



Capstone Courseware, LLC

33 Boylston Street  
Jamaica Plain, MA 02130

877-227-2477  
capstonecourseware.com

## 542. XML Programming Using Java

### Version 1.4

This four-day course builds skills in Java's XML processing APIs -- chiefly parsing using SAX and DOM and transformations using XSLT, all using the Java API for XML Processing, or JAXP. It also covers the newer Java API for XML Binding, or JAXB, which standardizes serialization according to XML Schema. 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 first module introduces the 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 is covered, working from simple SAX event handling through patterns for understanding document content from event sequences, to error handling and document validation. Students then learn how to read document information using the DOM's tree model and API, and move on to using the DOM to modify and to create new documents and information nodes. The final two chapters of this module cover the DOM2 Traversal and Events modules and the JAXB, with a focus on XML serialization and persistence.

The second module introduces students to the XPath and XSLT specifications, and how to use JAXP as an interface to XML transformations. Students learn the basic JAXP Transformer architecture, develop fluency in the exacting but powerful XPath syntax, and then build a number of XSLT transformations. Study of XSLT is arranged first to develop control over output production, including a solid understanding of the sometimes mysterious built-in template rules, template matching, priority and modes, and control of whitespace production. Then students turn towards the source document and learn to extract single values, to make shallow and deep copies of source elements, to use variables, and to use flow-control constructs to effect conditional processing and loops. In the module's final case study, students build a servlet-based Web application that uses JAXP and XSLT to produce dynamic content based on an XML data source.

The course software also includes an optional overlay of workspace and project files to support use of the Eclipse IDE in the classroom. (This requires that the instructor be experienced in use of Eclipse and able to walk students through basic



tasks in the IDE.)

## 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.
- Use JAXP to write entirely portable XML parsing code.
- 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.
- Use DOM Traversal to simplify and make parsing algorithms more effective.
- Use DOM Mutation Events to track changes to an XML document.
- Use the JAXB to generate persistent Java object models based on XML Schema.
- Implement XML persistence using the JAXB.
- Write simple and complex queries into XML document content using XPath.
- Use XSLT for XML-to-XML transformations.
- Use the built-in template rules correctly to process the right source information.
- Use mode and priority to control template matching.
- Control exact production of text, HTML and XML elements, and whitespace.
- Derive source document content and make copies of node trees.
- Use looping and conditional processing to manage output production.
- Build J2SE and Web applications that leverage XSLT transformation logic.

**Timeline: 4 days.**

### IDE Support: Eclipse 3.2

In addition to the primary lab files, an optional overlay is available that adds support for Eclipse 3.2. Students can code and build all exercises from within the IDE. Some exercises can be tested successfully from within the IDE, but most 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

#### **Chapter 5. DOM Level 2 Modules**

- DOM Traversal
- The DocumentTraversal Interface
- Node Filters
- The NodeIterator Interface
- The TreeWalker Interface
- DOM Events
- Mutation Events
- Handling Events
- Event Flow
- Capturing and Bubbling
- Cancelable Events

#### **Chapter 6. XML Serialization and the Java API for XML Binding (JAXB)**

- XML for Object Persistence
- Persistence Strategies
- The Memento Pattern
- Deserialization with SAX
- Object Persistence with the DOM
- Adapting Object Models to the DOM
- The Java API for XML Binding
- Marshalling, Unmarshalling, and Validation
- Schema as Object Models
- UML for XML
- Mapping XML to Java: Simple Types, Complex Types, and Collections





## 542. XML Programming Using Java

Outline

- Object Factories
- Customizing JAXB Bindings
- The DOM vs. JAXB
- JAXB for Persistence
- Automatic Translation





## **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

### **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

**Appendix A. Learning Resources**

**Appendix B. Quick Reference: W3C Namespaces**

**Appendix C. UML for XML Schema**

**Appendix D. Quick Reference: XML and DTD Grammar**

**Appendix E. Quick Reference: XPath and XSLT**

**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 1.4 SDK.
<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.

