# Spring introduction

SPRING 5.X

* Method injection
* Lookup method injection
  + Difference between programming language, software technology and framework
  + Introduction to Spring Framework
  + Evolution of spring Framework
  + Modules of Spring in Spring 1.x,2.x,3.x,4.x and 5.x
  + MVC Architecture
  + Role of spring framework in MVC Architecture application development
  + Definition of spring framework
  + POJO class, POJI, javaBean, Component Class, spring Bean classes

# Spring Core Module

* + Introduction to IOC
  + Introduction to Spring Container/IOC Container
  + Types of Dependency Injections
    - Setter injection
    - Constructor injection
    - Aware injection
* Introduction to Design patterns
* Factory design pattern
* Strategy design pattern
* Layered Application demonstrating real time dependency injection
* Resolving /identifying params in constructor injection
* Bean inheritance
* Collection merging
* Null injections
* Bean alias
* Default bean ids
* Performing dependency lookup by using IOC container
* <Idref>tag
* Understanding factory methods
* Factory method bean installation
  + Stack factory method bean installation
  + Instance factory method bean installation
* Singleton java class and its usecases
* Bean scopes
  + Singleton
  + Prototype
  + Request
  + Session
  + Application
  + Websocket
* Bean Wiring
  + Explicit wiring or auto wiring
  + By name
  + By type
  + Constructor
  + Autodetect
* P-namespace, c-namespace
* Application context container
  + Preinstantiation of singleton scope beans
  + Working with properties files
  + I18N(internalization)
  + Event handling
  + BEANFACTORY VS APPLICATION CONTEXT
  + AUTOMATIC REGISTRATION OF BEAN post processor and bean factory post processor
* Bean life cycle
  + Declarative approach
  + Programmatic approach
  + Annotation driven approach
* Nested IOC containers
  + Presentation tier
  + Business tier
  + Various attributes of <ref> tag
* FactoryBean
* ServiceLocator as factory bean
* factory method bean instantation based service locator
* Method replacement/method injection
* Aware injection
* Lookup method injection
* Bean postprocessor
* BeanFactory Post Processor
* Property editors
* Custom property editors
* Spring expression language

# Spring core module with annotations

* Spring stereo type annotations
  + @component, @service, @controller, repository and etc
  + @auto wired, @qualifier, @lazy and etc
* Working with java config annotations
  + @named, @inject, @Resource and etc
* Working with properties files in annotations environment
* Developing layered applications in annotations environment

# Spring core module with 100% code/java config Approch

* working with @Bean, @configuration, @lazy, @property source and etc
* developing layered application
* working with annotation Config application context

**Spring boot core**

* spring boot starters
* understanding @spring boot applications
* auto configuration
* example applications
* spring profiles
* spring boot stand alone flow
* working with sts plugins eclipse to develop spring boot application

# Spring JDBC/DAO

* introduction
* plan JDBC limitations
* spring JDBC /DAO advantages
* working with different data sources
* JDBC Template
* JNDI Registry and service managed JDBC connection pool
* Callback interfaces
* Batch processing/ updating
* Named parameter JDBC template
* Working with simple JDBC insert, simple JDBC call
* Simple JDBC call to call PL/SQL procedures
* Mapping SQL operations as subclasses
* Spring JDBC / DAO with annotations Spring JDBC / DAO with 100% code approach
* Spring boot JDBC/ DAO
* Spring boot DAO - JDBC template
* Spring boot DAO -sample JDBC insert
* Spring boot application flow
* Working with data sources through AutoConfiguration in spring boot 1.x,2.x

# Spring AOP model

* Introduction
* Need of AOP
* Proxy design pattern
* AOP Terminologies /principles
  + Aspect, advice, after advice, Joinpoint, Pointcut
  + Target class, proxy class, weaing
* Types of advices
  + Before advice, after advice, around advice

Throw advice

* Types of point cuts
  + Static pointcuts, dynamic point cuts
* Programmatic spring AOP
* Declarative Spring AOP
* @AspectJ style AOP support
* Spring AOP/Aspectj AOP with annotations
* Spring AOP/Aspectj AOP with 100% code approach
* Spring AOP/Aspectj AOP with spring boot AOP

# Spring transaction management

* Introduction to transaction management
* Local transaction Vs distributed transactions
* 2pc principle
* Transaction models
  + Flat transaction model
  + Nested transaction model
* Need of spring transaction management
* Choosing spring transaction manager
  + DataSource transaction manager
  + Hibernate transaction manager
  + JTAT transaction manager and etc
* Different ways of implementing of spring transaction management
  + Programmatic approach
  + Declarative approach using spring AOP/Aspectj AOP
  + Annotation driven approach using aspect AOP
  + 100% code driven approach using Aspectj AOP
  + Spring boot driven approach using AspectJ AOP
* Transaction attributes
* Transaction and integration testing
* Distributed transaction management implementation using web logic sever, Atomikos API
* Configuration transaction isolation levels
  + Read uncommitted, read committed
  + Repeatable Read, serializable
* Working with rollBackfor, norollback for timeout and etc in transaction management

# Spring MVC

* Introduction to MVC
* Understanding mvc1,mvc2 architectures
* Front controller design pattern
* Intercepting filter vs front controller
* Different types of servelts url patterns
* Spring mvc resoures
* Spring mvc flow
  + Structural flow
  + Strategy flow(code based flow)
* dispatcherServlet
* different controller classes
  + paramaterizableViewcontroller
  + urlfilenameviewcontroller
  + abstract controller
  + simple form controller
  + multi action controller
  + abstractWizardForm controllerand etc
* developing mini project with curd operation
* contextloaderlistner
* working with two containers
* hadler Mappings
  + bean name urlhandlerMapping
  + simple url handlerMapping
* controllerClassNameHandlerMappi ng
* defultannotatingHandlerMapping
* request mappingHandlerMapping and etc
* Handler Mapping chaining
* Form validations
  + Enabling server side validations only whenclient side validation are not

Enabled

<form:errors>

* View resolvers
  + InternalresourceViewResolver
  + Urlbasedviewresolver
  + resourceBundleViewresolver
  + xmlviewResolver
  + tilesviewresolver
  + BeannameviewResolver and etc
* Viewresolverchaining
* Views
  + Internal resource view
  + Jstlview,tilesview,abstractpdf view
  + Abstractxlsview and etc
* Exception handling in spring MVC
* Tiles integration withspring MVC
* Message sources and I18n
  + Formatting lables,formatting numbers
  + Formatting dates
  + Formatting currency symbol
  + Locale
* Mvc namespace
* Handler interceptrs/adapters
  + Checking browser type
  + Checking time out period
  + Preventing double posting problem
* Pdf views and excel views
* File uploading and downloading
* Spring MVC with annotarions
  + Annotation driven controllers
    - @requestmapping,@controller
    - @modelattribute,@sessionattribut e,@requestparam
    - Requestview to viewname translator
    - Mvc namespace
    - Annotation driven from validation using hibernate validator api,jee validator api
* Spring mvc with 100% code approach
  + Dynamic registration of servlet
  + Webapplication intitializer
  + Springservelt container intializer
  + @enablewebmvc,@import
  + abstractAnnottionconfigDispatherS ervleInitializer
* Springboot mvc
  + springServletintilizer
  + working with embedded tomcatserver
  + springboot mvcflow
  + springboot devtools
  + application.properties,application.y ml in springboot
  + developing mini project with curd operation
  + solving double posting problems in spring boot mvc using postredirectGetpattren
* Profiles in spring,springboot

# Spring security

* Introduction
  + Authnitication,authorization
* Authentication manager authentication info provider
* Need of spring security
* Deliagatingfilterproxy
* Security namespace
  + Formlogin
  + Remember me
  + Session concurrency
  + Logout and etc
* Working with different authenticating providers
  + Xml file, property file,database,ldap sever
* Security examples
  + Using xml configurations
  + Using annotation configurations
  + 100% code driven configurations
  + Spring boot configurations
* Using ldap server

# Spring social

* Introduction
* Need of spring social
* Understanding PASS and SASS
* Accessing facebook data
* Accesing twitter data
* Spring social example

# Spring ORM

* Introduction ORM
* Spring ORM advantage
* Integration with hibernate using hibernate template
* Hibernate template and its methods
* Hibernate DAO support
* Hibernate Callback interfaces

# Spring data and spring data JPA

* Need of spring data
* Spring data Jpa
* Finder methods of finder api
* Repositories
  + Jparepository
  + Curd repository
  + Paging and sorting repository
* Spring data custom query
  + Automatic custom query
  + Manual custom query
* Spring data jpa exception translator
* Difference b/w hibernate and spring data JPA
* Interacting with mango DB
* Junit test cases

# Spring batch

* Need of batchprocessing
* Need of spring batch
* Understanding spring batch architecture
* Working with batch namespace
* Working with different itemReaders,item writers and item processors
* Converting database data to csv file
* Converting csv file to xml file
* Converting xml file to csv file
* Spring batchapplication using spring boot
  + jobbuilderFactory
  + stepBuilderFactory
  + step point TaskLet and etc
* Working with scheduler

# Spring webservices

* Understanding mail server and mail clients
* Understanding mail message structure
* Spring mail abstraction over java mail
* Spring mail using springboot

# Spring with oAuth protocol

* Understanding oauth 2
* Understanding roles in oauth 2
  + Resourse owner
  + Client
  + Resourse server
  + Authorization server
* Working with sso(single sign on)

# Introduction to spring microservices How to explain project architectures

* Servelt,jsp project architecture
* Spring mvc project architecture
* Spring boot with micro service project architecture
* Adapting agile methodology
* SOAP integration
* Restful integration
* Handling spring security in webservices

# Spring mail

* Understanding java mail API
* Understanding SMTP,POP3,IMAP protocols