# Course Design and Outline

## 

| Learning Program | Duration |
| --- | --- |
| **Java Web Services** | 3-4 Days |

## Pre-Reading Material

1. XML : <http://www.itechcollege.com/courses/XML/>
2. XML Schema: <http://www.w3schools.com/xml/xml_schema.asp>
3. Web Services : <http://docs.oracle.com/javaee/6/tutorial/doc/gijti.html>
4. Web Services: <http://www.java2blog.com/2013/03/web-service-tutorial.html>
5. SOAP : <http://www.w3schools.com/xml/xml_soap.asp>

**Note: use CTRL+Mouse click to open theses links or paste them to the browser address bar**

# 

# Learning program overview

|  |  |
| --- | --- |
| Course name and description | * **Java Web Services** * This training is designed only for those participants who have previous experience in Java/Jee development and want to build, host Web Services and Web Services Clients on Java Platform |
| Course learning objectives | * Apply the JAX-WS API in the creation of SOAP Web Services and clients * Apply the JAX-RS API in the creation of RESTful Web Services * Handle errors and exceptions in Web Services and clients * Create XML documents using namespace declarations and XML schema * Produce and consume XML and JSON content using JAXB * Create RESTful Web Service clients using the Jersey Client API * Understand the role of Web Services |
| Effectiveness measure | * Assessments are built for all programs, team members mandatorily need to complete both pre and post assessments, else participation will be considered as void |

# Program outline

|  |  |  |
| --- | --- | --- |
| **Program Name** | **Java Web Services** | **Est. time** |
| **Instructional Strategy** | * In this course, we would first introduce the concept, followed by hands on activity in each of the lesson to ensure learners are able to apply the concept into the systems or application. * At the end of every learning objective we will present some quiz questions to participants. * The Assessment which is Post would be directly linked to the Learning Objectives of the program stated above. | |
| **Lesson** | **Topics** |  |
| **Introduction to Web Services** | * Why Web Services? * Service-Oriented Architecture * HTTP and XML * Simple Object Access Protocol (SOAP) * Web Service Description Language (WSDL) * Universal Description, Discovery and Integration (UDDI) * The WS-I Basic Profile * REST | 1.5 hour |
| **SOAP** | * Messaging Model * Namespaces * SOAP over HTTP * The SOAP Envelope * The Message Header * The Message Body * SOAP Faults * Attachments | 30 min |
| **Web Services Description Language (WSDL)** | * Web Services as Component-Based Software * The Need for an IDL * Web Services Description Language * WSDL Information Model * The Abstract Model - Service Semantics * Message Description * Messaging Styles * The Concrete Model - Ports, Services, Locations * Extending WSDL – Bindings. * Service Description | 2 hour |
| **J2SE,JavaEE and Web Services Support and Web Services API** | * Hosting Web Services: Scenarios * Invoking Web Services: Scenarios * J2EE and JAX-RPC * AXIS 1.4 (Overview) * J2EE Server Implementations * Web Services for Java EE (WS4JEE) * The Automated Approach: JAX-WS and JAXB * Manual Options: SAAJ and JAXP * Portable Web-Services Metadata * Overview of Metro and Axis2 | 1 hour |
| **JAXB** | * 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 * In-Memory Validation * Hands on session | 1.5 hours |
| **Web Services Development Approaches and Tools** | * Top Down approach (java to WSDL) * Bottom up approach (WSDL to Java) | 10 min |
| **JAX-WS** | * POJO Based Web Services   1. The @Web Service Annotation   2. Generated Code   3. Compilation and Assembly   4. Deployment   5. Runtime Behavior   6. Scope of Code Generation   7. Hands on session   8. More JAXB: Mapping Collections   9. More JAXB: Mapping Enumerations   10. The @Web Method, @XmlParam, and Related Annotations   11. Scope of Code Generation   12. More JAXB: Mapping Inheritance   13. Controlling the XML Model   14. Controlling the WSDL Description   15. Hands on session | 3 hours |
| **Exception Handling** | * Stubs and Proxies * Generated Code * Locating a Service * Invoking a Service * Hands on session | 3 hours |
| **EJB Web Services** | * Creating a web services from ejb3 Session bean * Deploying the ejb3 Web Service * Creating client for the deployed services * Invoking the Web Service using the client | 1hours |
| **Dispatch and Provider API** | * Stepping Down * The Provider<T> Interface * Implementing a Provider * JAXB Without WSDL * Integrating JAXP * The Dispatch<T> Interface * Building Clients * Hands on session | 2 hours |
| **Message Handlers** | * Handling SOAP Headers * Servlet Endpoint Context * Message Context and SOAPMessageContext * Message Handlers and Handler Chains * Processing Model and Patterns * Client-Side Handlers * Hands on session | 1.5 hour |
| **Handling Binary Content** | * The WS-I Attachments Profile * Attachments Metadata * JAX-WS Support: Binding MIME Types * SAAJ Support via JAF * The Message Transmission Optimization Mechanism * XML-binary Optimized Packaging. * Hands on session | 1 hours |
| **Restful Web Services with RestEasy (JAX-RS)** | * Introduction to REST   + REST and the Rebirth of HTTP   + RESTful Architectural Principles * Designing RESTful Services   + The Object Model   + Model the URIs   + Defining the Data Format   + Assigning HTTP Methods * Your First JAX-RS Service   + Developing a JAX-RS RESTful Service   + Deploying Our Service   + Hands on session * HTTP Method and URI Matching   + Binding HTTP Methods   + @Path   + Subresource Locators * JAX-RS Injection   + The Basics   + @PathParam   + @MatrixParam   + @QueryParam   + @FormParam   + @HeaderParam   + @CookieParam   + Common Functionality   + Hands on session * RESTful Java Clients   + java.net.URL   + Apache HttpClient   + RESTEasy Client Framework   + RESTEasy Client Proxies   + Hands on session * JAX-RS Implementations (Discussion)   + Jersey   + Apache CXF   + JBossRESTEasy | 4 hours |