**Introduction to Spring**

Spring is an open source framework created to address the complexity of enterprise application development. One of the chief advantages of the Spring framework is its layered architecture, which allows you to be selective about which of its components you use while also providing a cohesive framework for J2EE application development

### Features:

#### Lightweight:

spring is lightweight when it comes to size and transparency. The basic version of spring framework is around 1MB. And the processing overhead is also very negligible.

#### Inversion of control (IOC):

The basic concept of the Dependency Injection or Inversion of Control is that, programmer do not need to create the objects, instead just describe how it should be created. No need to directly connect your components and services together in program, instead just describe which services are needed by which components in a configuration file/xml file. The Spring IOC container is then responsible for binding it all up.

#### Aspect oriented (AOP):

Spring supports Aspect oriented programming .  
[Aspect oriented programming](http://en.wikipedia.org/wiki/Aspect-oriented_programming) refers to the programming paradigm which isolates secondary or supporting functions from the main program's business logic. AOP is a promising technology for separating crosscutting concerns, something usually hard to do in object-oriented programming. The application's modularity is increased in that way and its maintenance becomes significantly easier.

#### Container:

Spring contains and manages the life cycle and configuration of application objects.

#### MVC Framework:

Spring comes with MVC web application framework, built on core Spring functionality. This framework is highly configurable via strategy interfaces, and accommodates multiple view technologies like JSP, Velocity, Tiles, iText, and POI. But other frameworks can be easily used instead of Spring MVC Framework.

#### Transaction Management:

Spring framework provides a generic abstraction layer for transaction management. This allowing the developer to add the pluggable transaction managers, and making it easy to demarcate transactions without dealing with low-level issues. Spring's transaction support is not tied to J2EE environments and it can be also used in container less environments.

#### JDBC Exception Handling:

The JDBC abstraction layer of the Spring offers a meaningful exception hierarchy, which simplifies the error handling strategy. Integration with Hibernate, JDO, and iBATIS: Spring provides best Integration services with Hibernate, JDO and iBATIS