

## Mastering XML, XSLT and XSL using Java (JAXP) (TT4380)

**Length:** 5 days

### Description

This course is an intensive, hands-on treatment of how to construct, generate, and transform XML within Java and J2EE applications. The course is a balanced mixture of theory and practical labs designed to take students from the basic fundamentals of XML processing through related advanced technologies such as Java/XML interoperability. The students study the various APIs in a structured manner to enable them to master the concepts, ideas, and patterns, which are reinforced in the lab exercises.

### Audience

This is an **intermediary** level Java programming course, designed for those needing to process XML from within Java applications.

### Prerequisite

Previous experience or knowledge of HTML is helpful. Students are expected to know the Java programming language.

### Topics

#### XML CONTENT

##### *XML Overview*

- What is XML?
- History: SGML; HTML
- XML Author Roles; Tool Roles
- XML Applications

##### *XML Mechanics*

- XML Document Structure
- Well-Formed and Valid XML Documents
- Structure, Content and Format

#### XML STRUCTURE

##### *Structure Using DTDs*

- Document Type Definition (DTD)
- DOCTYPE Statement
- Complex Content Models

##### *Namespaces*

- Namespaces Associate an Alias with a URI
- Declaring Namespaces
- Namespaces Best Practices

##### *Structure Using Schemas*

- XML Schemas: Objectives
- Schemas Data Types
- Complex Types
- Schema Components
- Global and Local Components
- Derived Complex Types
- Associating Schemas with XML Instances
- Reuse and Manageability of Schemas
- Schema Composition

#### XML FORMATTING

##### *CSS and Rendering XML*

- XML's Impact on Styling

## Mastering XML, XSLT and XSL using Java (JAXP) (TT4380)

- Cascading Style Sheets and XML
- Rendering Exceeds CSS's Abilities

### *XSL Transformations*

- XPath Describes Locations Within XML
- XSLT is Rule-Based Transformation Language
- XSL is Oriented Towards Formatting
- XPath Accesses Parts of Document
- XSLT Templates Specify Output Replacement
- XSLT Uses XPath Expressions Within Templates

### *XSLT and XPath*

- XPath Expressions
- Abbreviated Axis Forms
- Predicates As Optional Filters
- XPath Operators; Functions; Examples
- Working With XPath
- XSLT Stylesheet Structure
- Templates: Rules in a Stylesheet
- Apply-Templates Directs Processing
- value-of to Extract Values
- Built-in Templates
- Text Handling
- Calling Templates
- Passing Parameters
- Conditional Processing Constructs
- Looping With <xsl:for-each>
- Sorting
- Constructing A New Node

### *XSL FO (Formatting Objects)*

- XSLT Designed to Support XSL-FO
- XSL Family Working Together
- XSL-FO Support for Paged Media

## JAXP AND SAX PARSING

### *JAXP and SAX (Part 1)*

- Processing XML Using Parsers
- Validating and Non-validating
- JAXP Overview
- JAXP Usage Patterns
- SAX Parsing Using JAXP

### *JAXP and SAX (Part 2)*

- SAX Events
- SAX API
- Handling Exceptions
- SAX Content Handling
- Designing SAX Content Handlers

## JAXP AND DOM PARSING

### *JAXP and DOM (Part 1)*

- JAXP Usage Patterns with DOM
- DocumentBuilders
- DOM Parsing Using JAXP

### *JAXP and DOM (Part 2)*

- DOM Concepts
- DOM Nodes
- DOM API
- Handling Exceptions
- DOM Operations and Processing
- Designing DOM Processing

## RENDERING: JAXP AND TRANSFORMATIONS

### *XSLT Transformations in Java*

- JAXP/TRaX Concepts and Terminology
- Transformer Patterns
- Stream to Stream Transformations
- DOM to DOM Transformations
- Exception Handling with Transformers
- Identity Transforms
- Performance Considerations
- Working With XSLT Processors

## Mastering XML, XSLT and XSL using Java (JAXP) (TT4380)

### **BINDING JAVA AND XML**

#### *XML Interoperability with Java: JAXB*

- JAXB Concepts
- JAXB Class Generation
- JAXB Runtime Features
- On-demand Validation
- Consuming XML Using JAXB
- Generating XML Using JAXB

### **SECURITY AND JAXP**

#### *XML Signature, Encryption, and XWSS*

- Concepts and Terminology
- XML Signature
- XML Encryption
- XWSS: XML Security
- Processing Unsafe XML
- Vulnerabilities and Attacks

### **BEST PRACTICES**

#### *Performance Issues*

- Review of Best Practices