

CS489 – Applied Software Development

Final Exam: Structure and Review Points

The Final Exam will consist of questions given in 2 parts, as follows:

Part 1: Theory – This part will be composed of Knowledge-based questions, including Short answer questions, True/False questions (some will require given a rationale), Multiple-Choice questions and some short/small coding questions.

Part 2: Software Development skills – Application Coding / Problem solving

This part will be composed of question(s) requiring Object-oriented Software requirements analysis, domain modeling, design and implementation tasks.

Part 1: Not more than 40%

Part 2: At least 60%

The Exam will be all computer-based. i.e. You will use computers for both parts 1 and 2. The exam duration will be 2.5 hours (timed from 10 am to 12:30 pm).

The following are the relevant lessons/topics and sample questions:

Lesson 6: Data Persistence:

1. What is Data persistence?
Data persistence is a mechanism for storing data into non-volatile storage so that it can outlive the lifespan of the software that generated the data. Having persistence, data ensure that the data remains accessible, reliable and interact even beyond the execution of the system.
2. Database - Relational Databases, SQL Databases and No-SQL Databases
3. JDBC – Java Database Connectivity API
Java SE module - java.sql
Commonly used JDBC API classes and interfaces – DriverManager, Connection, Statement (Prepared Statement, Callable Statement), ResultSet, SQLException
4. Jarkata Persistence Api (JPA) – defines ORM for Enterprise Java applications
 - a. JPA Annotations
 - i. @Entity
 - ii. @Id
 - iii. @GeneratedValue – strategy: GenerationType.AUTO, GenerationType.IDENTITY, GenerationType.SEQUENCE etc.
 - iv. Data Validation
 1. Database Column Constraints
 - a. @Column(nullable=false)
 - v. Bean Validation
 1. @NotNull – null value is invalid, an empty string “” is valid.
 2. @NotEmpty – null value and empty is invalid.
 3. @NotBlank – null, empty string, whitespaces is invalid.
5. @OneToOne

6. @OneToMany and @ManyToOne
7. @ManyToMany (resulting in creating a Join Table)
8. @Transient...
etc.....
9. Hibernate–ORM framework that implements the JPA specification
Hibernate is an open-source Object-Relational Mapping (ORM) framework for implementing data persistence for enterprise applications on Java platform.
Hibernate is an implementation of the Jarkarta Persistence specification – It is a JPA provider.
10. SpringDataJPA–the dependency in Spring platform for Data persistence using JPA.

Additional Notes

Types of Data Persistence

- Unstructured file-based persistence – eg CSV
- Semi-Structured file-based persistence – eg JSON
- In memory data storage – caching
- Database Persistence – Relational Databases, SQL Databases and No-SQL Databases

ORM (Object Relational Mapping)

A technique for translating or converting data between relational database and object oriented programming language. ORM essential creates/implements a “virtual” object database layer that can be used from within the OO programming language to access data.

Lesson 7: Restful Web API Development

1. What is REST?
REST stands for REpresentational State Transfer.
It is an architectural style for Distributed Hypermedia Systems.
It is originated from the dissertation paper by Roy Fielding
It serves as the underlying mechanism on which the World-Wide-Web (WWW) operates.
2. What is Restful Web API?
RESTful Web API is one of the most common architectural styles for implementing an enterprise application that exposes or serves backend data as web service endpoints.
3. etc

Lesson 8: Full-stack Application Development

1. What is full-stack?

A full-stack web application is a software application that consists of both frontend and backend components. The frontend handles the user interface and interacts with users, while the backend manages data storage, business logic, and communication with other services or databases. Full-stack developers are proficient in both frontend and backend development, allowing them to build and maintain all aspects of a web application.

2. What is backend?

The backend, on the other hand, is the part of the application that users don't see. It consists of the server, application logic, and database, managing data and performing operations based on user requests.

3. What is frontend?

The frontend of a web application refers to the part that users interact with directly in their web browsers. It includes elements like the user interface, design, and user experience.

Lesson 9: Security

1. What is software security?

Software Security describes methodologies, frameworks, processes and strategies for enhancing security and reducing vulnerabilities.

2. What is Authentication?

Authentication is the process of verifying user's identity. Ascertaining that someone who they claim to be. (eg. userId and password)

3. What is Authorization?

Authorization is mechanism for establishing access-level and granting appropriate rights and privileges.

4. Spring Security – offers a highly customizable authentication and access-control framework for spring-based applications.

Key Concepts in Spring Security

- Spring Security Filter Chain – for Servlet-Driven spring web applications, the filter chain plays the central role for securing given HTTP request URL Path.
- Spring Boot Auto Configuration – simply adding dependency named “spring-boot-starter-security”, enables default config.
 - a. What are some important security types in Spring Security? (5 types)
 - HTTP Basic Authentication: enabled in HTTP server or Servlet Container
 - Form-based, custom Database Authentication – use Database to store/authenticate the user credentials.
 - LDAP-driven Authentication – eg. Microsoft Windows Active Directory Service
 - OAuth/OpenID Connect Authentication – Login With Google or Microsoft or Facebook or GitHub
 - Token-based Authentication – using JSON Web Token (JWT).

5. What is Session-based auth vs Token-based auth?

Session-based authentication stores user session information on the server side, typically using cookies, and associates it with a unique session ID. The

server keeps track of the session state, allowing users to access protected resources until the session expires or is invalidated.

Token-based authentication, on the other hand, relies on issuing tokens (like JSON Web Tokens or JWTs) to authenticated users. These tokens contain user identity and permissions and are typically stored on the client side (e.g., in local storage or cookies). Tokens are sent with each request to authenticate the user and access protected resources, eliminating the need for server-side session storage.

6. What is JWT?
JWT stands for JSON Web Token. It's a compact, URL-safe means of representing claims securely between two parties. It's commonly used for authentication and information exchange in web development.
7. Cryptography – Techniques for securing communication through data encryption.
8. Spring Security API – Authentication Manager and Authentication Provider
 - Authentication Manager – an interface with just one method “authenticate” which process an authentication request.
 - Authentication Provider – an interface that can be implemented by one or more classes and provides Authentication Service using a given authentication mechanism eg. DaoAuthenticationProvider, LdapAuthenticationProvider, JwtAuthenticationProvider etc.

Lesson 10: Container and Containerization

1. What is a container and what do you mean by containerization?
Container is a software packing mechanism which enables the encapsulation of an application together with all its dependencies into a single distributable unit. It offers lightweight solution, portability and isolation.

Containerization is the practice of packing the software code together with the operating system libraries and any libraries required to run the code, to create a single lightweight executable – called container - that runs constantly on any infrastructure.

We need containers to make the application software run correctly across multiple execution environments/infrastructures and to solve the issue of working on one machine and failing everywhere else.

2. Virtualization and What are its drawbacks?
Virtualization is the process of creating a virtual (rather than actual) version of something, such as a server, operating system, storage device, or network

resource. It allows multiple virtual instances to run on a single physical machine, enabling better resource utilization and isolation.

Drawbacks of virtualization include:

Performance Overhead: Virtualization can introduce overhead due to the emulation of hardware and the additional layer of abstraction, leading to decreased performance compared to running directly on hardware.

Resource Overhead: Each virtual machine (VM) requires a portion of the host's resources, including CPU, memory, and storage, which can lead to inefficient resource allocation and potential contention.

Complexity: Managing a virtualized environment can be complex, requiring expertise in configuring, monitoring, and troubleshooting both the virtualized infrastructure and the virtual machines themselves.

Security Concerns: Virtualization introduces additional attack vectors, such as vulnerabilities in the hypervisor or misconfigurations of VMs, which can potentially compromise the security of the entire virtualized environment.

Licensing Costs: Some virtualization technologies require licenses for both the hypervisor and guest operating systems, leading to increased licensing costs for organizations deploying virtualized environments.

3. What is Docker?

Docker is a platform that allows developers to package, distribute, and run applications within containers, which are lightweight, portable, and isolated environments. It simplifies the process of deploying software by ensuring consistency across different environments and making it easier to manage dependencies.

4. How do you use Docker to containerize your application software?

To containerize your application with Docker, you create a Dockerfile that describes your app's environment and dependencies. Then, you build an image from the Dockerfile using the `docker build` command. Once the image is built, you run a container using the `docker run` command, specifying the image you created. Docker containers isolate your app and its dependencies, making it easy to deploy and run consistently across different environments.

Additional Notes: Docker Image Creation functionality was introduced in SpringBoot version 2.3. The main dependency required for this is “Spring-Boot-Devtools”.

Containerization vs Virtualization

- In contrast to Virtualization, Containers are more portable and more resource-efficient.
- Containers are lightweight and portable because the host operating system's kernel.

Lesson 11: Software Testing

1. What makes automated testing better than manual testing?
2. Types of Testing
 - a. Unit Test – is a test that test one single class
Stay within the boundaries – component boundary and layer boundary
Deals mostly with testing business logics
 - b. Integration Test
Go outside of our boundry – call the database, send message, read a file from the disk
Often time consuming operations

Additional Notes: Test Case – test one single method, Test Class – test one single class, Test Suite – a collection of test cases

*** Assert Methods

Good Unit Test – FIRST => Fast, Isolated, Repeatable, Self-validating, Timely

3. What is Mocking?
Mocking is a technique used in software development to simulate the behavior of real components, such as modules, functions, or objects, in order to test other parts of the code. It involves creating fake versions of these components that behave in predefined ways, allowing developers to isolate and test specific parts of their code without relying on the actual implementations.
4. Mockito
Mockito is a popular Java mocking framework used for testing. It allows developers to create mock objects in unit tests, enabling them to simulate the behavior of real objects and dependencies. Mockito provides a simple and flexible API for mocking and verifying interactions with these mock objects during testing.
5. JUnit
JUnit is a popular open-source unit testing framework for Java programming language. It provides a simple and standardized way to write and execute repeatable tests for Java code. With JUnit, developers can define test cases, execute them, and verify expected behaviors, helping to ensure the reliability and correctness of their code.

Mock is all about interaction. Eg. Did the service class call correctly the DAO class?

Stub is all about state. Eg. Return some dummy data. `assertEquals(4, items.count())`

Lesson 12: Software Deployment

1. How do you deliver finished software product to end user?
After design and development is complete, it comes time to deliver the finished working software product to the consumer. In modern software development practice, Cloud-based services are more preferred option for deploying enterprise web applications.
2. Microsoft Azure Cloud Service
Azure offers a global distributed cloud computing platform for enterprise application development, hosting and management.

Note: For some of the questions/tasks in the part 2 of the Exam, you will be expected to take screenshot(s) of your work/result(s), and save them to an image file (.png or .jpg or .jpeg only) and include these in the zip file, which you will submit.

Also, when you take the screenshots, it should be of the entire computer screen (NOT a snippet or a window)

//-- The End --//