Spring

spring is a lightweight framework. It compromises several modules such as IOC, AOP, DAO content, ORM, mvc etc

- Inversion of control (IOC) & dependency injection
  There are designs patterns that used to remove dependancy from programming code. so spring IOC is responsible for injecting the dependancy. we provide metadata to the IOC container through xml or annotation
  - makes the code loosely coupled
  - easy maintanance
  - easy to test

Advantages of spring
- predefined templates - less code
- loose coupling - because of dependancy injection
- Easy to test - no server required
- Light-weight - no force to inherit class or implement interface
- Fast development -
- Abstraction
- declarative support for caching, validation, transaction and formatting

- Data access framework
  allows the developer to use persistent APIs such as JDBC hybernate, solves problems in connecting with JDBC deal with exceptions implement transaction management

- spring MVC

  Allows to build web apps based on MVC architecture All the request made by a user first go through the controller then to different views, (JSP pages or servlets), forms validation, handling.

- Transaction management

  This provides Java Transaction API (JTA) for global transaction manged by application Server & local transaction managed by JDBC hibernate or other data access APIs

- spring web service :

  provide layered based approaches managed by XML parsing. spring provides effective mapping of transmitting incoming XML message request to an object and the developer to easily distribute XML message between 2 machines.

- JDBC abstraction layer

  Helps the user in handling errors in easy and efficient manner, the JDBC code can be reduced handles exception such as driver not found. All SQL exceptions are translated into the data access exception class.

- spring test context framework

  provides unit & integration testing

spring core

provides the IOC container,

- two types of implementation

1) Bean factory & Application context

defined by org.springframework.beans. Beanfactory interface and acts as a container for beans. It allows to configure and specification of dependencies from program logic. Beanfactory is responsible for instantiating the object. It also configures and assembles dependencies b/w objects XML bean factory class is most common implementatn of Beanfactory interface

Spring AOP

AOP breaks down the program into aspects or concerns. The aspects are regular spring beans or classes annotated with @aspect. Then aspects helps in transaction management and logging & failure monitoring of a system

Spring Boot

Spring boot makes it easy to create standalone productn grade spring based apps you can just run

- opinionated
- convention over configuratn
- stand alone
- production ready