



Capstone Courseware, LLC

33 Boylston Street  
Jamaica Plain, MA 02130

877-227-2477  
capstonecourseware.com

## 117A. The Spring Framework

### Version 2.0.1

This course enables the experienced Java developer to use the Spring Application Framework to manage objects in a lightweight "IoC" (inversion-of-control) container; to create simple and complex Web applications; and to manage persistent objects using Spring's support for DAOs and transaction control. Spring is a far-reaching framework that aims to facilitate all sorts of Java development, including every level of multi-tier distributed systems. Here we focus on the Core and MVC modules, with a lighter (but by no means dismissive) touch on persistence through DAO and ORM modules.

The Core module gives the developer declarative control over object creation and assembly; this is useful for any tier of any Java application. So is Spring's validation framework, and so we study these things in a mix of standalone (J2SE) applications and Web applications deployed to the Tomcat server/container. Then students build Web applications that use the Spring MVC framework to rationalize their designs into coherent request/response cycles. They use Spring command objects to manage HTML forms and their data, and connect these to the validation framework. We connect our applications to persistent stores and study the DAO and ORM modules, to better understand JDBC and Hibernate persistence models and declarative transaction control.

This course is a variant of Course 117. To serve a range of audiences who are interested in different parts of the Spring framework and who have different backgrounds, we're maintaining three variants. Course 117 covers the same basic material but has a few less code exercises and follows a more aggressive, three-day timeline. Course 117B has the additional exercises and more relaxed pace found in this course, but de-emphasizes the Web module for a three-day timeline as well.

A bridge module is also available that illustrates how Spring's persistence module compliments the Hibernate ORM framework; this brief presentation



can easily be added to the end of the class and works especially well to cap off a week of training using this course and our Hibernate course.

## Prerequisites

- Java programming -- Course 103 is excellent preparation.
- Servlets programming -- Course 110.
- JSP -- Course 112.
- Basic knowledge of XML -- Course 501.





## Learning Objectives

- Understand the scope, purpose, and architecture of Spring
- Use Spring's Inversion of Control to declare application components, rather than hard-coding their states and lifecycles
- Use Dependency Injection to further control object relationships from outside the Java code base
- Create validators for business objects, and associate them for application-level and unit-testing uses
- Build a Web application as a Spring DispatcherServlet and associated application context, with declared beans acting as controllers, command objects, and view resolvers
- Build and manage HTML forms with Spring command objects and custom tags
- Use Spring interceptors to implement horizontal features in the Web application
- Connect business objects to persistent stores using Spring's DAO and ORM modules

**Timeline: 4 days.**

## IDE Support: Eclipse WTP 1.5

In addition to the primary lab files, an optional overlay is available that adds support for Eclipse WTP 1.5. Students can code, build, deploy, and test all exercises from within Eclipse, and take advantage of WTP's built-in editors and wizards for web applications, XML files, JSPs, and more. See also our orientation to Using Capstone's Eclipse Overlays, and please be advised that this is an optional feature; it is not a separate version of the course, and the course itself does not contain explicit Eclipse-specific lab instructions.





## Chapter 1. Overview

- Web Applications
- J2EE: The Good, The Bad, and the Ugly
- Enter the Framework
- Spring Modules
- Controlling Object Creation
- Web Applications
- Persistence Support
- Aspect-Oriented Programming
- Integrating Other Frameworks

## Chapter 2. Core Techniques

- Component-Based Software
- JavaBeans, Reconsidered
- The Factory Pattern
- Inversion of Control
- XML View: Declaring Beans
- Java View: Using Beans
- Singletons and Prototypes
- Initializing Bean State

## Chapter 3. Dependency Injection

- Complex Systems
- Assembling Object Graphs
- Dependency Injection
- Single and Multiple Relationships
- The Utility Schema
- Autowiring
- Bean Aliases
- Order of Instantiation

## Chapter 4. Validation

- Validators
- The Errors Object
- ValidationUtils
- Error Messages and Localization
- Nested Property Paths

## Chapter 5. The Web Module





- Servlets and JSPs: What's Missing
- The MVC Pattern
- The Front Controller Pattern
- DispatcherServlet
- A Request/Response Cycle
- The Strategy Pattern
- JavaBeans as Web Components
- Web Application Contexts
- Handler Mappings
- "Creating" a Model
- View Resolvers

### **Chapter 6. Customizing Spring MVC**

- HandlerMapping Options
- ViewResolver Options
- Chaining View Resolvers
- Triggering Redirects

### **Chapter 7. Controllers and Commands**

- Working with Forms
- Command Objects
- The Template Method Pattern
- Command Controllers
- Data Binding
- MultiActionController
- Scope and Granularity of Command Objects

### **Chapter 8. Web Binding and Validation**

- Property Editors
- Custom Property Editors
- Registrars
- Validating Form Input

### **Chapter 9. Form Controllers**

- Form Controllers
- AbstractFormController
- SimpleFormController
- Spring Custom Tags
- <form:form> and Friends





<form:errors>  
Reporting Errors

## Chapter 10. Refining the Handling Cycle

The Intercepting Filter Pattern  
Exception Handling  
Interceptors  
The Decorator Pattern  
Context and Lifecycle  
Awareness Interfaces  
Support and Utility Classes  
"Death By XML"

## Chapter 11. The Persistence Tier

The DAO Pattern  
The DaoSupport Hierarchy  
The DataAccessException Hierarchy  
JDBC DAOs  
JdbcTemplate and RowMapper  
Object/Relational Mapping  
Hibernate DAOs  
Transaction Control  
AOP vs. Annotations

## Appendix A. Learning Resources

### System Requirements

<b>Hardware Requirements (Minimum)</b>	500 MHz, 256 meg RAM, 500 meg disk space.
<b>Hardware Requirements (Recommended)</b>	1.5 GHz, 512 meg RAM, 1 gig disk space.
<b>Operating System</b>	Tested on Windows XP Professional. Course software should be viable on all systems which support a J2SE 5.0 JDK.
<b>Network and Security</b>	Limited privileges required -- please see our standard security requirements at <a href="http://capcourse.com/Guides/Security.html">http://capcourse.com/Guides/Security.html</a> .
<b>Software Requirements</b>	All free downloadable tools.

