



Capstone Courseware, LLC

33 Boylston Street
Jamaica Plain, MA 02130

877-227-2477
capstonecourseware.com

542. XML Programming Using Java

Version 6.0

This four-day course builds skills in Java's XML processing APIs. The course is intended for students with a working knowledge of XML -- and possibly DTDs or XML Schema -- who want to build XML applications or components using the Java language. Everything in the course adheres to W3C and Java standards for completely portable code.

The course comprises three modules, which treat XML parsing, XML transformations, and XML binding. The first module introduces the Java API for XML Processing (JAXP) and the two main Java APIs for parsing XML documents: SAX and the DOM. Students learn the basic JAXP architecture and how to create parsers that expose SAX or DOM APIs, and how to configure parsers according to the SAX features and properties specification. SAX parsing gives way to reading and writing document information using the DOM tree model and API.

The second module introduces students to the XPath and XSLT specifications, and how to use JAXP as an interface to XML addressing and transformations. Students learn the JAXP's innovative system of Transformers, Sources, and Results; develop fluency in the exacting but powerful XPath syntax; and then build a number of XSLT transformations.

The third module introduces the newer and more powerful Java API for XML Binding, or JAXB, in its current incarnation, which is version 2.1. JAXB is now emerging as the tool of choice for most XML processing tasks where an XML Schema is available or easy to derive: based on that schema, JAXB-generated types can make it quite easy to read, manipulate, and write XML information models.

Prerequisites

- Experience in Java Programming, including object-oriented Java and the Java streams model, is essential. Course 103, "Java Programming," is excellent preparation.
- Basic understanding of XML is required. Course 501, "Introduction to XML," is recommended.
- XML Schema is used peripherally in the course, and knowledge of this



technology will be helpful, but is not required.





Learning Objectives

- Understand the use of SAX and DOM APIs for XML parsing.
- Understand the need for JAXP as an additional layer to the standard contract between applications and parsers.
- Parse element and attribute content, processing instructions, and other document information using SAX.
- Parse documents using the DOM.
- Modify, create and delete information in an XML document using the DOM.
- Write simple and complex queries into XML document content using XPath.
- Use XSLT for XML-to-XML, XML-to-HTML, or XML-to-text transformations.
- Build standalone and web applications that build on XSLT transformation logic.
- Use JAXB to generate persistent Java object models based on XML Schema.
- Implement XML persistence using the JAXB.

Timeline: 4 days.

IDE Support: Eclipse Europa

In addition to the primary lab files, an optional overlay is available that adds support for Eclipse Europa. Students can code and build all exercises from within the IDE. Some exercises can be tested from within the IDE; many must be tested from the command line. 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.





Module 1. XML Parsing Using Java

Chapter 1. The Java API for XML Processing (JAXP)

- Parsing XML
- SAX and DOM
- What the W3C Says
- What the W3C Doesn't Say
- Sun and Apache
- JAXP
- Parser Factories
- Pluggable Parsers
- Parser Features and Properties

Chapter 2. The Simple API for XML (SAX)

- Origins of SAX
- The SAX Parser
- The SAX Event Model
- Reading Document Content
- Handling Namespaces
- SAX Features for Namespaces
- Parsing Attributes
- Error Handling
- DTD Validation
- Schema Validation
- Handling Processing Instructions

Chapter 3. The Document Object Model (DOM)

- Origins of the DOM
- DOM Levels
- DOM2 Structure
- The Document Builder
- DOM Tree Model
- DOM Interfaces
- Document, Node and NodeList Interfaces
- Element and Text Interfaces
- Finding Elements By Name
- Walking the Child List
- The Attribute Interface





- Traversing Associations
- The JAXP Transformer Class
- Sources and Results
- Combining SAX and DOM Processing
- Namespaces and the DOM

Chapter 4. Manipulating XML Information with the DOM

- Modifying Documents
- Modifying Elements
- Modifying Attributes
- Managing Children
- Seeking a Document Location
- The ProcessingInstruction Interface
- Creating New Documents





Module 2. XML Transformations Using Java

Chapter 1. Using the JAXP for Transformations

- XPath, XSLT and Java
- The Transformer Class
- The TransformerFactory Class
- Sources and Results
- Identity Transformations
- Creating Transformations from Stylesheets
- Template Parameters
- Output Methods and Properties
- Evaluating XPath Expressions

Chapter 2. XPath

- Use of XPath in Other XML Technologies
- XPath Expressions
- The Axis
- The Node Test
- The Predicate
- XPath Types
- XPath Functions
- Implied Context
- Querying with XPath
- XPath and the DOM

Chapter 3. Templates and Production

- Rule-Based Transformations
- Templates and Template Matching
- Built-In Template Rules
- Recursion Through Templates
- Template Context
- Output Methods
- Controlling Whitespace
- Literal Replacement Elements
- Formalizing Text, Elements and Attributes
- Defining Target Vocabulary
- Generating Processing Instructions





Chapter 4. XSLT: Dynamic Content and Flow Control

- Web Applications Using XSLT
- J2EE and JAXP
- Deriving Source Content
- Getting Source Values
- Attribute Value Templates
- Copying Source Elements and Trees
- Looping
- Conditionals





Module 3. The Java API for XML Binding

Chapter 1. The Java API for XML Binding

- The Need for Data Binding
- XML Schema
- Two Paths
- JAXB Compilation
- Mapping Schema Types to Java
- Java-to-XML Mapping Using Annotations
- Marshaling and Unmarshaling
- Working with JAXB Object Models

Appendix A. Learning Resources

Appendix B. Quick Reference: W3C Namespaces

System Requirements

Hardware Requirements (Minimum)	500 MHz, 256 meg RAM, 500 meg disk space.
Hardware Requirements (Recommended)	1.5 GHz, 512 meg RAM, 1 gig disk space.
Operating System	Tested on Windows XP Professional. Course software should be viable on all systems which support a Java 6 Developer's Kit.
Network and Security	Limited privileges required -- please see our standard security requirements at http://capcourse.com/Guides/Security.html .
Software Requirements	All free downloadable tools.

