

# Java SE, Part 4 of 4: XML, Security, and JUnit

page 1

**Meet the expert:** Susan Bryant is an experienced IT trainer and consultant with a broad array of skills. She has over 15 years experience in information systems with roles including systems consulting, project management, staff management, staff mentoring, and certified technical trainer. Susan has a strong technical knowledge of IBM WebSphere Application Server, WebSphere Portal Server, WebSphere Process Server, Lotus Domino, and web application development technologies including, Struts, JSF, EJBs and AJAX.

**Prerequisites:** This course is intended for programmers who are familiar with Java and want to learn about the advanced features of Java. Students should have a good understanding object-oriented programming using Java.

**Runtime:** 06:40:59

**Course description:** This course explores Java SE topics that provide the finishing touches when building professional Java applications using Eclipse. Students will learn about parsing XML documents using both SAX and DOM methodologies. Students will understand how to take advantage of distributed objects, add security, use the Java Native Interface (JNI) and utilize Test-Driven Design through JUnit. This course concludes with a detailed look at Java features introduced with recent versions of Java.

## Course outline:

### Architecture Design

- Introduction
- Tiered Architectures
- 2 Tier Architecture
- 3 Tier Architecture
- N Tier Architecture
- Presentation Layer
- HTML Clients
- Java Server Pages
- Middle Tier
- Java Applications
- Enterprise JavaBeans
- COM/CORBA
- Message Oriented Middleware
- Demo: Whiteboard
- Summary

### Architecture Technologies

- Introduction
- Model View Controller
- Advantages/Disadvantages
- Model 1 Architecture
- Model 2 Architecture
- Extensible Markup Language
- XML
- Demo: Whiteboard
- Summary

### SAX Parsing

- Introduction
- SAX
- How it Works

- Core SAX2 Handler Classes
- SAX2 DefaultHandler
- SAX Events
- Ignorable Whitespace
- XML Reader Interface
- XMLReader Features
- XMLReader Factory
- Prepare SAX Parser Object
- Parse XML with SAX - Steps
- Define an Event Handler
- Prepare SAX Parser Object
- Demo: SAX Parsing
- Summary

### SAX Parsing 2

- Introduction
- Define an Event Handler
- Element Attributes
- Get Number of Attributes
- Get Name of Attributes
- Get Attribute Values
- Get Attribute Types
- Example: Define Event Handler
- Characters
- Using Characters
- Error Handling
- ErrorHandler Interface
- Parse XML Document
- Run the SAX Application
- Entity Resolver

- Locator
- Document Locator
- Demo: SAX Parsing 2
- Summary

### DOM Parsing

- Introduction
- DOM
- Limitations of SAX
- XML as an Object Model
- DOM Parsing
- Parse XML with DOM - Steps
- Prepare DOM Parser Object
- Parse XML Document
- Parse Exceptions
- Example: SimpleDOMParser
- Demo: DOM Parsing
- Summary

### DOM Parsing Elements

- Introduction
- Nodes
- Basic Node Types
- Less Common Node Types
- Node Interface
- Document Interface
- NodeList Interface
- Element Interface
- Attr Interface
- Text Interface
- Writing DOM
- Demo: DOM Parsing Elements

- Summary

### Distributed Objects RMI

- Introduction
- Serialization
- Example: Serialization
- Externalizable
- Remote Method Invocation
- Steps to implement RMI
- Remote Object Registry
- Dynamic Class Loading
- RMI and Applets
- Demo: Distributed Objects RMI
- Summary

### Java Security

- Introduction
- Attacks and Dangers
- JDK Security Features
- Concepts of Computer Security
- Encryption
- Cryptography Algorithm
- Message Digest
- Symmetric Ciphers
- Asymmetric Ciphers
- Digital Signature
- Authentication
- Certificate Manipulation
- Java Cryptography Architecture
- Java Cryptography Extension
- Using the MessageDigest Class
- Example: MessageDigest Class

(Continued on page 2)

# Java SE, Part 4 of 4: XML, Security, and JUnit

page 2

- Demo: Java Security
- Summary

## Java Security Architecture

- Introduction
- Using the Signature Class
- Java Security Architecture
- JDK 1.0 Security Model
- JDK 1.1 Security Model
- JDK 1.2 Security Model
- JDK 1.4 Security Enhancement
- Protection Domains & Security
- ProtectionDomain Class
- Permission Classes
- Using Permission Classes
- Policy Class
- Policy Configuration File
- AccessController Class
- SecurityManager Class
- Using the SecurityManager Class
- Dynamic Class Loader
- Example: Security Check
- Java Security Tools
- Using Java Security Tools
- Demo: Security Architecture
- Summary

## Java Native Interface

- Introduction
- JNI Overview
- JNI Architecture
- Calling C Functions
- The Header File
- Passing Simple Parameters
- Mapping Java Types
- Calling Java Methods
- The Invocation API
- Exception Handling
- Native Exception to Java
- Java Exception in Native Code
- Code Sample
- Demo: Java Native Interface
- Summary

## JUnit Basics

- Introduction
- What is JUnit
- Why JUnit
- The xUnit Philosophy
- Test-Driven Design
- A JUnit Test
- Running the Tests
- Swing-based Test Runner
- Text-based Test Runner
- JUnit Basics

- assertTrue
- assertEquals
- assertSame
- assertNull
- The Failure Message
- The Test Class
- The Test Method
- Demo: JUnit
- Summary

## Test Suite and Test Strategies

- Introduction
- The Test Suite
- JUnit with Annotations
- JUnit 4 Test Suite
- JUnit Design
- Testing Strategies
- Testing Simple Java Classes
- Testing with Databases
- Testing Web Applications
- JUnit with Ant
- Demo: Test Suite & Strategies
- Summary