SPRING FRAMEWORK

1. widely used EE framework. EE: "Also called Jakarta EE (Enterprise Edition or previously named J2EE) is indeed an umbrella or abstract specification (expressed in a formal language). Anybody is open to develop and provide a working implementation of the specification. Enterprise JavaBeans (EJB) is part of the Jakarta EE specification. It's in the Jakarta EE API."
2. Simplifies Jakarta EE development
3. Helps developer to be more productive at work.
4. Focuses on several areas of an application.
5. Provides a wide range of features: Spring core, MVC, AOP for cross-cutting logic "concerns that are spread across various parts of your application like u have a lot of modules in your application having different concerns one is doing one thing, other one is doing completely a separate thing and they are communicating with each other" e.ge., @AuditTrail, Spring ORM AND Spring JDBC (for persistence layer)
6. Dependency Injection (helps to create loosely coupled applications.)

Cohesion: Togetherness of module's elements. High cohesion means functionalities of one module are strongly related. Example of low Cohesion (Adder class has add() which is logic method and input(), display() which are ui methods) (low usability, readability, testability) Example of High Cohesion (Window class has input() and display() which are ui methods, whereas Calculator class has Add(), Subtract(), Multiply() and Divide() methods which are logic methods.) Coupling: Dependency of one module on other modules. Low coupling or loose coupling means a module has fewer relationships with other modules. Example of High coupling: mail system is being used in employee system, client system, and server system.

Spring boot

1. Less code, flexibility, and provides the easiest way to code with annotation configurations, default code
2. Less time to develop standalone application with less configurations.
3. Auto-configuration is significant feature.

Benefits:

1. Dependency Resolution: No need to worry about Dependency. (But for spring boot we use spring-boot-starter-web and for persistence and hibernate we use spring-boot-starter-data-jpa and we don't need to identify version.) (Web app with spring framework and hibernate, I have to identify and specify all dependencies with versions that should match)
2. Minimum configuration: add few properties in application.properties file (like jdbc driver, url username and password for persistence and jpa related properties for hibernate).
3. Embedded server for testing: Tomcat, WebLogic servers are required to run application for spring. Spring boot provide internal Embedded Tomcat server (so run directly as Java application.)
4. Bean auto scan (for spring I have to specify the component scan) (but for spring boot we can add @ComponentScan() or follow package structure)
5. Health Metrics: Thread dumps, memory occupation (for spring boot we use Actuator for monitoring or observing) (for spring we use customized framework or any 3rd party service).