

ISLEVER

# 1Z0-861

Java Enterprise Edition 5 Business Component  
Developer Certified Professional Upgrade  
Exam

DEMO

<https://www.islever.com/1z0-861.html>

<https://www.islever.com/oracle.html>

For the most up-to-date exam questions and materials, we recommend visiting our website, where you can access the latest content and resources.

---

**QUESTION NO: 1**

A developer wants to create a business interface for both local and remote usage. For performance reasons the remote interface should NOT be called by a client in the same JVM. Which statement is required to accomplish this, assuming there is no deployment descriptor?

- A.** The business methods are defined in one interface which must be annotated with both `@Local` and `@Remote`.
- B.** The business methods are defined twice in one interface. One method is annotated with `@Local` and the other is annotated with `@Remote`.
- C.** The business methods are defined in a common interface by two other interfaces which are annotated with `@Local` and `@Remote` respectively. The bean implements the super interface.
- D.** The business methods are defined in a common interface. It is extended by two interfaces, annotated with `@Local` and `@Remote` respectively. Both interfaces are implemented by the bean class.

**Answer: D**

**Explanation:**

**QUESTION NO: 2**

A developer is creating an entity which is mapped to a table that has a primary key constraint defined on two character columns and would like to use mapping defaults as much as possible to simplify the code. Which two mapping options can be chosen? (Choose two.)

- A.** Use an `@Id` property that constructs a private field as a concatenation of two columns.
- B.** Use a separate class to map those two columns and use an `@IdClass` annotation to denote the primary key field or property in the entity.
- C.** Use a separate `@Embeddable` class to map those two columns and use an `@EmbeddedId` annotation to denote a single primary key field or property in the entity.
- D.** Use a separate `@Embeddable` class to map those two columns and add two fields or properties to the entity, each marked as `@Id`, that correspond to the fields or properties in the embeddable class.
- E.** Use a separate class to map those two columns. Specify that class using `@IdClass` annotation on the entity class. Add two fields or properties to the entity, each marked as `@Id`, that correspond to the fields or properties in that separate class.

**Answer: C,E**

**Explanation:**

**QUESTION NO: 3**

---

---

A developer maps the abstract entity class Account with concrete entity sub. classes CreditCardAccount and SavingsAccount using the single table per class hierarchy strategy. Which two statements are true? (Choose two.)

- A. Instances of CreditCardAccount and SavingsAccount are stored in the same table.
- B. All columns that correspond to fields declared in Account must be defined as nullable in the database.
- C. The fields declared in Account are stored in a different table than the ones declared in CreditCardAccount and SavingsAccount.
- D. All columns that correspond to fields declared in CreditCardAccount or SavingsAccount must be defined as nullable in the database.

**Answer: A,D**

**Explanation:**

#### **QUESTION NO: 4**

A developer writes an enterprise application and packages it into an .ear file. The application contains two persistence units defined at the .ear level with persistence unit names FooPU and BarPU. The application also contains an ejb. jar with one stateless session bean. Which code, when added to the stateless session bean class, injects an EntityManagerFactory at runtime?

- A. @PersistenceUnit  
private EntityManagerFactory emf;
- B. @PersistenceContext  
private EntityManagerFactory emf;
- C. @PersistenceUnit(unitName="BarPU")  
private EntityManagerFactory emf;
- D. @Resource(name="BarPU",type=EntityManagerFactory.class)  
private EntityManagerFactory emf;

**Answer: C**

**Explanation:**

#### **QUESTION NO: 5**

Which two are true about EJB 3.0 exception classes? (Choose two.)

- A. The javax.ejb.NoSuchEJBException is an application exception.
- B. The javax.ejb.EJBException extends java.lang.RuntimeException.
- C. The javax.ejb.EJBTransactionRequiredException is an application exception.

- 
- D. An application exception must NOT be a subclass of java.rmi.RemoteException.
  - E. The javax.ejb.EJBTransactionRolledbackException is an application exception.
  - F. Any subclass of java.lang.RuntimeException is always considered a system exception.

**Answer: B,D**

**Explanation:**

### **QUESTION NO: 6**

Which statement about the combination of mapping defaults, annotations, and XML descriptors is correct?

- A. All mapping annotations must always be processed by the persistence provider.
- B. Some annotations, like the @Entity annotation, must always be processed by the persistence provider.
- C. The mapping information for an entity class specified by annotations and in XML descriptors must be distinct.
- D. If multiple entity listeners are defined, the order in which they are invoked can be defined or overwritten in the XML descriptor.

**Answer: D**

**Explanation:**

### **QUESTION NO: 7**

An Application Assembler is given the following stateless session bean:

- 10. @Stateless public class MyBean implements MyInt {
- 11. @RolesAllowed("SECRET")
- 12. public void methodA(int x) {}
- 13. public void methodA(String y) {}
- 14. public void methodB(String z) {}
- 15.}

A deployment descriptor is also supplied, a portion of which reads as follows:

- 20. <methodD.permission>