In this course, you will be introduced to the fundamentals of Service Oriented Architectures (SOA) and the core standards that enable web services. You will focus on what separates an ad hoc set of web services (NOT SOA) from a managed, vibrant, reusable catalog of enterprise services (definitely SOA). You will examine the entire spectrum from the promise of cloud computing to the grit of XML content. You will get a clear picture of how a service orientation can fundamentally change the dynamics of how software is developed and "lives" within an enterprise. You will gain a better understanding of the fundamental technologies used in web services. This understanding is critical to being able to diagnose, troubleshoot, tune, and perform other life-cycle activities. You will leave the course armed with the required skills to design, implement, test, and support J2EE web services.

Skills Gained

- Basic concepts of SOA to the identification and design of web services
- Web services and the core technologies involved
- Concept of layered services including orchestration
- Design, develop, and deploy real-world J2EE web services
- Expose existing Java components as XML web services
- Write Java components that access remote Web Services hosted by a third party
- WSDL (Web Services Description Language) document
- Parse, process, and respond to a SOAP message
- Concepts behind Representational State Transfer (REST) and implementation of a REST-based web service
- Implement handlers to inject cross-cutting solutions for security, logging, auditing, and other needs
- Work with WS-Security to protect content, resources, and other assets
- Utilize web service-related design patterns

Who Can Benefit

J2EE developers and architects who need to identify, design, and implement web services

Prerequisites

One to two years of working knowledge with Servlets and JSPs, and should be familiar with XML, Namespaces, and XML Schema

Course Details

SOA

- Architectural Style - Common Framework
- Loose Coupling - Spectrum of Options
- Software Agents - Services
SOA - The Business Proposition
- Motivation for SOA
- Typical Software Project
- Service Model
- Service Consumer
- Service Bus
- Commonality is Critical Element of SOA
- Service Provider
- Business Process: OpenCheckingAcct
- SOA Addresses
- SOA Help Deals with Change
- Leverage SOA to
- Benefits of SOA
- SOA Maturity Models Abound
- Incremental Adoption of SOA

SOA - An Architectural Perspective
- Enterprise Application Layers
- Application Layers at a Glance
- Services and their Formal Contracts
- Services Should be Stateless
- Objects, Components, and Services
- Challenge and Limitations of Legacy Systems

SOA - A Development Perspective
- Strategic Orientation
- Tactical Strategy
- Lifecycle Phases
- SOA Roles and Skills (Existing and New)
- Business is the Starting Point
- Service-Oriented Analysis and Design (SOAD)
- SOAD Process
- Service Identification
- Service Modeling Guidelines
Service-Oriented Architecture
- Technical Principles
- Logical Components of a SOA
- Business Process-Driven Development
- SOA Business Modeling and Reference Architecture
- Business vs. Application
- Service Layers
- Application, Functional, and Business Process Services
- Two Messaging Models
- Publish/Subscribe
- Logical View of Publish/Subscribe
- Point-to-Point (P2P)
- Message Servers
- SOAP
- Role and Uses of an Enterprise Service Bus
- Enterprise Service Bus
- Challenge and Solution of Handling Transactions
- Security
- Governance

Service-Oriented Analysis and Design
- Lifecycle Phases
- Service Identification
- Service Specification
- Results
- Service Realization
- SOA Reference Architecture
- Results and Realization
- Systematic Process to Achieving these Results

Web Services
- Crossing Boundaries
- Six Key Components
- Characteristics and Architecture
- Technology Comparison
- Enable Decoupling
- Challenges
- Secure Services
- Spec and Standard Evolution
- Interoperability Organization

**Web Services, Java, and J2EE**
- XML Signature
- XML Digital Signatures
- XML Encryption
- Java API for XML Processing (JAXP)
- Java Architecture for XML Binding (JAXB)
- JAX-WS
- SOAP with Attachments API for Java (SAAJ)
- JAX-WSA and XWSS
- APIs
- J2EE and Web Services
- Web Services Metadata
- Web Services Interoperability Technologies (WSIT)
- JBossWS
- Java Web Services Developer Pack (JWSDP)
- WebSphere WS

**Web Services Quickstart**
- WSIT Tool Support
- How is WSIT Used?
- Web Service Development with WSIT
- Debugging Web Services
- TCP/IP Monitors Provide View of Wire

**XML, Namespaces, and Schema**
- XML
- XML Application-Specific Information
- Content
- Structure
- XML Transformation to HTML
- XML Separates Structure, Content, and Format
- XML Namespaces
- W3C XML Schemas

**XML in Java - JAXP and JAXB**
- XML Parsers
- XML and Java
Simple Object Access Protocol (SOAP)
- Anatomy of a SOAP Message
- SOAP and HTTP
- Typical SOAP Scenario
- SOAP Specification Provides
- Uses of SOAP
- Remote Procedure Calls

SOAP in Detail
- Protocols Used with Web Services
- Request and Response
- SOAP Envelope
- SOAP Header
- SOAP Body
- SOAP Request/Response Example
- SOAP Defines a Fault Element
- SOAP Data Model
- Using XSD Data Type
- SOAP Styles and Modes
- SOAP with Attachments
- SOAP Messaging
- Endpoint Behavior
- SOAP Encoding Styles
- SOAP 1.1 vs. SOAP 1.2
- Connections
- Security Concerns Relative to SOAP

REST
- Characteristics
- Elements
- Web Service Terms
- REST Design Principles
- SOAP vs. REST
- RESTing in Java
- Security Concerns Relative to REST

**WSDL**
- Web Services
- WSDL in Practice
- WSDL Extensibility
- WSDL/SOAP Namespaces
- WSDL Elements
- WSDL Anatomy
- WSDL 2.0

**Accessing Web Services Using Java (JAX-WS)**
- Architecture
- JAX-WS vs. RMI
- JAX-RPC
- JAX-WS Basics
- JAX-WS Features
- Web Service Annotation
- JAX-WS Programming Model
- JAX-WS Handlers

**Working with JAX-WS**
- JAX-WS Service
- JAX-WS Development Process
- WSDL-Generated Service
- JAX-WS Client
- Static Web Service Client
- Dynamic JAX-WS Client
- DII JAX-WS Client
- SOAP Handlers

**Handlers**
- Basics
- WSEE and Handlers
- Handler Life Cycle
- Applying Handlers
- Handler Interface and Implementation
Web Services for J2EE (WSEE)
- WSEE and WSEE Server Programming Model
- Routing SOAP requests to an EJB
- WSDD
- WSEE Client Programming Model
- Client Programming Model
- Types of WSEE Clients
- Static WSEE Client
- Dynamic WSEE Client
- DII WSEE Client
- WSEE Client Packaging

Discovery
- Issues with Broadly Scoped Discovery
- Discovery Options
- UDDI

XML Signature and Encryption
- Cryptography Addresses Many Aspects of Security
- XML Challenges
- XML Signature
- Standard for Digital Signature
- XML Encryption

Web Services Security (WS-Security)
- Securing a Web Service
- Transport-Level Security
- Secure Sockets Layer (SSL)
- Using Transport-Level Security
- Message-Level Security
- Web Services Security Roadmap
- WS-Security Enables Interoperability
- Security Tokens
- Message Authentication
- XML Signature and Encryption
- XWSS
Securing Untrusted Input
- Input Data Attacks
- Protecting a Web Service
- Tenacious D
- Responding to Error State
- Best Practices for Untrusted Data
- Additional Types of Attacks

SOA Best Practices
- Planning
- Standardizing
- Designing
- Managing
- Implementing

SOA Patterns
- Direct Connections
- Broker Interactions
- Serial Process Flows
- Serial and Parallel Processes

SOA Anti-Patterns
- Service Identification and Design Anti-Patterns
- ServiceRealization Anti-Patterns