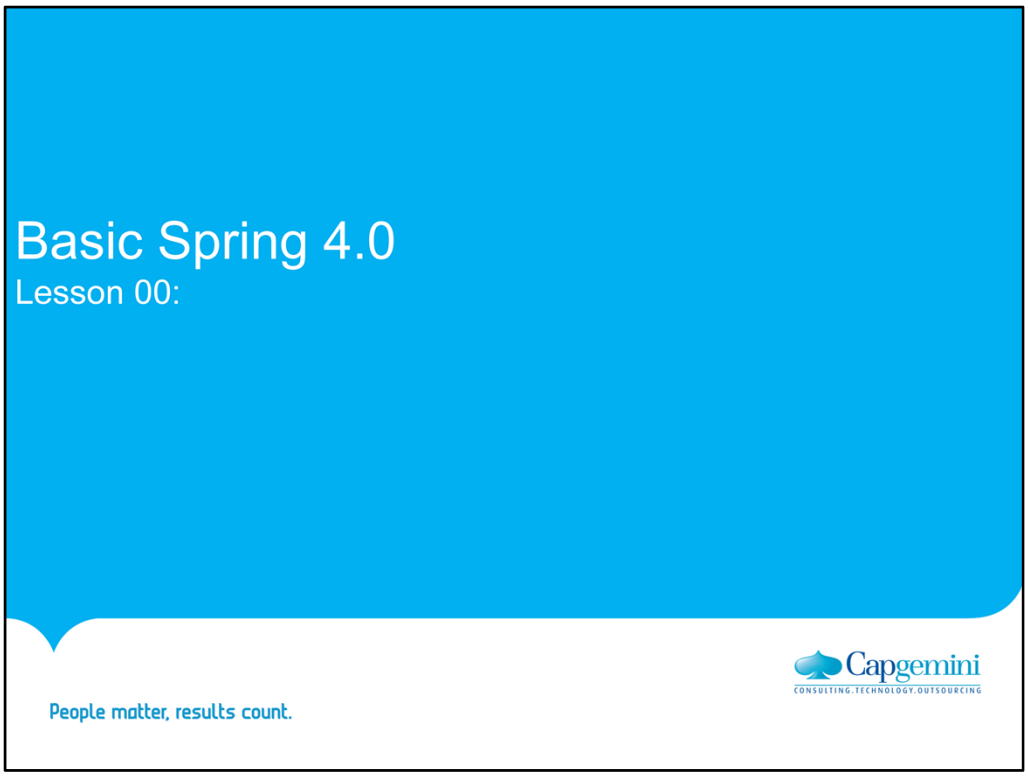


Document History

Date	Course Version No.	Software Version No.	Developer / SME	Change Record Remarks
21-Jul-2006	Ver1.0	1.2.6	Shrilata Tavargeri	NA
Jan-2009	Ver2.0	2.5	Shrilata Tavargeri	Changed material with new changes introduced in ver 2.5 and inputs from BU
Aug-2011	Ver 3.0	3.0	Shrilata Tavargeri	Changed material with new changes introduced in ver 3.0 and inputs from BU
June 2013	Ver 4.0	3.0	Mohan Chinnaiah	Revamped materials according to new requirements
May 2015	Ver 5.0	4.0	Rathnajothi Perumalsamy	Changed material with new changes introduced in ver 4.0
June- 2016	Ver 6.0	4.0	Vinod Satpute Yukti Valecha Tanmaya Acharya	Modified as per Toc for ELTP

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 2

Keep this as a hidden slide.
Note to co-ordinators: Not to be printed for the class book.

Course Goals and Non Goals

- Course Goals
  - Understand the benefits of using Spring
  - Understand the principles of IoC and AOP
  - Be able to use AOP to handle cross-cutting concerns
  - Connect business objects to persistent stores using Spring's DAO modules
  - Use the Spring MVC web framework to develop flexible web applications
  - Introduction to Spring Testing
- Course Non Goals
  - Design patterns, Spring Integration with different technologies

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 3

Pre-requisites

- Core Java , Java 8 features and JDBC
- XML, DBMS/SQL
- Servlets, JSP
- Concepts of MVC, Design patterns

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 4

Intended Audience

- All Java application developers especially Enterprise Java Programmers
- Software designers

Day Wise Schedule

- Day 1
  - Lesson 1: Introduction to Spring Platform and environment
  - Lesson 2: Introduction to Spring Framework, IoC
- Day 2
  - Lesson 2: Introduction to Spring Framework, IoC (Contd..)
  - Lesson 3 : SpEL (Spring Expression Language)
- Day 3
  - Lesson 4: Spring MVC framework
- Day 4
  - Lesson 5: Spring JPA Integration
- Day 5
  - Lesson 6: AOP (Aspect Oriented Programming)

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 6

Table of Contents

- Lesson 1: Introduction to Spring Platform and Environment
  - 1.1 Introduction to Spring Platform and environment
  - 1.2 Introduction to Spring
  - 1.3 Spring Projects at a glance
  - 1.4 Spring IO Platform
    - 1.4.1 Spring Framework
    - 1.4.2 Spring Boot
- Lesson 2: Introduction to Spring Framework, IoC
  - 2.1 What is Spring Framework, Benefits of Spring
  - 2.2 The Spring architecture
  - 2.3 IOC – Inversion of control, wiring beans
  - 2.4 Bean containers, lifecycle of beans in containers
  - 2.5 Customizing beans with PostProcessors
  - 2.6 Annotation-based configuration

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 7

Table of Contents

- Lesson 3: Introduction to SpEL (Spring Expression Language)
  - 3.1 SpEL Expression fundamentals
  - 3.2 Expression Language features
  - 3.3 Reduce configuration with @Value
- Lesson 4: Spring MVC framework
  - 4.1 Introduction: DispatcherServlet, Handler mappings, Resolving views
  - 4.2 Annotation-based controller configuration
  - 4.3 Introduction to REST web Services
  - 4.4 REST Controllers on the top of MVC

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Table of Contents

- Lesson 5: Spring JPA Integration
  - 5.1 Spring support for JPA
  - 5.2 Implementing Spring JPA integration
  - 5.3 Spring Data JPA
- Lesson 6: AOP (Aspect Oriented Programming)
  - 6.1 AOP concepts
  - 6.2 AOP support in Spring using @AspectJ support
  - 6.3 AOP support in Spring using Schema-based AOP support

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 9

References

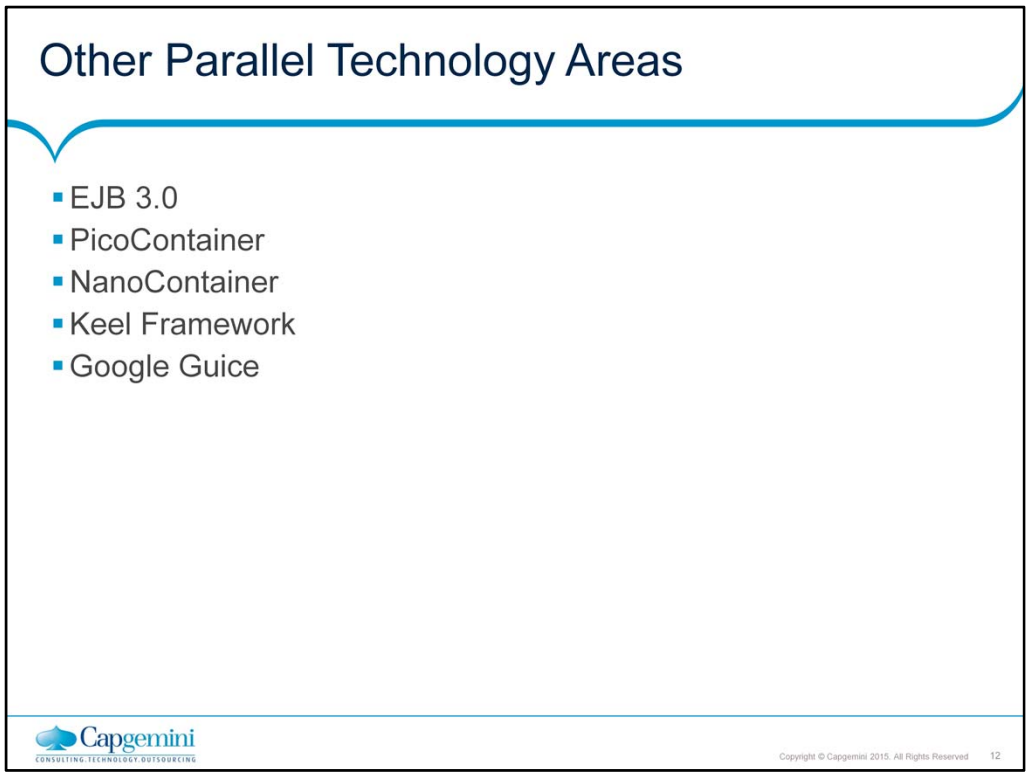
- Spring in Action, Fourth Edition, Manning publications by Craig Walls
- Spring-framework-reference.pdf from SpringSource (this is available in the downloaded Spring software)

Software required

- JDK version 1.8 + with help, Netscape or IE
- MS-Access/Connectivity to Oracle database
- WildFly
- Eclipse Luna
- Spring 4.0 API with docs

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 11



PicoContainer: is an exceptionally small DI (Dependency Injection) container that allows to use DI for your application without introducing any dependencies other than PicoContainer itself
NanoContainer: is an extension to PicoContainer fro managing trees of individual PicoContainer containers.
Keel Framework: is more of a metaframework, in that most of its abilities come from other frameworks that are all brought together under one roof.
Google Guice: focuses purely on DI.