local or remote interfaces, but not both |
| D. | **May have any combination of local or remote interfaces** |

5. The bean class for an entity that uses the EJB 2.0 model of container-managed persistence:

|  |  |
| --- | --- |
| A. | Must implement java.io.Serializable |
| B. | Is only used for better integration with popular IDEs |
| C. | **Must be abstract** |
| D. | Must not be abstract |

6. The legal collection class type(s) to represent a many-valued relationship are:

|  |  |
| --- | --- |
| A. | java.util.Collection only |
| B. | **java.util.Collection and java.util.Set** |
| C. | java.util.List, java.util.Set, and java.util.Map |
| D. | java.util.Map only |

7. If you call a "set" abstract accessor for a cmr field, it can:

|  |  |
| --- | --- |
| A. | Throw a java.sql.SQLException |
| B. | Automatically cascade delete an entity |
| C. | **Automatically change the value of cmr fields in up to three additional beans** |
| D. | You can never call an abstract accessor |

8. You can only specify cascade-delete on a relationship role if:

|  |  |
| --- | --- |
| A. | The role has a multiplicity of 'One' |
| B. | **The other role in the relationship has a multiplicity of 'One'** |
| C. | The role has a multiplicity of 'Many' |
| D. | The other role in the relationship does not already use cascade-delete |

9. If an entity has a relationship to another entity, it must:

|  |  |
| --- | --- |
| A. | Declare that entity reference in the deployment descriptor using the <ejb-ref> element |
| B. | Declare that entity reference in the deployment descriptor using the <ejb-local-ref> element |
| C. | Declare that entity reference in the deployment descriptor using the <ejb-link> element |
| D. | **No entity reference declaration is required** |

10. The bean developer must be cautious when iterating over a relationship collection class, because:

|  |  |
| --- | --- |
| A. | Concurrent transactions can modify the values |
| B. | Changes to its contents can violate foreign key constraints in the database |
| C. | **Changes to its contents can trigger a referential integrity change to the collection class itself** |
| D. | The container developer is not required to support the Iterator class's next method |

11. The EJB 2.0 specification introduces ejbSelect methods. These are:

|  |  |
| --- | --- |
| A. | **Abstract methods in the bean class that call a query** |
| B. | The local home interface equivalent of a finder method |
| C. | Methods in the component interface that call a query |
| D. | A method that allows the bean developer to choose between a local and remote interface |

12. The new EJB Query Language (EJB-QL) has three clauses: select, from, and where. Of these:

|  |  |
| --- | --- |
| A. | Only the from clause is mandatory |
| B. | **Only the select and from clauses are mandatory** |
| C. | Only the where clause is mandatory. |
| D. | All clauses are mandatory. |

13. In EJB-QL, date and time literals are represented by:

|  |  |
| --- | --- |
| A. | A string in the format MM-DD-YYYY HH:MM:SS +MILLISECONDS |
| B. | A string in a locale-determined format |
| C. | A string in a vendor specific format |
| D. | **A long value that represents a millisecond count, such as 979837813000** |

14. The type of interface (local or remote) returned from an ejbSelect statement can be:

|  |  |
| --- | --- |
| A. | Specified in the query |
| B. | **Specified in the deployment descriptor** |
| C. | Specified by the client |
| D. | Is always a local interface |

15. The functions available for use in the where clause of an EJB-QL query are:

|  |  |
| --- | --- |
| A. | All functions defined in standard SQL |
| B. | All functions defined by the target database |
| C. | All functions with defined escapes in JDBC 2.0 |
| D. | **A limited subset of the functions defined for JDBC 2.0** |

**Enterprise JavaBeans 3.0 Simplified API Quiz**

1. How many Local business interfaces can one EJB 3.0 Session bean have?

|  |  |
| --- | --- |
| A. | 0 |
| B. | 1 |
| C. | 2 |
| **D.** | **As many as it wants.** |

2. Given a Session bean class with the following annotations:

private @EJB(name="fooejbref") Foo foo;

private @Resource SessionContext sessionCtx;

Which of the following could be used to retrieve the ejb dependency declared above?

|  |  |
| --- | --- |
| A. | (Foo) sessionCtx.lookup("foo"); |
| B. | (Foo) sessionCtx.lookup("fooejbref"); |
| C. | InitialContext ic = new InitialContext(); (Foo) ic.lookup("java:comp/env/foo"); |
| D. | InitialContext ic = new InitialContext(); (Foo) ic.lookup("java:comp/env/fooejbref"); |
| E | A and C |
| **F** | **B and D** |

3. Which are valid declarations for a Local business interface?

public interface Foo1 {

public void foo();

}

import javax.ejb.Local;

@Local

public interface Foo2 {

public void foo();

}

import javax.ejb.\*;

@Local

public interface Foo3 extends EJBLocalObject {

public void foo();

}

import javax.ejb.\*;

@Local

@Remote

public interface Foo4 {

public void foo();

}

|  |  |
| --- | --- |
| A. | Foo1, Foo2, Foo3 |
| B. | All of the above |
| C. | Foo2 |
| **D.** | **Foo1, Foo2** |

4. An interceptor class may contain:

|  |  |
| --- | --- |
| A. | AroundInvoke methods. |
| B. | Lifecycle Callback methods. |
| **C.** | **A and B** |
| D. | None of the above. |

5. Given the following @EJB declarations:

@EJB(beanName="foo") Foo ref1;

@EJB(beanName="foo") Foo ref2;

@EJB(beanName="bar") Bar ref3;

@EJB(beanName="bar") Bar ref4;

Where ejb "foo" is a Stateful Session bean with Local business interface Foo and ejb "bar" is a Stateless Session bean with Local business interface Bar

After injection has completed, what would be the output of the following code?

System.out.println(ref1.equals(ref2));

System.out.println(ref3.equals(ref4));

|  |  |
| --- | --- |
| A. | false false |
| **B.** | **false true** |
| C. | true false |
| D. | true true |

6. Which is a valid declaration of a PostConstruct method in a Session bean class?

public @PostConstruct void postConstruct1() {}

private @PostConstruct void postConstruct2() {}

|  |  |
| --- | --- |
| A. | postConstruct1 |
| B. | postConstruct2 |
| **C.** | **A and B** |
| D. | None of the above. |

7. Given the following Message Driven Bean declaration:

@MessageDriven

public class FooMsgBean implements javax.jms.MessageListener {

public void onMessage(Message m) { ... }

}

Assuming it is deployed without an ejb-jar.xml, which best describes the transactional nature of the onMessage method ?

|  |  |
| --- | --- |
| A. | The bean has bean-managed transactions. It is the bean developer's responsibility to demarcate transactions within onMessage(). |
| B. | The bean has container-managed transactions and the onMessage method has tx attribute NOT\_SUPPORTED. |
| C. | **The bean has container-managed transactions and the onMessage method has tx attribute REQUIRED.** |
| D. | The bean has container-managed transactions and the onMessage method has tx attribute REQUIRES\_NEW. |

8. Given the following Remote business interface:

@Remote

public interface Hello {

public String hello();

}

Which exceptions can be received by the caller as a result of invoking the hello() method?

|  |  |
| --- | --- |
| A. | **javax.ejb.EJBException** |
| B. | java.rmi.RemoteException |
| C. | A and B |
| D. | None of the above. |

9. In a Stateful Session bean, dependency injection takes place:

|  |  |
| --- | --- |
| A. | After the @PostConstruct lifecycle callback is invoked on the bean instance. |
| B. | **Before any business methods or lifecycle callback methods are invoked on the bean instance.** |
| C. | After the ejbCreate method is called on the bean instance. |
| D. | Stateful Session beans do not support dependency injection. |

10. Which of the following results in the removal of an EJB 3.0 Stateful Session bean?

|  |  |
| --- | --- |
| A. | Calling a business method that has been annotated with @Remove. |
| B. | Calling remove() on an EJBObject created from the bean's Adapted Home interface. |
| C. | Throwing a runtime exception from one of the bean's business methods. |
| D. | **All of the above.** |