**CS544**

**Enterprise Architecture**

## Exam 1 July 2017

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 ORM technologies:

T F An example of impedance mismatch is the fact that a RDB puts information in rows and an OO

language puts information in Objects

EXPLAIN:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

T F A good use case for using an ORM is complex interactions between entities

EXPLAIN:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

T F The value of a good ORM is that it automatically takes care of **all** the issues relating to a RDB

EXPLAIN:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

T F JPA is an industry standard for ORMs and has made Hibernate obsolete.

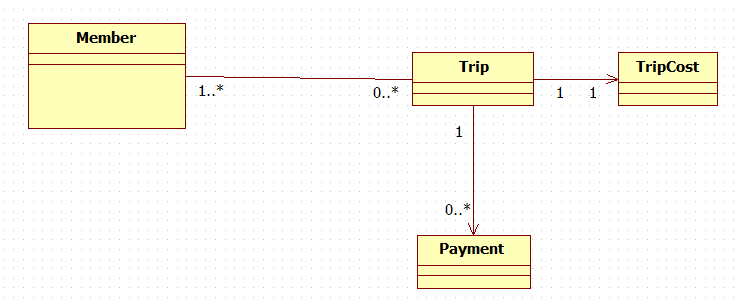
EXPLAIN:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

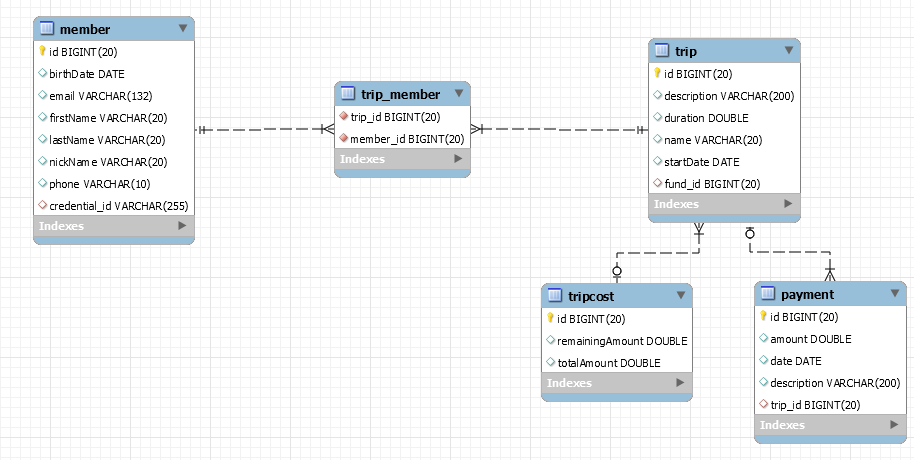
T F Native SQL Queries are supported by a good ORM solution and are recommended as the first

choice in accessing entity relationships.

EXPLAIN:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. [20 points] Annotate the Domain Objects based on the Domain Model and Entity Relationship Diagram provided. NOTE: All the fields are not listed. Only annotate the fields that are listed.





**Member.java**

**public** **class** Member {

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

**private** String firstName;

**private** String lastName;

**private** String nickName;

**private** Gender gender;

**private** String email;

**private** String phone;

**private** Date birthDate;

**private** List<Trip> trips;

**Trip.java**

**public** **class** Trip {

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

**private** String name;

**private** String description;

**private** Double duration;

**private** Date startDate;

**private** Date endDate;

List<Member> members;

Fund fund;

List<Payment> payments;

**Fund.java Payment.java**

**public** **class** Fund { **public** **class** Payment {

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

**private** Double remainingAmount; **private** String description;

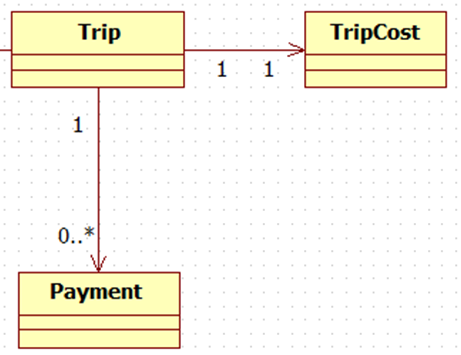
**private** Double totalAmount; **private** Double amount;

**private** Date date;

1. [15 points] Transaction management is an important part of RDBMS oriented enterprise applications Spring provides core functionality to assist in transaction management. Describe the Spring transaction functionality, how it is implemented, how it facilitates ORM Transaction management. Include an explanation on how it supports the RDBMS ACID properties of Consistency and Isolation. Be specific. Give Examples.

1. [15 points] ] For the following relationships implement a Batch fetch of all Trips with their Payments collection.

Assume the Payment collection is fetch LAZILY.



What performance problem[s] does the batch fetch address?

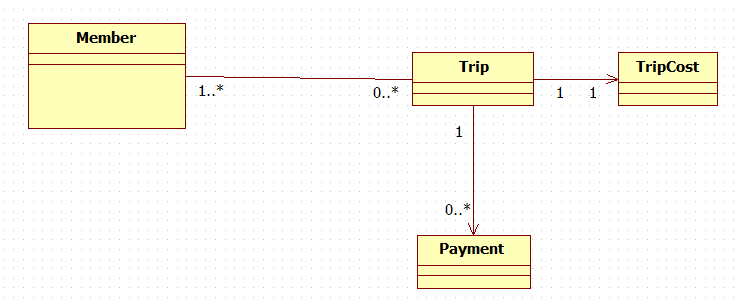
How does it work?

Explain the “algorithm” based on a universe of 20 Trips each with a collection of 5-10 Payments.

1. [15 points] IoC and DI are part of the Spring Core Technologies. Explain in detail what they are and how they work. Explain it in terms of the “Essence of a Spring Application” and the basic component s that make up a Spring Application.
2. [15 points] Implement a parameterized JQPL query with this signature:

**public** Member findByEmailAndTotalCost(String email,Double amount)

The query looks up all Members[s] by email that has a Trip associated with it that has a Trip Cost greater than the supplied amount value. Refer to Problem #2 for field names.



The Query should be a parameterized query. Also show the modifications to all classes in order to adhere to the N-Tier architecture convention. Identify the specific packages that each modified class is in.