# HW#2 Talha Alçorba

## 1-) IOC and DI means ?

Inversion of Control, i.e. IoC, is actually a design principle. In object-oriented design, which we call object oriented design, it recommends that different types of controls be decoupled to ensure a loose connection between application classes.October is meant by control, the additional responsibilities that a class has other than its main responsibility.

Dependency Injection, on the other hand, is a design pattern that applies the IoC principle that I described at the very beginning to reverse the formation of dependent objects, that is, it is a design pattern that provides the Ioc principle.

## 2-) Spring Bean Scopes ?

In the spring bean configurations, bean attribute called 'scope' defines what kind of object has to created and returned. There are 5 types of bean scopes available, they are:

**1) singleton:** Returns a single bean instance per Spring IoC container.

**2) prototype:** Returns a new bean instance each time when requested.

**3) request:** Returns a single instance for every HTTP request call.

**4) session:** Returns a single instance for every HTTP session.

**5) global session:** global session scope is equal as session scope on portlet-based web applications.

If no bean scope is specified in bean configuration file, then it will be by default 'singleton'.

## 3-) What does @SpringBootApplication do ?

@SpringBootApplication specifies that the class is a configuration class and triggers automatic configuration through scanning through @EnableAutoConfiguration and the @ComponentScan description in the component.

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

Aspect is a programming paradigm that processes the cross cutting concern parts of our applications, and the starting point of the paradigm is to create solutions for concerns. The motivation for using this structure is to comply with principles such as single responsibility and don't repeat yourself.

Benefits of Spring AOP:

- Our application is more flexible and easy to manage,

- Getting rid of repetitive code layout,

- A cleaner and more understandable code jul,

- Separation of core logic and concerns from each other.

Some examples of use from everyday life:

**Logging** → Logging of requests and responses coming to our service.

**Transaction Management** → Performing the refund process after the error that will occur in the employee code loop from the receipt of the payment.

**Performance** → Calculation of the running time of the methods.

**Validation** → Check the user's e-mail permission before the e-mail will be sent.

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

The Singleton design pattern is used to get a single instance of a class.The purpose is to provide a global access point to the created object.As long as the system is running, a second instance will not be created, so that the desired object will be guaranteed to be created only once.Singleton objects are created once when they are first called, and subsequent requests are met through this object.

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

Spring Boot Actuator provides production-ready features of applications (health check, disk usage, heap dump, etc.) it activates automatically and provides a structure that allows you to interact with different HTTP endpoints. If you want the Spring Boot Actuator to be active in the project, the following Maven dependency block must be added to the pom file.

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

Spring is a widely used Java EE framework for building applications while Spring Boot is widely used to develop REST APIs. Spring aims to simplify Java EE development that makes developers more productive while Spring Boot aims to shorten the code length and provide the easiest way to develop Web Applications. The primary feature of the Spring Framework is dependency injection while The primary feature of Spring Boot is Autoconfiguration. It automatically configures the classes based on the requirement.

## 8-) Why to use VCS ?

Version control system (VCs), revision control revision control or Source Control (source control) he passes, change management system. It is a system that saves changes made to one or more files and allows you to return to a specific version later.

While the term ”version control" is often associated with programmers, it is equally important for writers, journalists, and even college students. Examples of common services that automatically keep track of document revisions and versions include Google Docs and DropBox. dec.

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

The following five concepts make up our SOLID principles:

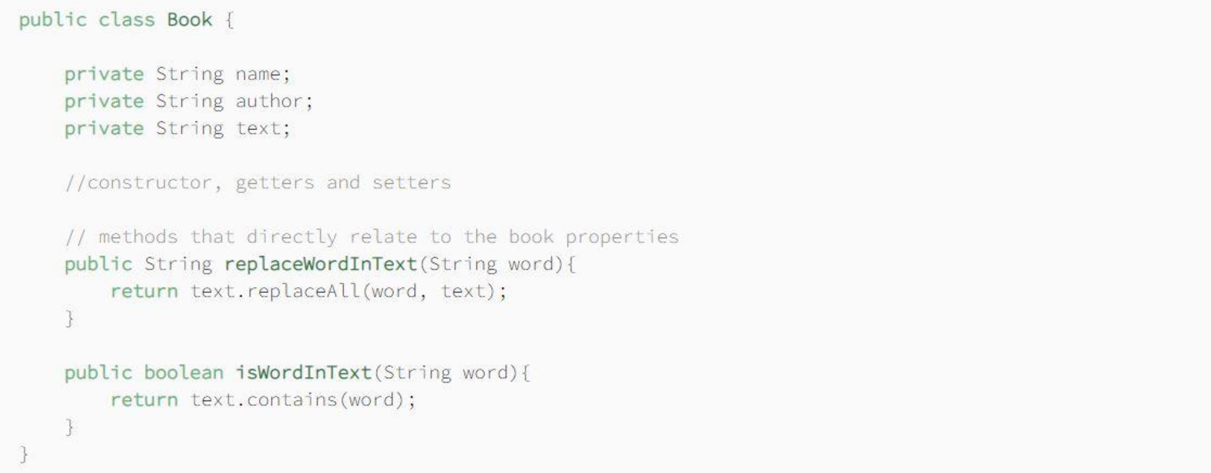
1-)Single Responsibility

2-)Open/Closed

3-)Liskov Substitution

4-)Interface Segregation

5-)Dependency Inversion



```
public class Book {

    private String name;
    private String author;
    private String text;

    //constructor, getters and setters

    // methods that directly relate to the book properties
    public String replaceWordInText(String word){
        return text.replaceAll(word, text);
    }

    public boolean isWordInText(String word){
        return text.contains(word);
    }
}
```

## 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 ?

Starters are a set of dependency descriptors that you can briefly add to your application. It saves you the hassle of searching for each of the technologies you want to use and adding them one by one decently. Thanks to the starters, you can easily add the Spring and related technologies you need to your application. If you want to use Spring and JPA as an example, just add the spring-boot-starter-data-jpa dependency to your project.

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

A cache is a high-speed data storage layer that stores a temporary subset of data. Caching allows efficient reuse of previously received or calculated data. When the data related to the caching method is requested in the subsequent process, a higher performance is achieved because the data is accessed from the primary storage location.

Data in a cache is usually stored in hardware such as RAM, and a connection may need to be established through software to access the data. The purpose of caching is to improve data access performance by minimizing the need to access the slow storage layer located at the bottom.

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

Logging operations are very important in terms of tracking the flow and problems in our project. The logging process is quite easy with Spring Boot. It uses Logback as the default for the Spring Boot Logging function, but it also works compatible with other logging libraries.

What we need to do to enable Logging in Spring Boot projects.properties or applications.you can start the logging process by adding it to the yml file.

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

One of the most important needs in Web API development is the need for documentation. Because what API methods do and how they are used should be clear in the documentation. It is both difficult to write api documentation manually and impossible to keep it up to date. In some form, it is necessary to produce this documentation up-to-date. Swagger is coming to our rescue here.

In order for the Swagger to be included in the Gradle project, it must be added to the (build.gradle) file as a dependency.