Day 1 –

* DBMS & RDBMS
* Types of RDBMS (Oracle, MySQL, MSSQL, Postgres, DB2, H2(In-Memory Database)
* TYPEs of SQL (DDL,DML,TCL,DCL,DQL)
* Constraints & Clause
* Joins
* Ways of Accessing the Database (Using CLI, Using GUI, Using Programming Lang)
* CRUD Operations
* 5 Steps to interact with DB (Loading, Connecting, Quering, Processing, Closing)
* JDBC API (Connection, Statement/PreparedStatement/CallableStament, ResultSet, Driver, DriverManager)

Day 2 –

CRUD operation using Core Java Applications. (Adding Driver Jar to the Build path is enough)

CRUD operation using Web Applications. (Add Drivers to Build Path & Deployment Assembly)

Generic CRUD using ResultSetMetaData

Spring Framework – Java based framework used to create loosely couple Enterprise Java Applications.

Framework of Frameworks (It support other framework – Struts, Hibernate)

Design Patterns (What is Patterns, Why Patterns)

Important Design Patterns used in Spring Framework

Day 3 –

Singleton, Inversion of Control(IoC), Dependency Injection (DI)

Different ways of creating object and initializing it.

Types of Constructors (Default & Parameterized)

Default/No-Arg Constructor

Parameterized Constructor (Fully Parameterized & Partially Parameterized)

IoC – Changing the control – Managing the life cycle of beans. (The way how object is created and managed) – Ola Example

Spring – IoC container take cares of Bean Handling

Types of Classes

Spring modules (Core, MVC, AOP, Test, Security, Data)

Spring Core ( Core, Beans, Context, ASM, Log, EL)

Creating Beans using Spring framework.

ApplicationContext.xml , BeanFactory

Spring Configuration file – XML file

XML file is both case and space sensitive.

Web.xml (Deployment Descriptor) – From Java 5 (Annotations)

<https://spring.io>

<https://docs.spring.io/spring-framework/docs/current/reference/html/core.html>

Scopes of Bean

Ways of Configuring the Bean (XML Config. & Annotation Config.)

Spring MVC framework

Spring based Annotations

MVC – Model, View & Controller

Scopes of Bean (Bean Scope)

Singleton Bean –

Request

Http – HyperText Transfer Protocol

Hypertext –

Protocol – Set of Rules

Http is a stateless protocol –

Bean Scopes

1. Singleton
2. Request/page (scope=”request”)(@RequestScope)
3. Prototype/Proxy (scope=”prototype”)
4. Session (@SessionScope) (scope=”session”)
5. Application (@ApplicationScope) (scope=”application”)
6. Websocket

viewTable.jsp?table=course&id=2

requested resource = viewTable.jsp (File)

table& id are requestParameters

Spring MVC

Annotation Driven Beans

Types of Injections

1. Constructor Injection
2. Setter Injection

Injection – Injecting the Object at the required time (Early/Lazy) time of creation/time of needed

Ways of Configuring Beans

1. XML based Configuration
2. Annotation based Configuration

Autowiring

Wiring – Is a process of Creating and managing the beans

Types of Wiring

* No
* ByName
* ByType
* Constructor

@Autowired – Property Level Annotation

@Component

@Bean

AOP - Aspect Oriented Programming

AOP is a module in Spring Framework.

Annotation – In Java5, they have introduced Annotation concept.

Annotation – Meta data

Meta Data = Data about Data.

Java Complete Reference = Java Programming Lang (Data)

1. TOC (Table of Contents)
2. Index

Annotation are meta data which provides more/detailed information about class/method/property/variable etc.,

Generally annotation starts with @ symbol

@Deprecated

@SuppressWarnings

Meta Annotations - @Retention, @Documented, @Target (Annotations used in custom annotations)

Spring uses the power of Annotations

1. Spring Core Annotations
2. Spring MVC Annotation
3. Spring Bean Annotation
4. Spring Config Annotation
5. Spring Data Annotation (JPA)
6. Spring Scheduling Annotation
7. Spring Boot Annotations

<https://www.javacodegeeks.com/2019/05/spring-core-annotations.html>

<https://www.geeksforgeeks.org/spring-core-annotations/>

Property/Field Level Injection

Public class Student {

@Autowired

Address address;

}

Constructor Level Injection

Public class Student {

Address address;

@Autowired

Student (Address address) {

This.address=address;

}

}

Spring MVC also called as Spring Web Module.

Web Applications

* Runs on Web/Application Server (Tomcat)
* Web.xml (Deployment Descriptor)
* Maven

Maven is a Open Source, Project Management Tool for Java.

JVM based Languages (JAVA, Kotlin, Groovy)

Spring Boot supports JVM based language

Maven is used to perform

* To Build (Source -> Byte code)
* To Deploy (Deploying the package in Server)
* To Test (Running Unit Test code)
* To Package (Converting multiple files to single JAR/WAR file)
* To Manage Dependency (Adding all libraries and dependencies in the form of jar files)

Backbone of Maven is pom.xml file

POM – Project object Model

groupId (package name)

artifactId (application name)

version()

<https://www.javatpoint.com/spring-mvc-form-example>

DispatcherServlet will act as a FrontController in Spring

ModelAndView (Data and UI)

Model(Data), View (UI), Controller (Java/Spring code)

Spring MVC

1. Web.xml is the Entry point (index.jsp)
2. courseForm – Check for a request mapping in CourseController
3. Take the modelandview Object to spring-servlet.xml

Entity (Bean)

EntityController (BeanController)

DAO (Interface, Implementation) (BeanRepository,BeanRepositoryImpl)

Service(Interface,Implementation) (BeanService,BeanServiceImpl)