DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
THIRD SESSIONAL
SUBJECT: (CE424) JAVA TECHNOLOGIES

Examination	**: B.Tech Semester IV**	**Seat No.**	: 82
Date	**: 21 / 03 / 2024**	**Day**	**: THURSDAY**
Time	**: 1:00 PM TO 2:15 PM**	**Max. Marks**	: 36

INSTRUCTIONS:

1. Figures to the right indicate maximum marks for that question.
2. The symbols used carry their usual meanings.
3. Assume suitable data, if required & mention them clearly.
4. Draw neat sketches wherever necessary.

Q.1 Do as directed. **[12]**

CO4-R (a) Which are the 3 annotations combined by @SpringBootApplication annotation? [2]

CO4-N (b) Match the followings: [2]

Annotation for Class	**Purpose / Usage**
1. @Service	a. Typically used for database operations.
2. @Controller	b. Typically used for mapping HTTP requests to specific handler methods.
3. @Repository	c. Typically used for implementing business logic.
4. @RequestMapping	d. Typically used to define the behavior of web endpoints in a Spring Boot application.

CO4-A (c) Write a pointcut expression for getter methods of any class in com.example.dao package. [2]

CO4-R (d) Draw the state transition diagram of the hibernate Entity object. [2]

CO4-U (e) i) Enlist any two ORM mismatch. [2]
ii) Enlist any two inheritance Strategies in hibernate.

CO4-C (f) Consider the following list is created using JSTL core tags. [2]
<c:set var="names" value="${['Ram', 'Bharat', 'Sita', 'Laxman']}" />
Write a code snippet using JSTL core tag and EL to display the list **"names"**.

Q.2 Attempt *Any TWO* from the following: **[12]**

CO4-R (a) 1. What is advice and what are the different types of advices supported by Spring AOP? [1]
2. In one line explain what is the annotation for each advice and when will each advice be executed? [5]

CO4-A (b) Write Spring boot application code for REST API to perform basic CRUD operations on Student entity. Write following components: **RestController and Repository by extending JpaRepository**. Assume that Student Entity has 2 properties, ID of type integer and Name of type String along with getters/setters/constructors and other related methods. [6]

CO4-U (c) Explain each line of the following code: [6]

```
@Configuration
public class DemoSecurityConfig {
@Bean
public UserDetailsManager  userDetailsManager(DataSource dataSource) {
JdbcUserDetailsManager jdbcUserDetailsManager = new
                                JdbcUserDetailsManager(dataSource);

jdbcUserDetailsManager.setUsersByUsernameQuery(
"select user_id, pw, active from members where user_id=?");

jdbcUserDetailsManager.setAuthoritiesByUsernameQuery(
"select user_id, role from roles where user_id=?");

return jdbcUserDetailsManager;
}
@Bean
public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
http.authorizeHttpRequests(configurer -> configurer
.requestMatchers(HttpMethod.GET, "/api/employees").hasRole("EMPLOYEE")
.requestMatchers(HttpMethod.GET, "/api/employees/**").hasRole("EMPLOYEE")
.requestMatchers(HttpMethod.POST, "/api/employees").hasRole("MANAGER")
.requestMatchers(HttpMethod.PUT, "/api/employees").hasRole("MANAGER")
.requestMatchers(HttpMethod.DELETE, "/api/employees/**").hasRole("ADMIN")
);
http.httpBasic(Customizer.withDefaults());
http.csrf(csrf -> csrf.disable());
return http.build();
}}
```

Q.3 Attempt the following: [12]

CO4-A (a) Annotate the following class to define the class as an Entity class by considering the points given below. [6]

i)The **Mobile_no** must be an auto-increment and primary key column in the table.

ii)The **Price** must not be persisted into the database table.

iii) The **Model_name** must be of length 100 characters.

iv)The **Manufactured_date** must store only **DATE** in the database table.

```
    class Mobile    {
            private int Mobile_no;
            private String Model_name;
            private int Price;
            private Date Manufactured_date;
    }
```

CO4-C (b) Consider the given Entity class and answer the following questions. [6]

i) Write a hibernate query to display the names of all the regular customers.

ii) Write a hibernate query to update the customer name for a customer with cust_id=1.

iii) Write a hibernate query to display the highest amount spent by the customer.

```
@Entity
class Customer
{   @Id
    private int cust_id;
    private String cust_name;
    private String cust_type;     //cust_type=regular/irregular
    private int amount;
}
```

Note:- Assume hibernate.cfg.xml containing the configurations is available.

OR

Q.3 Attempt the following: [12]

CO4-A (a) The social media application requires storing the details of users and their posts using the Hibernate framework. Use the necessary annotations to define the given entity classes with bidirectional relationships in Hibernate and it is important to ensure that only two tables are generated for association mapping. [6]

class User
{ private int user_id;
private String User_name;
private List<Post> posts;
}
	class Post
	{ private int post_id;
	private String post_details;
	private User posted_by;
	}

CO4-C (b) Develop a Hibernate application to perform the operation as mentioned below. Consider the Entity class User and Post from the **Q-3 (a).** [6]

i) Implement a method addPost() to save the post details into the database.

ii) Implement the method display() to display all the posts created by the given user.

Note:- Assume hibernate.cfg.xml containing the configurations is available.

Bloom's Taxonomy levels: R-Remembering, U- Understanding, A-Applying, N-Analyzing, E- Evaluating, C-Creating