HOMEWORK #2

1-) IOC and DI means ?

Inversion of Control, IoC is a design principle which recommends the inversion of different kinds of controls in object-oriented design to achieve loose coupling between application classes. Dependency Injection (DI) is a design pattern used to implement IoC. It allows the creation of dependent objects outside of a class and provides those objects to a class through different ways.

2-) Spring Bean Scopes ?

IoC, Beanlerin ne zaman ve ne şekilde oluşturulacağına karar veriyor. IoC beanleri yönetiyor. Default olarak singleton olarak oluşturuyor, tek obje oluşabiliyor.

3-) What does @SpringBootApplication do ?

Configuration, Auto configuration, component scan

Spring Boot @SpringBootApplication annotation is used to mark a configuration class that declares one or more @Bean methods and also triggers auto-configuration and component scanning. It’s same as declaring a class with @Configuration, @EnableAutoConfiguration and @ComponentScan annotations.

4-) What is Spring AOP ? Where and How to use it ?

AOP is a programming paradigm that aims to increase modularity by allowing the separation of cross-cutting concerns. It does this by adding additional behavior to existing code without modifying the code itself. Transaction management for connection to databases.

5-) What is Singleton and where to use it ?

As a bean, Only one instance will be created for a single bean definition per Spring IoC container and the same object will be shared for each request made for that bean.

As a pattern, Singleton pattern says that one and only one instance of a particular class will ever be created per classloader.

6-) What is Spring Boot Actuator and Where to use it ?

The spring-boot-actuator module provides all of Spring Boot’s production-ready features.

Monitoring our app, gathering metrics, understanding traffic, or the state of our database become trivial with this dependency.

7-) What is the primary difference between Spring and Spring Boot ?

Spring boot is more efficient and faster than Spring. Spring boot makes auto configuration.

8-) Why to use VCS ?

version control system

9-) What are SOLID Principles ? Give sample usages in Java ?

bir class bir iş yapmalı

L, alt class üst class yerine kullanılabilmeli

I, interface in tek sorumluluğu olmalı

Dependency Inversion, alt sınıfta bişey değiştiğinde üst sınıf eklenmemeli

10-) What is RAD model ?

RAD Model or Rapid Application Development model is a software development process based on prototyping without any specific planning. In RAD model, there is less attention paid to the planning and more priority is given to the development tasks. It targets at developing software in a short span of time.

11-) What is Spring Boot starter ? How is it useful ?

Spring Boot Starters were introduced to solve this problem so that the developers can spend more time on actual code than Dependencies. Spring Boot Starters are dependency descriptors that can be added under the **<dependencies>** section in pom.xml.

Spring Boot starters can help to reduce the number of manually added dependencies just by adding one dependency.

12-) What is Caching ? How can we achive caching in Spring Boot ?

Caching is a part of temporary memory (RAM). It lies between the application and persistence database. It stores the recently used data that reduces the number of database hits as much as possible. In other words, caching is to store data for future reference.

In Spring, the **cache abstraction** is a mechanism that allows consistent use of various caching methods with minimal impact on the code.

13-) What & How & Where & Why to logging ?

java logging api, provide the objects, methods, and configuration necessary to create and send log messages. Java provides a basic logging API through the **[java.util.logging]** package

14-) What is Swagger? Have you implemented it using Spring Boot?

Dökümantasyon yazma adına otomatik yapıyor

Swagger (now the “Open API Initiative”) is a specification and framework for describing REST APIs using a common language that everyone can understand.