Ödev - 2

Q1 - Explain the meanings of IoC and DI.

A: IoC is a software engineering principle where object instances are managed to be more loosely coupled. It means that a framework or container will manage and/or control the objects or parts of your code that are affecting the flow of your program. IoC principle is obtained through several methods, one of them is dependency injection. Dependency Injection is a pattern that is being used to implement IoC. Dependency injection reduces the coupling between objects while increasing the abstraction of classes. Injecting an object into another is done by a constructor or a setter.

Q2 - Explain Spring Bean Scopes.

A: Spring bean scopes are used to configure Spring's behavior towards that bean during its lifetime.

Q3 - What does @SpringBootApplication do?

A: @SpringBootApplication is an annotation that triggers three features; @EnableAutoConfiguration, @ComponentScan, @Configuration which enables auto-configuration and component scan. It helps spring boot developers to build an app faster and register extra beans in the context.

Q4 - What is Spring AOP? Where and how to use it?

A: Aspect-Oriented Programming is a programming paradigm that is used to process cross-cutting concerns. AOP also increases modularity and flexibility of applications by separating cross-cutting concerns. It is used to solve problems that can occur across different modules of the program.

Q5 - What is singleton and how to use it?

A: Singleton is a spring bean scope configuration. When used on a bean scope, spring creates only single instance of that bean and all references for that bean directs to the same object. It is the default @Scope value if not mentioned otherwise.

Q6 - What is Spring Boot Actuator and where to use it?

A: Spring Boot Actuator is a spring dependency that provides production-ready endpoint features that enable programmers to monitor the health, state, and functionality of their components.

Q7 - What is primary difference between Spring and Spring Boot?

A: Spring boot is a tool that is built on top of already existing spring framework. It aims to remove or reduce boilerplate configurations that are required in a spring application. It enables even faster software development than spring does.

Q8 - What are reasons to use VCS?

A: It increases the traceability of the development process of our projects. VCS also enables the collaboration of team members and makes sure everyone is working on the same version of the project.

Q9 - What are SOLID principles? Give some sample usage in Java.

A: SOLID is an acronym for five design principles that focus on making coding more flexible, maintainable, adaptable, reusable, and sustainable. The initials in the acronym stand for; single-responsibility principle, open-closed principle, Liskov substitution principle, interface segregation principle, and dependency inversion principle.

Q10 - What is RAD model?

A: Rapid Application Development (RAD) is a programming development model and technique where the problem is divided into smaller pieces. Then those pieces can be assigned to different teams to increase quickness in development. RAD process is split into 5 phases; Business modeling, Data modeling, Process modeling, Application generation, Testing and turnover.

Q11 - What is Spring Boot starter? How is it useful?

A: Spring boot starters are dependency descriptors that bring jars to the classpath to make building easier and faster. They can be added to the **<dependencies>** section in the pom file.

Q12 - What is “Caching”? How is “Caching” achieved in Spring Boot?

A: Caching is keeping data that is either frequently used or has a chance to be used in a short span of time. It reduces the required number of requests to a database, thus it increases the performance of applications as sending requests to databases is costly. Spring Boot provides an auto-configured cache manager. It can be configured manually and further with spring annotations.

Q13 - Explain 5W of logging.

A: Logging is an act of keeping a log of events that occur in an operating system, software, or user actions. Logging should be easily readable by everyone and also parsable by computers. Irrelevant or insignificant data should not be included to ease the readability of logs. Logging is done in many scenarios, mostly when the following breakpoints occur;

- When reached to major branching points in the code
- When unexpected results or errors happen
- When significant points in business logic is processed

Q14 - What is “Swagger”? Have you implemented it using Spring Boot?

Swagger is an Interface Description Language that is built using several open-source tools. It is a tool to map RESTful APIs’ structure. It can be used to automate API documentation, code generation, and test-case generation.