1. **COURSE ID:** CIS 379   **TITLE:** Introduction to XML and JSON
   Units: 3.0 units **Hours/Semester:** 48.0-54.0 Lecture hours; and 96.0-108.0 Homework hours
   Method of Grading: Grade Option (Letter Grade or Pass/No Pass)
   Recommended Preparation:
       Eligibility for ENGL 838 or ENGL 848 or ESL 400.

2. **COURSE DESIGNATION:**
   Degree Credit
   Transfer credit: CSU; UC

3. **COURSE DESCRIPTIONS:**
   Catalog Description:
       Comprehensive course in XML (eXtensible Markup Language). Includes writing well-formed and valid
   XML, the use of DTDs (Document Type Definitions), XML Schema, RSS, CSS (Cascading Style Sheets),
   HTML and XSLT (eXtensible Style Sheet Language Transformation) for formatting; and advanced topics
   such as XPath and the Document Object Model (DOM). Introduction to JavaScript Object Notation
   (JSON), a language-independent, open-standard data format supported by all modern browsers. Includes
   JSON data types, name/value pairs, arrays, objects, and object serialization.

4. **STUDENT LEARNING OUTCOME(S) (SLO'S):**
   Upon successful completion of this course, a student will meet the following outcomes:
   1. Develop well-formed and valid XML documents for publishing on the Web.
   2. Explain Document Type Definitions (DTDs) and XML Schema (XSD), and illustrate how they are used to
      validate XML documents for data integrity using XML parsers.
   3. Employ Cascading Style Sheets (CSS) and the eXtensible Style Sheet Language (XSL, XSLT) to render
      XML or to transform XML documents into HTML files.
   4. Explain XPath and the Document Object Model and discuss their relationship to XML.
   5. Create a multi-file project using XML, DTD, RSS, CSS, XSD, XSL, and HTML file types.
   6. Use JSON data in a multi-file project.

5. **SPECIFIC INSTRUCTIONAL OBJECTIVES:**
   Upon successful completion of this course, a student will be able to:
   1. Develop well-formed and valid XML documents for publishing on the Web.
   2. Explain Document Type Definitions (DTDs) and XML Schema (XSD), and illustrate how they are used to
      validate XML documents for data integrity using XML parsers.
   3. Employ Cascading Style Sheets (CSS) and the eXtensible Style Sheet Language (XSL, XSLT) to render
      XML or to transform XML documents into HTML files.
   4. Explain XPath and the Document Object Model and discuss their relationship to XML.
   5. Create a multi-file project using XML, DTD, RSS, CSS, XSD, XSL, and HTML file types.
   6. Use JSON data in a multi-file project.

6. **COURSE CONTENT:**
   Lecture Content:
   1. Introduction
          A. History of World Wide Web
          B. World Wide Web Consortium (W3C)
          C. Structure
          D. Style
          E. XHTML
          F. Well-formed Documents
          G. Valid Documents

   2. Cascading Style Sheets (CSS)
          A. Inline Styles
          B. Linking External Style Sheets
          C. Cascading and Levels of CSS

e. Object
                    f. Null
             D. JSON Syntax
             E. Name/Value Pairs
             F. Object Serialization
             G. Data Portability

7. **REPRESENTATIVE METHODS OF INSTRUCTION:**
   Typical methods of instruction may include:
   A. Lecture
   B. Directed Study
   C. Activity
   D. Discussion
   E. Other (Specify): Lecture will be used to introduce new topics; Teacher will model problem-solving
      techniques; Class will solve a problem together, each person contributing a potential "next step"; Students
      will participate in short in-class projects (in teacher-organized small groups) to ensure that students
      experiment with the new topics in realistic problem settings; Teacher will invite questions AND
      ANSWERS from students, generating discussion about areas of misunderstanding; Teacher will create and
      manage an Internet conference for discussion of course topics; and students will work in small groups to
      solve programming assignments.

8. **REPRESENTATIVE ASSIGNMENTS**
   Representative assignments in this course may include, but are not limited to the following:
   Writing Assignments:
       Weekly assignments: textbook exercises with XML and JSON projects
   Reading Assignments:
       Weekly reading assignments from the textbook and other sources

9. **REPRESENTATIVE METHODS OF EVALUATION**
   Representative methods of evaluation may include:
   A. Class Participation
   B. Class Work
   C. Exams/Tests
   D. Group Projects
   E. Homework
   F. Projects
   G. Quizzes
   H. Written examination
   I. Bi-weekly quizzes (short answer--from textbook material) to provide feedback to students and teacher;
      Assessment of student contributions during class discussion and project time; Individual programming
      assignment; Midterm and Final exams (short answer (textbook material), general problem solving (similar
      to in-class work), short program segments (similar to programming assignments)); Assessment of group
      participation on course projects, including peer-assessment of participation and contribution to the group
      effort.

10. **REPRESENTATIVE TEXT(S):**
    Possible textbooks include:
    A. Shirrell. *XML: A Deeper Understanding*, online ed. http://www.xmlbook.info/, 2018
    B. Lazaris. *What is JSON? An Introduction and Guide for Beginners*, online ed.
       https://www.impressivewebs.com/what-is-json-introduction-guide-for-beginners/, 2017
    C. Walmsley. *XQuery: Search Across a Variety of XML Data*, 2nd ed. O'Reilly, 2016
    D. Marrs. *JSON at Work: Practical Data Integration for the Web*, 1st ed. O'Reilly, 2017

<div align="right">

Origination Date: November 2018
Curriculum Committee Approval Date: December 2018
Effective Term: Fall 2019
Course Originator: Melissa Green

</div>