



COURSE TITLE : Web Services
 COURSE PREREQUISITE : Introduction to Programming 1, Introduction to Programming 2, Data Structures, Web Programming
 COURSE DURATION : 16 weeks (3 hours/week)
 COURSE METHODOLOGY: Combination of lecture, lab exercises and written exercises.

Course Description

Web Services teaches students to enable applications to interact with one another over the web and integrate existing network computer infrastructure into the web.

Topics labeled * are advance topics but should be covered if time permits. Topics labeled ** are considered optional. Otherwise, the topic is mandatory.

Course Outline

Week	Topics
1	Lesson 1: Introduction to Web Services <ul style="list-style-type: none"> • How did web services came to be (A history of distributed computing) <ul style="list-style-type: none"> • Computing Advancement • What is a web service? • Why web services? • How can I use web services? • What comprises web services? • How does the web service work? • Future of web services
2-3	Lesson 2: XML Processing <ul style="list-style-type: none"> • XML Overview • A Simple Example • XML Schema Status and Motivation • Instance and Schema <ul style="list-style-type: none"> • Document Prolog, DOCTYPE, DTD • Element • Attribute • When to use Element, when to use Attribute • SimpleType • Complex Type • XML Handling • Document type and W3C XML Schema Definitions • Processing XML Document <ul style="list-style-type: none"> • SAX • DOM • Processing Phases • Namespaces • Transformation <ul style="list-style-type: none"> • XSLT

4-5	<p>Lesson 3: Accessing Web Service: SOAP</p> <ul style="list-style-type: none"> • Introduction to SOAP • Players in a SOAP Exchange • Anatomy of SOAP Message • Simple SOAP Message Example • The SOAP Specification <ul style="list-style-type: none"> • SOAP Envelope • SOAP Header • SOAP Body • SOAP Protocol Binding • SOAP Faults • SOAP Message Processing <ul style="list-style-type: none"> • SOAP Sender • SOAP Receiver • SOAP RPC Convention <ul style="list-style-type: none"> • Out and Inout parameters • Attachments with web services <ul style="list-style-type: none"> • Binary vs. Binary Data • MIME • DIME
6	<p>Lesson 4: Creating a service description using WSDL</p> <ul style="list-style-type: none"> • Introduction to WSDL • Application-to-application interaction • How does WSDL work? <ul style="list-style-type: none"> • Service Registry • Direct Access • Third-Party • Why WSDL? • WSDL Document Structure <ul style="list-style-type: none"> • Definition • Type • Message • PortType <ul style="list-style-type: none"> • One-Way • Request-Response • Solicit-Response • Notification • Binding • Service and Port • Tools • Future of WSDL
7	<p>Lesson 5: Web Service Registry</p> <ul style="list-style-type: none"> • Service Registry • What is UDDI? • Why UDDI? • How does UDDI work? • Scope of the Service Registry <ul style="list-style-type: none"> • Private Registry • Public Registry • Hybrid Registry • UDDI Specification <ul style="list-style-type: none"> • UDDI Data Types • UDDI API • Case Study • Java API for UDDI • JUDDI

	<ul style="list-style-type: none"> • UDDI 3.0.2 and the Future of UDDI
8-9	<p>Lesson 6: JAX-WS and JAX-RPC</p> <ul style="list-style-type: none"> • What is JAX-RPC? • Contrast JAX-RPC and JAX-WS <ul style="list-style-type: none"> • What is retained • What is changed • What is JAX-WS? <ul style="list-style-type: none"> • JAX-WS 2.0 Architecture • Introduction to Annotations <ul style="list-style-type: none"> • @WebService • @WebMethod • @WebParam • @WebResult • JAX-WS using Netbeans
10-11	<p>Lesson 7: JAXB (Data binding)</p> <ul style="list-style-type: none"> • XML Review • What and Why JAXB? • JAXB Architecture <ul style="list-style-type: none"> • XML Data Binding Facility • Binding Runtime API • Binding Process • Custom Binding <ul style="list-style-type: none"> • Why Custom Binding? • Name Collision Resolution • Runtime Operations <ul style="list-style-type: none"> • Unmarshaling • Creating In-Memory Content Programmatically • Processing: Accessing & Modification • Marshaling • Validation • JAXB Roadmap
12**	<p>Lesson 8: REST</p> <ul style="list-style-type: none"> • What is REST? • HTTP Methods • Why REST? • RESTful Web Services <ul style="list-style-type: none"> • A Simple Example • Principles of REST Web Service Design • REST vs. RPC <ul style="list-style-type: none"> • Contrast through example
13*	<p>Lesson 9: WS-Specifications</p> <ul style="list-style-type: none"> • WS-Addressing • WS-Routing • WS-Policy
14*	<p>Lesson 10: Web Application Architecture Web Services Support over J2EE</p> <ul style="list-style-type: none"> • SOA • BPEL • OpenESB • JBI • CASA • SCA • Composite Applications
15*	<p>Lesson 11: Web Services Design Patterns and Best Practices</p>

	<ul style="list-style-type: none">• Web Services Best Practice
16	<p>Lesson 12: Web Service Security</p> <ul style="list-style-type: none">• Introduction to Security<ul style="list-style-type: none">• Authentication• Authorization• Data Integrity• Confidentiality• Trust• Availability• Auditing• Why does web services require tighter security?• Basic Cryptographic Concepts• Security Policy Across the Service Stack<ul style="list-style-type: none">• Transport Level Security• Message Level Security• Application Level Security• SSL and Its Limitations• Implementing Web Service Security Solutions

