**Q1.** Which DataSource matches the following description ?                                    
  
"You want to acquire the connection only once and hold on to it.   
You should call the destroy() method when you are done with database processing."

**1)**. AbstractDataSource

**2)**. **SingleConnectionDataSource**

**3)**. **BasicDataSource**

**4)**. DriverManagerDataSource

**Q2.** To use repository from Sping Data, create an interface to extend following interface

**1)**. **org.springframework.data.repository.Repository(T,ID extends Serializable>**

**2)**. **org.springframework.data.repository.Repository(ID extends Serializable,T>**

**3)**. org.springframework.data.repository.Repository(<T >)

**4)**. 4) none of the above.

**Q3.**Classes annotated with \_\_\_\_\_\_\_\_\_\_ are identified as persistent classes by the annotatedClasses property of the AnnotationSessionFactoryBean.

**1)**. **@Persist**

**2)**. @Mapping

**3)**. @Repository

**4)**. **@Entity**

**Q4.** Spring provides a class \_\_\_\_\_\_\_\_\_\_\_\_\_ to  
work with JPA specific transactions.

**1)**. org.springframework.transaction.PlatformTransactionManager

**2)**. **org.springframework.orm.jpa.JpaTransactionManager**

**3)**. **all of the above**

**4)**. none of the above

**Q5.** How many ways you can acquire DataSource?

**1)**. **DriverManagerDataSource**

**2)**. **JNDI Look upManagerDataSource**

**3)**. all of the above

**Q6.** <bean id="sessionFactory"  
class="\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_">  
       <property name="dataSource" ref="dataSource" />  
       <property name="annotatedClasses" value="com.capgemini.entity" />  
       <property name="hibernateProperties">  
            <props>  
                   <prop key="dialect"> org.hibernate.dialect.Oracle9Dialect </prop>  
            </props>  
       </property>  
</bean>  
  
Assuming that the fully qualified class name is give, which of the following class could go into the blank?

**1)**. LocalSessionFactoryBean

**2)**. RemoteSessionFactoryBean

**3)**. SessionFactoryBean

**4)**. **AnnotationSessionFactoryBean**

**Q7.**Classes annotated with the following annotations are scanned by Spring's <context:component-scan>.  
  
1. @Controller  
2. @BeanComponent  
3. @Service  
4. @Component  
5. @Repository  
6. @Database

**1)**. 1,2,4

**2)**. 1,4

**3)**. 1,4,5

**4)**. **1,3,4,5**

**5)**. **1,4,5,6**

**Q8.** Spring’s core transaction management abstraction is based on the interface?

**1)**. PlatformTransaction

**2)**. **PlatformTransactionManager**

**3)**. **TransactionManager**

**4)**. PlatformManager

**Q9.** Container-managed and application-managed entity managers are associated with an EntityManagerFactory instance.

**1)**. **TRUE**

**2)**. **FALSE**

**Q10.** Given the code snippet below, which of the following statements are true?  
   @PersistenceContext  
   private EntityManager entityManager;  
   public void updateUserProfile(Long usrId) {  
   User usr = entityManager.find(User.class,usrId); // Line 1  
   usr.setName(“john”); // Line 2  
   entityManager.merge(usr); // Line 3  
   entityManager.persist(usr); // Line 4  
   }

**1)**. At line 1 the user instance is a managed Entity (t)

**2)**. **At line 1 the user instance is a detached Entity**

**3)**. **At line 1 the find method will return null if the input userID is not found (t)**

**4)**. line 3 is mandatory to synchronize the changes done at line 2 to the database

**5)**. **) line 4 is mandatory to synchronize the changes done at line 2 to the database**