

Student Name:- \_\_\_\_\_ Student ID: \_\_\_\_\_



**CS545: Web Application Architecture**  
**Midterm Exam**

**Computer Professionals Program**

---

**Date: 06 - 08 -2023**

Theory		Cognitive skills		
Q1 (8)	Q2 (6)	Q3-P1 (9)	Q3-P2 (14)	Q3-P3 (3)

The exam takes 150 minutes.

Please read the exam policy before you start the exam.

Do not provide more than one answer to a question. If you do so only the first one will be evaluated.

There is no tolerance policy for academic dishonesty on exams. You will be asked to leave the exam room immediately without a warning if you do or violate one of the following things which means you will get an NC.

1. You are caught cheating or trying to cheat.
2. Answers should be written with a pen or pencil, but if you want to use a pencil, please bring your own eraser and sharpener you are not allowed to borrow from other students during the exam.
3. All mobile phones should be turned off and stored with your coat or backpack. You could also place it on the instructor's desk.
4. You are not allowed to go to the restroom or go out to the room for water.
5. You are not allowed to ask or get extra papers from other students.

**Question 1: Circle the correct answer**

**(6 points – each 1)**

**1) What is the purpose of the `@Service` annotation in Spring Boot?**

- a) It defines a class as a component for data access and persistence.
- b) It defines a class as a service layer component for business logic.
- c) It defines a generic class as a bean for dependency injection.
- d) It defines a class as a presentation layer component for handling web requests.

**2) Which annotation is used to send data from JSON request to a Spring Boot method?**

- a) `@ResponseBody`
- b) `@RequestBody`
- c) `@RestController`
- d) `@RequestMapping`

**3) To get the value of `id` in the URL below:**

`http://localhost:8080/products?id=1234`

- ```
@GetMapping("/products")
public List<Product> findProductById ( ????????? ) {
    return productService.findProductById(id); }
```
- a) `@ModelAttribute("id")` long id
  - b) `@PathVariable("id")` long id
  - c) `@RequestParam("id")` long id
  - d) `@RequestBody("id")` long id

**4) Which versioning type (REST) is the following URI implementing: `example.com/v1/users/1`**

- a) URI versioning
- b) Parameter versioning
- c) Custom-header versioning
- d) Mediatype-header versioning

**5) If the cascade type was `CascadeType.REMOVE` that propagates from a parent to a child entity. When we delete the 'parent' entity, the associated entity 'child' will also be deleted.**

- a) true
- b) false

**6) What are the three parts of a JWT?**

- a) Header, Payload, Signature
- b) Payload, Signature, Body
- c) Header, Body, Encryption
- d) Signature, Encryption, Body

**7) Native queries are more susceptible to SQL injection attacks compared to parameterized queries provided by JPQL.**

- a) true
- b) false

**8) Spring's AOP (Aspect-Oriented Programming) module implements AOP during runtime.**

- a) true
- b) false

**Question 2: Short Answers - SOLVE THREE**

**(6 points – each 2)**

**1) What is the difference between accessToken and refreshToken?**

**2) In the ORM entity life cycle. What is the difference between ‘transit’ and ‘persist’. Write your answer with a brief explanation.**

**3) If fetching the data JOIN fetch mode, it eagerly pulls all the required data in one query and solves the N+1 problem. Then why do we need LAZY loading from the beginning?**

**4) List two reasons with a brief explanation on why it would be beneficial to follow a stateless authentication approach.**

### Question 3)

#### Part-1

Below are tables for an Event Coordination system. Coordinators are in charge of multiple events, and these events have a group of tasks that have to be accomplished. Create and annotate the domains based on the database tables given below, considering the following: (9 points)

- A **Coordinator** can work on one or more **Events**.
  - An **Event** can have one or more **Coordinators** working on it.
  - An **Event** can have one or more **Tasks** to get it done.
  - Every **Task** should be assigned to one **Event**.
  - A **Coordinator** can have one **Address** and vice versa.
- All associations should be bi-directional
  - All primary keys should be auto generated.
  - Set JPA cascade operations as follows:
    - Any operation applied to the **Coordinator** should also apply to the **Address**
    - If an **Event** is deleted, it should delete all its **Tasks** assigned to it.
    - When retrieving an **Event** as a child association, it should load all its **Tasks** in one query and the fetch type should remain LAZY from **Event** to **Task**.
- (Bonus)

#### Event

| event_id | title              | state |
|----------|--------------------|-------|
| 1        | Conference         | IA    |
| 2        | Faculty meeting    | IA    |
| 3        | Technical workshop | FL    |
| 4        | Seminar            | TX    |

#### Task

| task_id | description               | event_id |
|---------|---------------------------|----------|
| 1       | Prepare room setup        | 1        |
| 2       | Confirm with participants | 1        |
| 3       | Prepare presentation      | 4        |
| 4       | Purchase refreshments     | 2        |
| 5       | Prepare roster            | 3        |
| 6       | Send invitations          | 2        |

#### Coordinator

| co_id | name    | gender | address_id |
|-------|---------|--------|------------|
| 1     | Dean    | male   | 3          |
| 2     | Yasmeen | female | 2          |
| 3     | Mira    | female | 1          |
| 4     | Zaineh  | female | 4          |

#### Coordinators Events

| co_id | event_id |
|-------|----------|
| 1     | 1        |
| 1     | 2        |
| 1     | 3        |
| 2     | 2        |
| 2     | 3        |
| 3     | 3        |
| 4     | 4        |
| 4     | 2        |

#### Address

| address_id | country       | city      | state | zipcode |
|------------|---------------|-----------|-------|---------|
| 1          | United States | Orlando   | FL    | 14565   |
| 2          | United States | Fairfield | IA    | 52556   |
| 3          | United States | Orlando   | FL    | 32832   |
| 4          | United States | Dallas    | TX    | 11234   |

## Part-2

Create RESTful web services for the Coordinator domain by following the n-tier architecture and implement the following requirements with best practices: (14 points)

- Create an endpoint for each CRUD operations – (findAll, findById, deleteById, save, update).
- Create an endpoint that returns the **Events** for a specific **Coordinator**.  
*For example: api/v1/coordinators/2/events → Output: events 2 and 3*
- Create an endpoint that returns the **Tasks** for a specific **Coordinator**.  
*For example: api/v1/coordinators/2/tasks → Output: tasks 4, 5, and 6*
- Create an endpoint that will retrieve all **Coordinators** that have events out of the state they live in.  
→ Output: coordinators 1, 2, 4
- Create an endpoint that will search all **Coordinators** based on the following criteria: ( % name % , = gender ). It must satisfy according to given criteria (AND). The returned value should be a list of **Coordinators**. This query should work dynamically according to any provided properties.  
*For example: api/v1/coordinators/filter?gender=female → Output: coordinators 2, 3, and 4*  
*For example: api/v1/coordinators/filter?gender=female&name=n → Output: coordinators 2 and 4*

**Note:** You may focus on the Controller and Repository implementations. You may add anything specific in the service layer if you are willing to add some business logic other than the usual service calls.

**Note:** If you need anything from the another domain, you may assume the [Entity]Service is completely functional according to your requirement. Just add an assumption.

---

## Part-3

Create an Aspect class that will have an annotation called @Confirm that will send a notification after any delete and update method on the CoordinatorController. You may use the following implementation for the advice. (4 points)

```
public void confirmation( Joinpoint joinpoint ){  
    // implementation to send confirmation email..  
}
```

**Note:** You do not have to fill out the implementation to send confirmation... . Assume it will do the required functionality.