Lesson 1 Java MVC framework
Servlets
Web.xml
JSP
- Scriptlets
- Expression language
- JSTL

MVC model 2 architecture

You need to know how a MVC model 2 application looks like.
You should be able to implement the servlet, the JSP (only between <body> tag) and the model class.
You should be able to use the following JSTL tags:
- <c:out>
- <c:foreach>
- <c:if>

Lesson 2 Java MVC framework
Front controller
Validation
Declarative routing
Reflection
Dependency injection
Annotation based configuration

You need to know what is involved to build your own MVC framework.
You need to know what reflection is, but you don't need to implement it.
You need to know how you can write your own annotation in Java

Lesson 3 Spring MVC
@Controller, @Service. @Repository, @Component
@RequestMapping
@RequestParam
@PathVariable

Dependency Injection: @Autowired

You need to know how SpringMVC works (internals)
You need to be able to write a Spring MVC controller and use these annotations at the right place

Lesson 4 Spring MVC Tag library
Data binding
@ModelAttribute
Spring Data Formatters

You need to be able to implement and use a modelattribute.
You should be able to use the following spring form tags:
- <form>
- <input>
- <select>, <option>
- <message>

You need to know how to write your own customer formatter

Lesson 5 Spring MVC State

Request scope
Session Scope
Context scope
@SessionAttributes
Post/Redirect/Get (PRG) Pattern
Forward & Redirect
Flash Attributes
Model, ModelMap, ModelAndView

You need to understand and use the different scopes.
You need to be able to implement the PRG pattern

Lesson 6 Spring MVC validation
HandlerMapping
Interceptors
@Valid and BindingResult
JSR330 validation constraint annotations @Size @NotEmpty @Min @Max @NotNull
<form:errors path=…
Custom validator

You need to understand and implement handlermapping and interceptors in a Spring MVC application
You need to understand and implement validation in a Spring MVC application

Lesson 7 Spring MVC View resolvers
UrlBasedViewResolver
ContentNegotiatingViewResolver
SpringMVC exception handling
@ExceptionHandler
@ControllerAdvice
You need to understand it, and you need to know how to apply this

Internationalization with Locale
Locale resolvers
Tiles
You need to understand these concepts.
Understand the high level idea.
You don't need to implement these concepts yourself

You **don't** need to know the different <%@taglib ..> tags:
<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>

You **don't** need to know details of the spring XML configuration file