1. Which method of the jpa entity manager would you use to force synchronizing the database with the entities of the persistence context?

a.synchronize() **b.flush()**  c.refresh() d.persist()

1. In which life cycle state the JPA entity class is synchronized with the database?

a.New **b.Managed** c.Synchronized d.Detached e.Removed

1. What causes the entity manager to move an Entity instance from the detached to the managed state?

2 Invoke the persist() method of the entity manager on the instance

3 **Invoke the merge() method of the entity manager on the instance**

4 Invoke the manage() method of the entity manager on the instance

5 Invoke the remove() method of the entity manager on the instance

1. Which method of the jpa entity manager would you use to force synchronizing the persistence context from the database?

2 synchronize()

3 flush()

**4 refresh()**

5 persist()

1. Given the code snippet below, which of the following statements are true? (choose 2)

@PersistenceContext

**private** EntityManager entityManager;

**public** **void** updateUserProfile(Long usrId) {

User usr = entityManager.find(User.**class**, usrId); // Line 1

usr.setName("JavaChamp"); // Line 2

entityManager.merge(usr); // Line 3

entityManager.persist(usr); // Line 4

}

**2 At line 1 the user instance is a managed Entity**

3 At line 1 the user instance is a detached Entity

**4 At line 1 the find method will return null if the input userID is not found**

5 line 3 is mandatory to synchronize the changes done at line 2 to the database

6 line 4 is mandatory to synchronize the changes done at line 2 to the database

1. What is considered optional while you define a persistence unit? (choose 2)

2 persistence.xml configuration file

3 Entity classes annotated with @Entity

**4 The ORM definition (object relational mapping)**

**5 entity manager configuration (e.g. toplink, hibernate)**

1. How to declare JPA Entity class? (choose 3)

2 Write a private final Entity class named after the mapped table

**3 Annotate the class with @Entity**

**4 Annotate the primary key attribute or its getter method with @Id**

**5 Define private attributes to map the table columns**

6 Write the finalize method

1. What are the possible state transitions for an Entity instance ? (choose 2)

**2 from managed to detached**

3 from detached to removed

4 from detached to new

5 from new to detached

**6 from managed to removed**

1. which of the following is NOT a jpa entity manager method?

2 persist

3 flush

4 contains

**5 detach**

6 find

1. You want to write a JPA Entity class to model a databank table named COOPERATE\_USER. Each user is uniquely identified in this table by his or her social security number SSN. Along with SSN the system keeps user name, job, address and birth date.How to declare such JPA Entity class?(choose 4)

**2 Write a CooperateUser public class**

**3 Annotate the CooperateUser class with @Table(name=" COOPERATE\_USER")**

**4 Annotate the CooperateUser class with @Entity**

5 The CooperateUser class must implement Serializable

**6 Define private attributes ssn, job, name, addess and birthdate**

1. Given the code snippet below, which of the following statements are true? (choose 2)

@PersistenceContext

**private** EntityManager entityManager;

**public** **void** registerUser() {

User usr = **new** User(); // Line 1

usr.setName("JavaChamp"); // Line 2

entityManager.persist(usr); // Line 3

}

2 At line 1 the user instance is a managed Entity

**3 At line 1 the user instance is a new Entity**

4 At line 1 a new record will be inserted in the user table

5 At line 2 the entityManager will immediately reflect user name change to the database

**6 At line 3 the entityManager will immediately reflect user name change to the database**

1. What causes the entity manager to move an Entity instance from the managed to the detached state?

2 Invoke the detach() method of the entity manager on the instance

3 Invoke the remove() method of the entity manager on the instance

4 Invoke the unsynchronize() method of the entity manager on the instance

**5 when the associated persistence context ends at the end of the transaction boundary.**

1. What is the impact of marking an attribute in a JPA Entity class with @Transient annotation?

2 This attribute will should not be serialized by the EJB container

**3 This attribute will should not be persisted by the EJB container**

4 This attribute will should not be garbage collected by the EJB container