**1 – IOC and DI means ?**

**Inversion** of Control (**IoC**) means that objects do not create other objects on which they rely to do their work. ... **Dependency Injection** (DI) means that this is done without the object intervention, usually by a framework component that passes **constructor** parameters and set properties.

**2 – Spring Bean Scopes ?**

|  |  |
| --- | --- |
| 1 | **singleton**  This scopes the bean definition to a single instance per Spring IoC container (default). |
| 2 | **prototype**  This scopes a single bean definition to have any number of object instances. |
| 3 | **request**  This scopes a bean definition to an HTTP request. Only valid in the context of a web-aware Spring ApplicationContext. |
| 4 | **session**  This scopes a bean definition to an HTTP session. Only valid in the context of a web-aware Spring ApplicationContext. |
| 5 | **global-session**  This scopes a bean definition to a global HTTP session. Only valid in the context of a web-aware Spring ApplicationContext. |

**3 – What does @SpringBootApplication do ?**

 Spring does not generate any code automatically and not using any xml configuration file . so spring uses internally pragmatically configuration done by **spring boot developer that are provided by jar**. To Enable preconfigured jars we just need to define dependency in pom.

**4 – What is Spring AOP ? Where and How to use it ?**

Spring AOP **enables Aspect-Oriented Programming in spring applications**. In AOP, aspects enable the modularization of concerns such as transaction management, logging or security that cut across multiple types and objects (often termed crosscutting concerns)

**5 – What is Singleton and where to use it ?**

It is used **where only a single instance of a class is required to control the action throughout the execution**. A singleton class shouldn't have multiple instances in any case and at any cost. Singleton classes are used for logging, driver objects, caching and thread pool, database connections.

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

Spring Boot actuator helps us to manage and monitor our application when we deploy our application to production. It contains various endpoints by which we can see where the resource is up, it has two types of endpoints which we can use to monitor our spring boot application which is JMX and HTTP, it provides us the production-ready application by which we can monitor the health of our application we should use this feature. It also applied some more things automatically in applications like metrics, health, and auditing.

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

Spring core is basically the module which encapsulates all the core functionalites present within the spring framework, functionalities like Dependency injection, Aspects etc etc.

Spring boot is the module in the spring framework which is primarily used for developing web applications. Via Its main slogan : convention over configuration, it eases the effort which is required from development to deployment through giving the developer a set of default configurations which he/she can override or use the default. You can easily spin up a web app which serves rest API’s in minutes, when compared to WAR files which needs a web container to run on, which in turn requires confuguration.

**8 – Why to use VCS ?**

 Version control is **important to keep track of changes** — and keep every team member working on the right version. You should 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 ?**

The SOLID principles of Object Oriented Design include following five principles:

* Single Responsibility Principle (SRP)
* Open Closed Design Principle
* Liskov Substitution Principle (LSP)
* Interface Segregation Principle (ISP)
* Dependency Injection or Inversion principle

**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 are dependency descriptors that can be added under the <dependencies> section in pom. xml. ... These starters give all the dependencies under a single name. For example, if you want to use Spring Data JPA for database access, you can include spring-boot-starter-data-jpa dependency.

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

Caching is a mechanism to enhance the performance of a system. It is a temporary memory that lies between the application and the persistent database. Cache memory stores recently used data items in order to reduce the number of database hits as much as possible.

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

The purpose of logging is to create an ongoing record of application events. Log files can be used to review any event within a system, including failures and state transformations. Consequently, log messages can provide valuable information to help pinpoint the cause of performance problems.

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

Swagger (now the “Open API Initiative”) is a specification and framework for describing REST APIs using a common language that everyone can understand. There are other available frameworks that have gained some popularity, such as RAML, Summation etc. but Swagger is most popular at this point of time considering its features and acceptance among the developer community.

It offers both human readable and machine readable format of documentation. It provides both JSON and UI support. JSON can be used as machine readable format and Swagger-UI is for visual display which is easy for humans to understand by just browsing the api documentation.