Umut Yıldız

1. IoC is a design principle. The purpose of IoC is that increase the loose coupling in application so we can mock and test easily. Dependency Injection is one of implementation of IoC. We should not use object instantiation in class of object so we can give reference of object through constructor (or other injection types such as setter injection).
2. When we declare a bean in spring, we can choose scope of that bean. There are 5 scopes of a bean in spring. **Singleton** is default in spring and we use singleton when we need only one instance of a bean. **Prototype** scope is that Spring IoC container creates a new instance of bean for every request. Other bean scopes are unique for web applications. **Request** is same logic with prototype but it is used for Http requests. When we use **Session** scope, a new bean is created for each Http session. The last one is **global-session** scope.
3. @SpringBootApplication describes that this application is a spring boot application and it provides to developers @EnableAutoConfiguration @ComponentScan and @Configuration. These annotations are used to enable Spring Boot's auto-configuration, searches packages which used @Component, describes that class which is used this annotation is a configuration class and we can use @Bean or import other configuration classes.
4. Spring Boot Applications are developed generally using N-Layered architecture and each layer has a different responsibility but developers need to some operations such as logging, validation, security etc. and defining them on each layer is the worst-case scenario. We can solve this problem with aspect-oriented programming. These cross-cutting concerns is implemented once and we can use in all packages without change code.
5. Singleton is a design pattern and it is creational design pattern. It is guarantee that only one instance is initialized in runtime. We can use when multiple classes need to use the same instance and when there should be only one object for the whole application. The class manages the instance of itself.
6. Spring boot's actuator module allows us to monitor and manage application usages in production environment, without coding and configuration for any of them. These monitoring and management information is exposed via REST like endpoint URLs.
7. Spring boot is a version of some features added in addition to Spring. Spring Boot is used for microservices or web application but Spring is used for Java EE apps.
8. We must use VCS because it provides that developer team members can work on same project easily. Every action on code is kept on history. We easily version the codes or changes we write.
9. SOLID principle includes 5 principles. Single Responsibility principle tells that each class has only one responsibility. For example, a Car class contains only fields and methods which related to Car. Open-closed principle tells that each unit of a class, a package etc. can be upgradable but can not changeable. Liskov Substitution principle tells that in short, this principle says that to build software systems from interchangeable parts, those parts must adhere to a contract that allows those parts to be substituted one for another. Interface Segregation principle tells those properties of a interface must be divided because a class which implements this interface should not be forced to use unnecessary properties by this interface. Dependency Inversion principle tells that we must not use directly concrete classes. We must use classes with interfaces. For example in Spring Boot, We have a UserServiceImpl class and this class implements UserService interface and when we want to use UserServiceImpl in UserController , we must use UserService to access UserServiceImpl.
10. The Rapid Application Development (or RAD) paradigm is focused on prototyping and an iterative process with little (or no) preparation ahead of time. In general, using a RAD

approach to software development implies focusing less on planning and more on development and creating a prototype. Business Modelling, Data Modelling, Process Modelling, Application Generation, Testing and Turnover are phases of RAD.

11. In spring, developers spend much time to run a basic spring application because of dependencies and it provides to developers, they focus on only business code, not manage dependencies.
12. We can imagine cache like intermediate holder between app and database. It provides decreasing requests to database for cost and it holds some data for future and it increases the speed of system. We can achieve that in Spring Boot using @EnableCaching annotation.
13. Logging is keeping a record of all data input, processes, data output, and final results in a program. This is part of a much more grand, complex process, though, so you want to program with a clear goal in mind, and not try to do several programming disciplines at once.
14. Swagger is a collection of rules (or standard) for describing REST APIs in a format. The format is human-readable as well as machine-readable. As a consequence, it may be used not only to exchange documentation among product managers, testers, and developers, but also to automate API-related operations via various tools. Yes, I have implemented it using Spring Boot.