HW - 2

1) IOC and DI means?
Ans:
Inversion of Control is a principle which is used in software engineering. In that principle objects or portions
of the program transferred into a container or a framework. Therefore, framework takes control of the flow of
a program and make calls to our custom code. Additionally, inversion of control can achieve through different
mechanisms such as; strategy design pattern, service locator pattern, and dependency injection(DI).

Dependency Injection is a pattern which is used to implement inversion of control. In dependency injection usage of an object is decoupled and it makes the class independent of its dependencies. It also helps to usage of SOLID's dependency inversion and single responsibility principles.

2) Spring Bean Scopes?
Ans:
Scope of a bean determines the life cycle and visibility of the bean in the contexts we use.
6 scope types were defined with the latest version of Spring Framework;

1. Singleton,
2. Prototype,
3. Request,
4. Session,
5. Application,
6. Websocket

** Last four scopes which mentioned above are only available in a web aware application.

3) What does @SpringBootApplication do?
Ans:
@SpringBootApplication annotation is used to declare 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) Why Spring Boot over Spring?
Ans:
Spring provides comprehensive infrastructure support for developing Java applications. It's included with Dependency injection and out the box modules like; Spring JDBC, Spring MVC, Spring Security, Spring AOP, Spring ORM, Spring Test. With these modules development time of an application is greatly reduced.

Spring Boot is basically an extension of the Spring framework and it eliminates the conventional configurations that is required for the spring application. Dependencies are optioned to simplify the build and
application configuration, embedded server to avoid complexity in application deployment, metrics, health check and externalized configuration, automatic config

for Spring functionality. With these features spring boot provides more faster and more efficient development ecosystem.

In a manner of deployment Spring boot has some advantages over the Spring which is:

1. Embedded container support
2. Provision to run the jars independently using the command java -jar.
3. Option to exclude dependencies to avoid potential jar conflicts when deploying in an external container
4. Option to specify active profiles when deploying
5. Random port generation for integration tests.

5)What is Singleton and where to use it?
Ans:
Singleton is a class which is only have one object at a time.
Purpose of using Singleton Class is to restrict the limit of the number of object creation to one. Therefore, it ensures the access control to resources, ex. Socket or database connection.
It also prevent the memory wastage since instance creation is restricted, as the object creation will only occur once.
That single object that created can used repeatedly according to needs and requirements. This is the reason why the multi-threaded and database applications mostly make use of the Singleton pattern in Java for caching, logging, thread pooling, configuration settings, and more.

6)Explain @RestController annotation in Spring boot?
Ans:
@RestController annotation is used to simplify the creation of RESTful web services.
It is specialized version of @Controller annotation which combines @Controller and @ResponseBody annotations.
The controller with annotation @RestController does not require @ResponseBody annotation.

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

Spring	Spring Boot
Allows loosely coupled applications.	Allows standalone applications.
Main feature dependency injection.	Main feature is auto-configuration.
Involves lots of boilerplate code.	Reduces boilerplate code.
Dependencies defined manually.	Starters take care of the dependencies.
Server setting up manually.	Embedded server like tomcat / jetty

8)Why to use VCS?
Ans:
VCS is used keep track of every modification that made on the code and allows to turn back when the mistake was made or can be used compare other versions of the code. Additionally it provides collaboration of team members at the same time.

9) What are SOLID principles? Give sample usage in JAVA?

Ans:
SOLID principles are five principles of object oriented class design.

1. Single Responsibility
2. Open-Closed
3. Liskov Substitution
4. Interface Segregation
5. Dependency Inversion

Single Responsibility Principle: A class should do one thing and therefore it should have only a single reason to change. Ex: If a class is a data container, like Book class or a Student class, and it has some fields regarding that entity, it should change only when we change the data model.
Open - Closed Principle: In that principle class should be open for extension and closed to modification. (Modification means change in the code of an existing class and extension means adding new functionality.)
Therefore new functionality can be added without touching the existing code for the class. (Since modifications can cause unwanted bugs and etc.)
Liskov Substitution Principle: Subclasses should be substitutable for their base classes. Which means, given class B is a subclass of class A, we should be able to pass an object of class B to any method that expects and object of class A and the method should not give any unwanted output in that case. This is the expected Behavior, because when we use inheritance we assume that the child class inherits everything that the superclass has. The child class extends the behavior but never narrows it down. Therefore, when a class does not obey this principle, it leads to some bugs which is hard to determine.
Interface Segregation Principle: It means keeping things separated, and the Interface Segregation principle is about separating the interfaces. It states that many client-specific interfaces are better than one general-purpose interface. Clients should not be forced to implement a function they do not need.
Dependency Inversion Principle: Classes should depend upon interfaces or abstract classes instead of concrete classes and functions. We want our classes to be open to extension, so we have reorganized our
Dependencies to depend on interfaces instead of concrete classes. Our PersistenceManager class depends on InvoicePersistence instead of the classes that implement that interface.
10) What is RAD model?
Ans:
Rapid application development model is based on prototyping and iterative model with no specific planning. In general, RAD approach to software development means putting lesser emphasis on planning tasks and more emphasis on development and coming up with a prototype.
11) What is Spring Boot Starter? How is it useful?
Ans:
Spring boot starters are dependency descriptors that can be added under the dependencies section in pom.xml file. There are various spring boot starters for different spring and related technologies.
It increases the productivity by decreasing the configuration time, managing the POM is easier, tested, production ready, and supported dependency configurations, no need to remember the name and version of the dependencies that is used.

12) What are the Spring Boot Annotations?

Ans:
@EnableAutoConfiguration, @SpringBootApplication, @ComponentScan, @Configuration.

13) What is Spring Boot dependency management?

Ans:
Spring Boot manages dependencies and configuration automatically, Each release of Spring Boot provides a list of dependencies that it supports. It is available as a part of the Bills of Materials that can be used with Maven. So we need not to specify the version of the dependencies in our configuration. Spring Boot takes care of all the dependencies, upgrades all dependencies automatically when the spring boot version is updated.

14) What is Spring Boot Actuator?

Ans:
Spring Boot Actuator is a sub project of the Spring Boot Framework. It uses HTTP endpoints to expose operational information about any running application. So it is used for monitoring and managing purposes for the application.