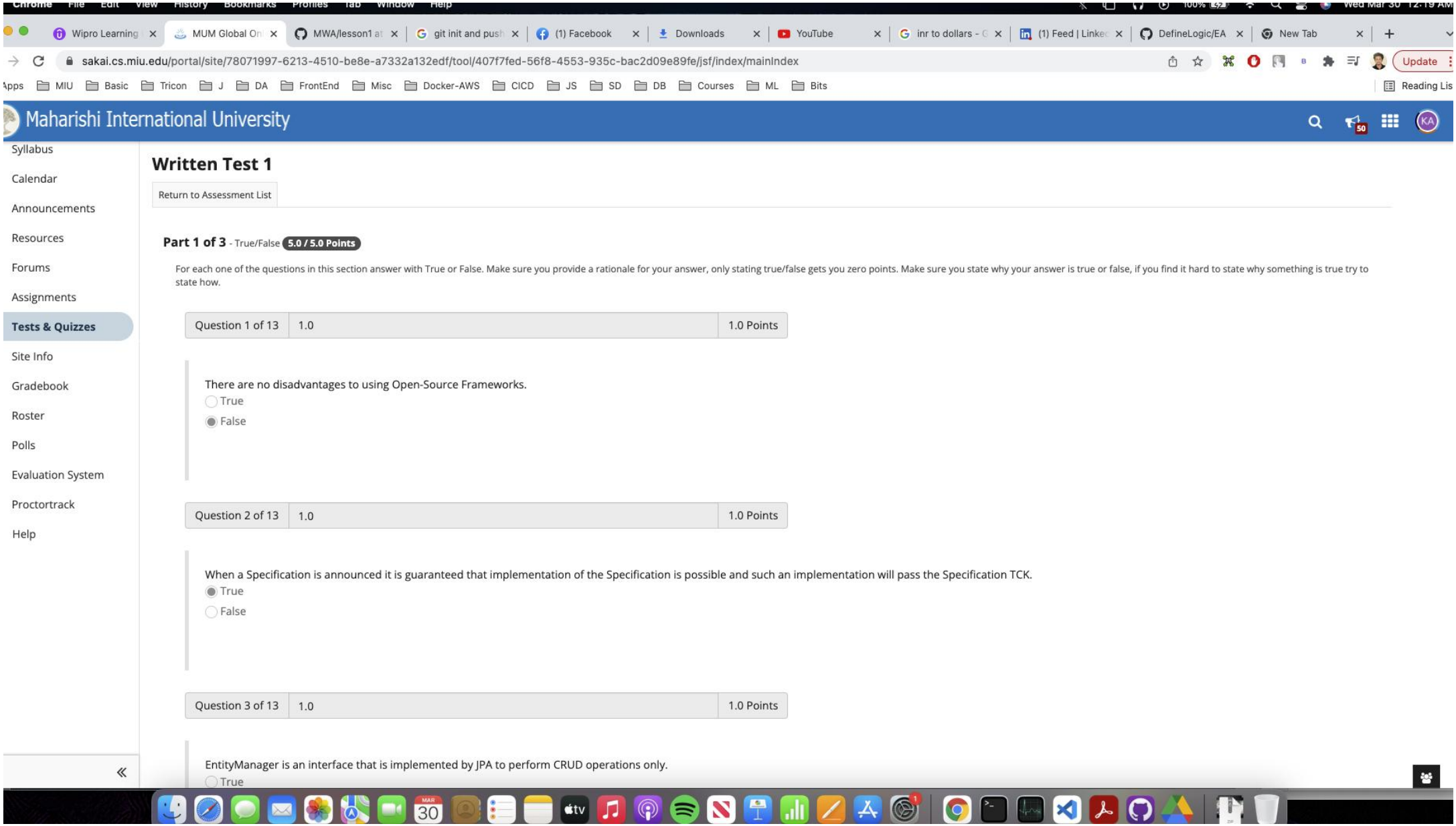
Maharishi International University
Syllabus
Calendar
Announcements
Resources
Forums
Assignments
Tests & Quizzes
Site Info
Gradebook
Roster
Polls
Evaluation System
Proctortrack
Help
Written Test 1
Return to Assessment List
Part 1 of 3 - True/False 5.0 / 5.0 Points
For each one of the questions in this section answer with True or False. Make sure you provide a rationale for your answer, only stating true/false gets you zero points. Make sure you state why your answer is true or false, if you find it hard to state why something is true try to state how.
Question 1 of 13 1.0 1.0 Points
There are no disadvantages to using Open-Source Frameworks.
True
False
Question 2 of 13 1.0 1.0 Points
When a Specification is announced it is guaranteed that implementation of the Specification is possible and such an implementation will pass the Specification TCK.
True
False
Question 3 of 13 1.0 1.0 Points
EntityManager is an interface that is implemented by JPA to perform CRUD operations only.
True

Question 3 of 13	1.0	1.0 Points

EntityManager is an interface that is implemented by JPA to perform CRUD operations only.

- ○ True
- ◉ False

Question 4 of 13	1.0	1.0 Points

If an application needs more than one EntityManager(em) instance the application needs to create more than one EntityManagerFactory (emf), one emf for each em.

- ○ True
- ◉ False

Question 5 of 13	1.0	1.0 Points

It is possible to migrate a system that is not using an ORM to use an ORM gradually thanks to JPA. Meaning, we do not have to convert everything at once and do it by parts.

- ◉ True
- ○ False

Part 2 of 3 - 4.6 / 5.0 Points

Question 6 of 13	1.9	1.0 Points

Part 2 of 3 - 4.6 / 5.0 Points

Question 6 of 13	1.9	1.0 Points

State two advantages and two disadvantages of using JPA.
Two Advantages of JPA:
1> JPA queries like JPQL and Criteria API are not DB specific , so incase we need to change the type of DB eg: from mysql to oracle we dont need to change the queries which are being used in the application. Suppose our DB is not a free software and if its license is ending and we dont want to pay for the software furthermore we can use any other free DB without having problem to change the code in our application. We just need to change the DB type in our configuration.
2> JPA implementations provides the caching mechanisms like shared cache , persistance cache which helps to improve the performance of the application.

Two DisAdvantages of JPA:

1> The performance of the queries might be slow because the conversion of code from JPA to DB specific queries requires additional computational time resulting more response time for the web application.

2> Use of the locking mechanisms in JPA not in a perfect way may result in deadlock causing the application to be not responsive.

Answer to true for False:
1>Open source software disadvantage is it have slow customer support systems. Only proprietery softwares invests in customer support systems. Also chances of open source being out of market and providing no backward compatibility is high.

2> One of the requirement for the specification is it needs to have a reference implemenation eg: JPA has eclipselink . which is a proof that the specification can be implemented so other implementors can have a guarentee it can be done.

3> Entity manager performs other operations also like beginning and ending transcations, provides 1st level of cache, provides methods like em.refresh() and em.flush to sync the DB and application etc. SO em is just not for performing CRUD operations.

4>We dont need to create EntityManagerFactory more than once for creating differnet entityManager . Since it is a factory interface which provides implementations which provides as many entity manager as we want by simplying calling the createEntityManager method.

5> It is possible to convert the JPA application gradually to ORM as JPA supports for native queries and named native queries which are nothing but plain SQL queires.

Comment:Switching DBs is thanks to JDBC not JPA.

Question 7 of 13	1.0	1.0 Points

Question 7 of 13	1.0	1.0 Points

One of the rules for an Entity is that it must have a default (empty) constructor. Why?
An entity must have a default(empty) constructor so that the jpa can create a new instance by using java reflection. If there is an arg constructor then only it requires the empty constructor since java doesn't create it by default.
Java reflection is used during the runtime by the framework to surpass the access modifiers as well as change other runtime behavioiurs.

Question 8 of 13	0.2	1.0 Points

When implementing inheritance using Joined Strategy each entity is in its own table, but we still have a discriminator. Why?
Joined strategy requiring discriminator is actually optional in JPA specfication. So the different implemenations can or cann't have a discriminator column. One of the advantage of having a discriminator is we can query the number of child classes types without using joins to other tables which provides better performance behviour.

Comment:I do not think the discriminator is optional for the Join Strategy. Please correct me if I am wrong.

Question 9 of 13	0.5	1.0 Points

What is the software engineering principle that DI (Dependency Injection) is implementing? Do not just state the principle explain a bit.
What is the value of using DI?
DI is implementing the engineering principle called Inversion Of Control. Inversion of control is the engineering principle in which the control of the flow or behvaiours of the software is given to the framework instead of User.
Dependency Injection is used by the spring framework where instead of creating the depending objects in the code , spring handles its creation using the constructor or field injections. Thus enabling us to achieve programming to interface. Programming to interface helps to achieve loose coupling.

Comment:What is DI and what does it do?

Question 10 of 13	1.0	1.0 Points

JPA cascade options are: PERSIST, MERGE, REMOVE, REFRESH, and DETACH. Why is there no cascade option for READ?

Question 10 of 13	1.0	1.0 Points

JPA cascade options are: PERSIST, MERGE, REMOVE, REFRESH, and DETACH. Why is there no cascade option for READ?
Option CascadeType.PERSIST refers to the behaviour where if one entity is saved using persist or merged than the related entity should also be saved. For Reading operation instead of CASCADE types we have separate option called Loading behaviour which can be eager or lazy.
for OneToOne and ManyToOne default is eager loading and for OneTOmany and ManyToMany default is lazy loading.

Part 3 of 3 - Architecture **2.95 / 5.0 Points**

Given each of the following scenarios identify the proper Architecture (Good) and the worst Architecture (Bad), and an architecture that is better than the bad one but not as great as the good one (Ugly). Make sure you state why an option is Good, Ugly, and Bad.

Question 11 of 13	0.2	2.0 Points

You are asked to lead a team to develop the ORM implementation of a bank system. The bank informed your team that currently, it has two kinds of Bank Accounts (Checking and Saving) with more than 250 million accounts. There are talks that in the near future a new type of account will be added (CD), and a year from now retirement accounts may be added.
Three team members have proposed different inheritance solutions:
1- Implement this using the Single-Table strategy.
2- Implement this using Joined strategy.
3- Implement this using Table-per-Concrete Class strategy.

Good: Option 1
Single Table is good as it prevents the join cost . And the writing operation is fast.

Ugly: Option 2
Bad because joins is costly , writing operion is costly.

Bad: Option 3

Union operation is costly.

Comment:But the bank is not telling us they have performance problems yet. They are expressing the possibility of expansions.

Question 12 of 13	0.75	1.0 Points

You are asked to lead a team to develop the ORM migration of a company from JDBC to JPA. You are tasked with migrating a major query function. The query in question is very complex and required expert SQL developers to be involved in optimizing it. The query involves more than 20 tables. The functionality using this query is critical for the company's success. This task needs to be completed as soon as possible. While this function is very critical, it is only executed once a day.
Three team members have proposed different inheritance solutions:
1- Implement this using Native Query and deliver it within the next sprint (2 weeks). Then create a task to switch to Criteria API after thorough testing (this will take at least 3 months).
2- Implement this using Dynamic Queries (JPQL) and replace tables with entities (could be done in 2 weeks).
3- Similar to option 1 but use NamedNative queries.
Good: Option 3

We can easily convert to any NamedNative query and after rigorous testing for 3 months to Criteria API we can finalize on the query. NamedNative also provides some syntax checking as the server boots up. And also Criteria API helps to simplify complex queries into functions easily.

Ugly: Option1

As good as first one except the syntax checking.

Bad: Option 2

It would work fine if the jpql written is perfect but since it is complex and no testing is involved and the query is critical to the organization it is not a good practice to finalize withing lesser time. If wrong it could lead to hazardous situation .

Question 13 of 13	2.0	2.0 Points

On a social media website, we wish to implement a message posting feature. Our clients with accounts on the website are able to post messages. A user can post one message at a time. One account can submit messages on an account page. The message is stored in a Message Entity.
Three team members have proposed different solutions:
1- Add Pessimistic Locking to Message adding code at the em level.

It would work fine if the jpql written is perfect but since it is complex and no testing is involved and the query is critical to the organization it is not a good practice to finalize withing lesser time. If wrong it could lead to hazardous situation .

Question 13 of 13	2.0	2.0 Points

On a social media website, we wish to implement a message posting feature. Our clients with accounts on the website are able to post messages. A user can post one message at a time. One account can submit messages on an account page. The message is stored in a Message Entity.
Three team members have proposed different solutions:
1- Add Pessimistic Locking to Message adding code at the em level.
2- Add Optimistic Locking to Message by adding @version to MEssage Entity.
3- There is no need to do anything since only one account is posting a message at a time. There is no chance for collisions.
By requirement I am assuming each user can have individual account row in DB table. And many messages in the Message table is having referenced to the Account Table records.

Good: Option 2
A user posting two messages at the same time from 2 devices is very low but not impossible. So we can return them with OptimisticLockException to try again.

Ugly: Option 1
Pessimistic locking is not so good because there is need of overhead of maintaining a lock but it will work .

Bad: Option 3
We need to do something as there may be chances of collisions .