CS544
Enterprise Architecture
Exam 1 Sample

Name_____

Student ID _____

NOTE: This material is private and confidential. It is the property of MUM and is not to be disseminated.

1. [15 points] **Circle w**hich of the following is TRUE/FALSE concerning Spring Inversion of Control/Dependency Injection:

    T F   Only Managed Beans can be injected in Spring, a POJO or JavaBean cannot.
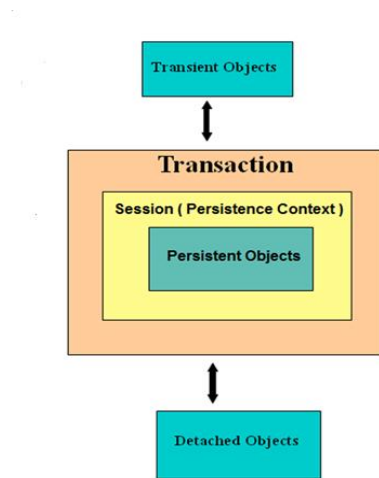
    EXPLAIN:_____.

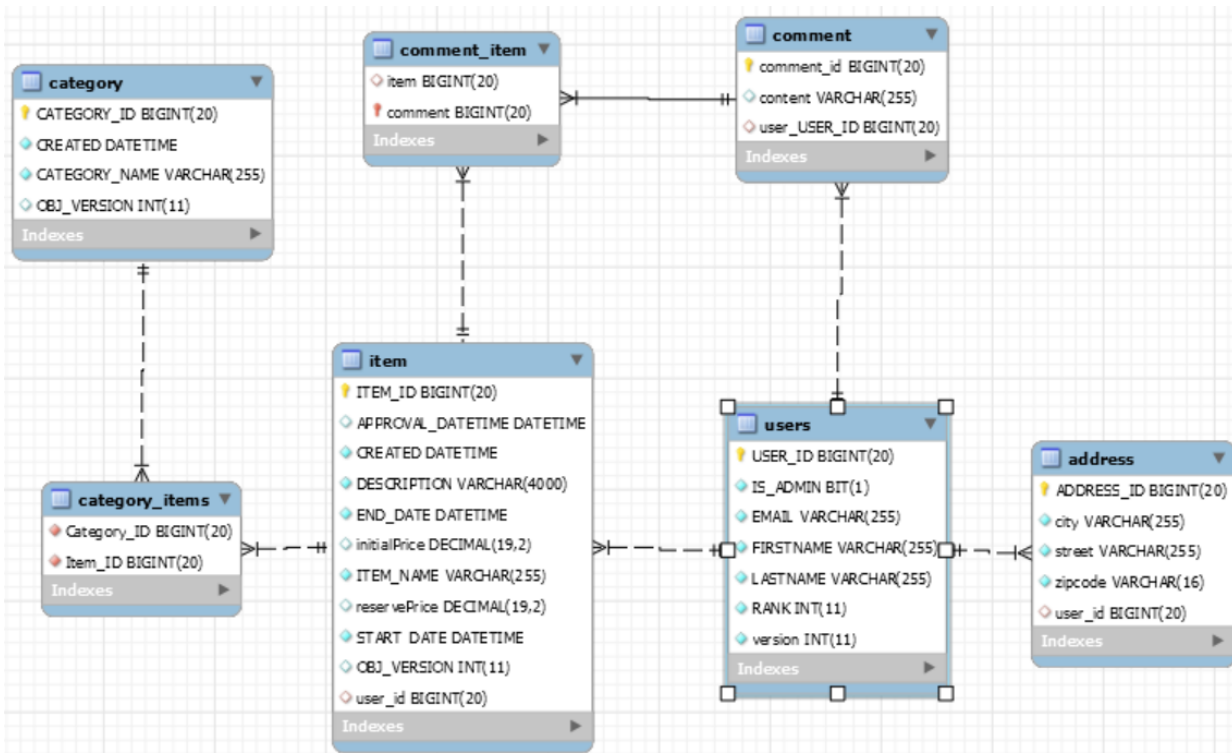   T   F   @Autowired works only on interfaces. It cannot work directly on classes.

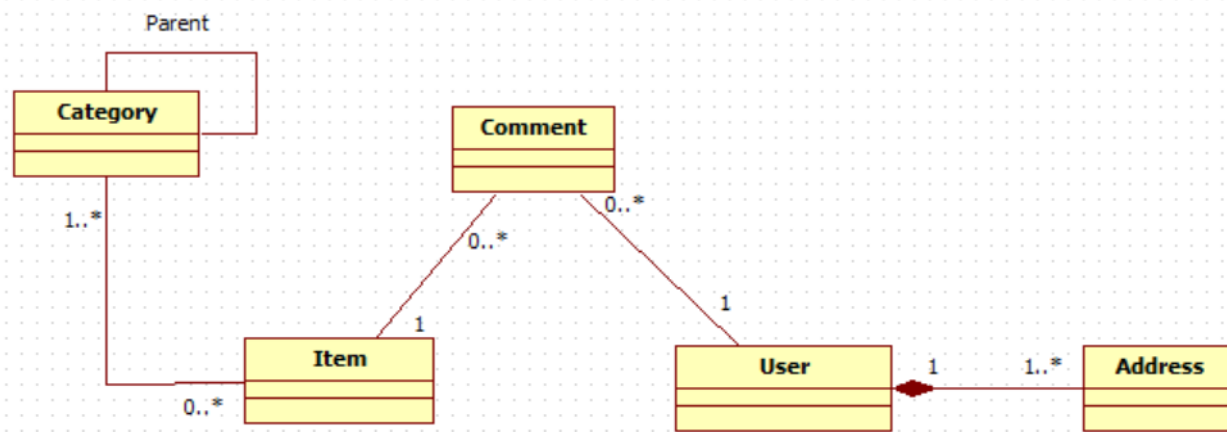   EXPLAIN:_____.

    NOTE: Usually 5 T/F staements n exam.

2. [15 points] In JPA, the Persistence Context plays an important role in the implementation of an ORM. Explain, by example the Entity lifecycle as it pertains to the following drawing.

3. **[20 points]** Annotate the Domain Objects based on the Domain Model and Entity Relationship Diagram provided. NOTE: All the Domain Objects are not listed. All the fields are not listed. Only annotate the objects and fields that are listed.

Parent

Category

Comment

0..*

1..*

0..*

1

1

Item

User

1

1..*

Address

0..*

---

category
- CATEGORY_ID BIGINT(20)
- CREATED DATETIME
- CATEGORY_NAME VARCHAR(255)
- OBJ_VERSION INT(11)

Indexes

comment_item
- item BIGINT(20)
- comment BIGINT(20)

Indexes

comment
- comment_id BIGINT(20)
- content VARCHAR(255)
- user_USER_ID BIGINT(20)

Indexes

item
- ITEM_ID BIGINT(20)
- APPROVAL_DATETIME DATETIME
- CREATED DATETIME
- DESCRIPTION VARCHAR(4000)
- END_DATE DATETIME
- initialPrice DECIMAL(19,2)
- ITEM_NAME VARCHAR(255)
- reservePrice DECIMAL(19,2)
- START DATE DATETIME
- OBJ_VERSION INT(11)
- user_id BIGINT(20)

Indexes

category_items
- Category_ID BIGINT(20)
- Item_ID BIGINT(20)

Indexes

users
- USER_ID BIGINT(20)
- IS_ADMIN BIT(1)
- EMAIL VARCHAR(255)
- FIRSTNAME VARCHAR(255)
- LASTNAME VARCHAR(255)
- RANK INT(11)
- version INT(11)

Indexes

address
- ADDRESS_ID BIGINT(20)
- city VARCHAR(255)
- street VARCHAR(255)
- zipcode VARCHAR(16)
- user_id BIGINT(20)

Indexes

```java
public class User {                          public class Comment {

    private Long id = null;                      private Long id = null;


     private int version = 0;                  private User user;


     private String firstName;                 private Item item;


     private String lastName;                  private String content;


    private String email;


    private int ranking = 0;


    private boolean admin = false;


     List<Comment> comments;


     List<Address> addresses;
```

```java
public class Item   {                        public class Category {

    private String name;                         private Long id = null;

    private String description;                  private int version = 0;

    private BigDecimal reservePrice;             private String name;

    private Set<Category> categories ;           private List<Item> items;

    List<Comment> comments;
```

4. [15 points] Implement a JQPL **parameterized** query that looks up a User **by email** who is selling a specific Item with an initial price greater than a specified dollar value.
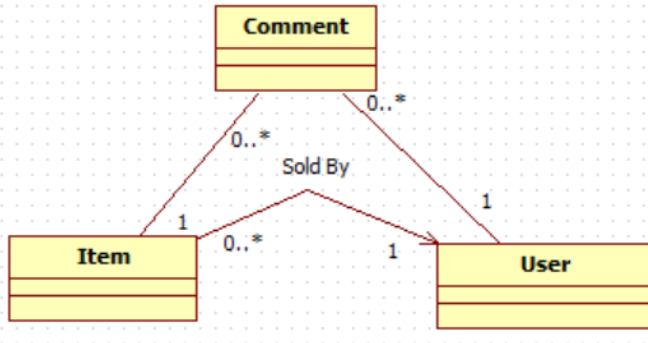For example:
Find John Smith who is selling an Item, "cardboard box" with an initial price greater than 70.00.
   Another example:
Find Will Henry who is selling an Item named "Pencil Set" with an initial price greater than 100.00.
   Item – User relationship [See relevant class properties in previous problem]:


```
In Item.java:
        @ManyToOne(fetch=FetchType.LAZY)
        @JoinColumn (name="user_id")
        private User seller;
```

Remember the Query is a ***parameterized query***. Also identify all the classes in the specific packages that need to be modified to implement the query in accordance with the N-Tier architecture convention. Describe the "pattern" that exists at the persistence layer.

ANSWER: