

DE ANZA COLLEGE
BUSINESS/COMPUTER SYSTEMS DIVISION
COURSE OUTLINE

Computer Information Systems 97YT

Effective Quarter Spring 2002

I. Catalog Information

CIS 97YT

Introduction to XML

Prerequisite: Computer Information Systems 89A or equivalent HTML experience.

Advisory: English Writing 100B and Reading 100 (or Language Arts 100), or English as a Second Language 24 and 72 (or English as a Second Language 4); Computer Applications and Office Systems 70AA

Three hours lecture, three hours laboratory.

II. Course Objectives

The student will:

- A. Write well-formed and valid XML.
- B. Use tools to verify well-formedness and validity.
- C. Create an ad-hoc markup language.
- D. Create a Document Type Definition for that language.
- E. Create a Relax NG or XML Schema definition for that language.
- F. Understand the concept of namespaces.
- G. Use Cascading Style Sheets to display XML.
- H. Understand the tree structure of an XML document.
- I. Learn XPath notation for selecting nodes from a document.
- J. Create XSLT transformations to convert XML to XHTML and/or XSL Formatting Objects
- K. Understand the concepts of Document Object Model (DOM) and Simple API for XML (SAX) parsers

III. Essential Student Materials

None

IV. Essential College Facilities

Access to a computer with Java runtime; open-source tools for XML: Xalan, Xerces, FOP.

Browser supporting full CSS with XML. (Mozilla, IE5.x)

V. Expanded Description

A. XML Basics

1. Origins of XML
2. Examples of XML Applications
3. Rules for well-formed documents
4. XHTML vs. HTML

B. Validating XML

1. Well-formed vs. valid documents
2. Validation methods
 - a. Document Type Definitions
 - b. XML Schema
 - c. Relax NG
3. Document Type Definitions
 - a. Elements
 - b. Attributes
 - c. General entities
 - d. Parameter entities
 - e. Conditional sections
4. Relax NG
 - a. Elements
 - b. Attributes
 - c. Data Types
 - d. Context dependencies

- C. Namespaces
 - 1. Definition
 - 2. Use with DTDs
- D. Cascading Style Sheets
 - 1. Properties and values
 - a. Font properties
 - b. Text properties
 - c. Borders, margins, and padding
 - d. Positioning
 - 2. Applying CSS to XML
- E. Document Structure
 - 1. Tree Structure
 - 2. Document Object Model
 - 3. XPath
 - a. Root node
 - b. Context node
 - c. Node axes
 - d. Full and abbreviated notation
 - e. XPath expressions
 - f. XPath functions
- F. XSL Transformations
 - 1. Repurposing documents
 - 2. The XSLT processing cycle
 - 3. Templates
 - 4. Extracting values
 - 5. Creating elements and attributes
 - 6. Numbering
 - 7. Variables
 - 8. Transforming XML to XHTML
 - 9. Basic XSL Formatting Objects (XSL-FO)
 - a. Page layout
 - b. Headers and footers
 - c. Block areas
 - d. Inline areas
 - e. Importing images
 - 10. Transforming XHTML to XSL-FO
- G. Parsing XML
 - 1. Document Object Model parsers
 - a. Functions for accessing nodes
 - b. Functions for creating nodes
 - c. Functions for rearranging tree structure
 - 2. Simple API for XML parsers
 - a. Event listener model
 - b. Document callbacks
 - c. Element callbacks
 - d. Text callbacks

VI. Assignments

- A. Required reading from text and handouts
- B. Six to ten assignments of XML activities

VII. Method of Evaluating Objectives

- A. A minimum of one one-hour midterm
- B. A comprehensive final exam or term project

VIII. Texts and Supporting References

A. Text:

Ray, Erik. Learning XML. Sebastopol, CA: O'Reilly & Associates, 2001.

B. References:

Harold, Elliotte Rusty, and W. Scott Means. XML in a Nutshell Sebastopol, CA: O'Reilly & Associates, 2001.