HW#2 – Baran AYDIN

1 – IOC and DI means ?

IOC (Inversion of Control) is a principle in software engineering which transfers the control of objects or portions of a program to a container or framework.

DI (Dependency injection) is a pattern we can use to implement IoC, where the control being inverted is setting an object's dependencies.

Connecting objects with other objects, or “injecting” objects into other objects, is done by an assembler rather than by the objects themselves.

---

2 – Spring Bean Scopes ?

The scope of a bean defines the life cycle and visibility of that bean in the contexts we use it.

Spring framework defines 6 types of scopes:

Singleton, prototype, request, session, application, websocket

---

3 – What does @SpringBootApplication do ?

The @SpringBootApplication annotation is equivalent to using @Configuration, @EnableAutoConfiguration, and @ComponentScan with their default attributes

---

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.

Instead, we can declare the new code and the new behaviors separately.

Spring's AOP framework helps us implement these cross-cutting concerns.

---

5 – What is Singleton and where to use it ?

In object-oriented programming, a singleton class is a class that can have only one object (an instance of the class) at a time.

After first time, if we try to instantiate the Singleton class, the new variable also points to the first instance created. So whatever modifications we do to any variable inside the class through any instance, it affects the variable of the single instance created and is visible if we access that variable through any variable of that class type defined.

---

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

In essence, Actuator brings production-ready features to our application.

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

The main benefit of this library is that we can get production-grade tools without having to actually implement these features ourselves.

Actuator is mainly used to expose operational information about the running application — health, metrics, info, dump, env, etc. It uses HTTP endpoints or JMX beans to enable us to interact with it.

Once this dependency is on the classpath, several endpoints are available for us out of the box. As with most Spring modules, we can easily configure or extend it in many ways.

---

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

Spring boot provides following features where spring does not;

Opinionated ‘starter' dependencies to simplify the build and application configuration

Embedded server(tomcat) to avoid complexity in application deployment

Metrics, Health check, and externalized configuration

Automatic config for Spring functionality – whenever possible

---

8 – Why to use VCS ?

Version control is important to keep track of changes and keep every team member working on the right version. We need to use version control software for all code, files, and assets that multiple team members will collaborate on.

---

9 – What are SOLID Principles ? Give sample usages in Java ?

Design principles encourage us to create more maintainable, understandable, and flexible software. Consequently, as our applications grow in size, we can reduce their complexity and save ourselves a lot of headaches further down the road.

Sample usage: Single responsibility

As we might expect, this principle states that a class should only have one responsibility. Furthermore, it should only have one reason to change.

---

10 - What is RAD model ?

RAD (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 ?

Dependency management is a critical aspects of any complex project. And doing this manually is less than ideal; the more time you spent on it the less time you have on the other important aspects of the project.

Spring Boot starters were built to address exactly this problem. Starter POMs are a set of convenient dependency descriptors that you can include in your application. You get a one-stop-shop for all the Spring and related technology that you need, without having to hunt through sample code and copy-paste loads of dependency descriptors.

---

12 – What is Caching ? How can we achieve caching in Spring Boot ?

The Java Object Cache provides caching for expensive or frequently used Java objects when the application servers use a Java program to supply their content. Cached Java objects can contain generated pages or can provide support objects within the program to assist in creating new content.

The Spring Framework provides support for transparently adding caching to an application. At its core, the abstraction applies caching to methods, thus reducing the number of executions based on the information available in the cache. The caching logic is applied transparently, without any interference to the invoker. Spring Boot auto-configures the cache infrastructure as long as caching support is enabled via the @EnableCaching annotation.

---

13 – What & How & Where & Why to logging ?

Logging is a powerful aid for understanding and debugging program's run-time behavior. Logs capture and persist the important data and make it available for analysis at any point in time.

---

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

Swagger allows you to describe the structure of your APIs so that machines can read them. The ability of APIs to describe their own structure is the root of all awesomeness in Swagger. Why is it so great? Well, by reading your API’s structure, we can automatically build beautiful and interactive API documentation. We can also automatically generate client libraries for your API in many languages and explore other possibilities like automated testing. Swagger does this by asking your API to return a YAML or JSON that contains a detailed description of your entire API.

And yes, I have implemented it once I was dealing with Spring boot CRUD application.
