What is Spring?

Created by Rod Johson (“Expert One-on-One: J2EE Design and Development”).

lightweight open source framework

developing loosely coupled

highly cohesive systems

what is coupling and cohesive?

Coupling is the degree to which one class knows about another class. Let us consider two classes class A and class B. If class A knows class B through its interface only i.e. it interacts with class B through its API then class A and class B are said to be loosely coupled.

Cohesion is used to indicate the degree to which a class has a single, well-focused purpose. Coupling is all about how classes interact with each other, on the other hand cohesion focuses on how single class is designed. Higher the cohesiveness of the class, better is the OO design.

Disadvantage:

**J2EE applications tend to contain excessive amounts of "plumbing" code.** Many code reviews repeatedly reveal a high proportion of code that doesn't *do* anything: JNDI lookup code, Transfer Objects, try/catch blocks to acquire and release JDBC resources. . . . Writing and *maintaining* such plumbing code proves a major drain on resources that should be focused on the application's business domain.

**The EJB component model**

Advantage:

**Spring enables you to enjoy the key benefits of J2EE, while minimizing the complexity encountered by application code.**

**The essence of Spring is in providing enterprise services to Plain Old Java Objects (POJOs). This is particularly valuable in a J2EE environment, but application code delivered as POJOs is naturally reusable in a variety of runtime environments.**

IOC:

Spring's flavor of IoC for configuration management is rather more specific. Consequently, Martin Fowler, Rod Johnson, Aslak Hellesoy, and Paul Hammant coined the name *Dependency Injection* in late 2003 to describe the approach to IoC promoted by Spring, PicoContainer, and HiveMind-the three most popular lightweight frameworks.

DI :

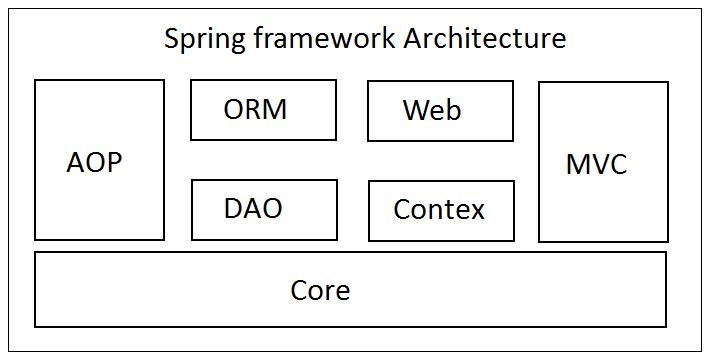
**Dependency Injection is based on Java language constructs, rather than the use of framework-specific interfaces. Instead of application code using framework APIs to resolve dependencies such as configuration parameters and collaborating objects, application classes expose their dependencies through methods or constructors that the framework can call with the appropriate values at runtime, based on configuration.**

Dependency Injection is a form of *push* configuration; the container "pushes" dependencies into application objects at runtime. This is the opposite of traditional *pull* configuration, in which the application object "pulls" dependencies from its environment. Thus, Dependency Injection objects never load custom properties or go to a database to load configuration — the framework is wholly responsible for actually reading configuration.

Container :

IoC container takes responsibility for object instantiation

**SPRING ARCHITECTURE:ITECTURE:**





Java Singleton and spring Singleton are same?

Spring mvc

<https://jeromejaglale.com/doc/java/spring>

All advanced spring related demos

<http://krams915.blogspot.com/p/tutorials.html>

Spring 4 features:

<https://javabeat.net/spring-4/>

books for spring :

<https://www.codejava.net/books/recommended-books-for-spring-framework>

References :

<https://www.quora.com/How-do-I-learn-Spring-Framework>