College of San Mateo
Official Course Outline

1. **COURSE ID:** CIS 114 **TITLE:** JavaScript/Ajax Programming
   Units: 4.0 units **Hours/Semester:** 48.0-54.0 Lecture hours; 48.0-54.0 Lab hours; and 96.0-108.0 Homework hours
   Method of Grading: Grade Option (Letter Grade or Pass/No Pass)
   Recommended Preparation:
   Completion of CIS 111 or CIS 254.

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

3. **COURSE DESCRIPTIONS:**
   Catalog Description:
   Study of the JavaScript programming language. Provides an overview of HTML5 and CSS, client-side programming, variables, arrays, functions, closures, event handlers, objects, form validation, cookies, and the DOM. Introduces Ajax (Asynchronous JavaScript and XML) technologies, design patterns, server-side programming, RSS, JSON, open-source libraries, and advanced topics such as ECMAScript versions, security, performance, and Web services. Intended for students with previous programming experience.

4. **STUDENT LEARNING OUTCOME(S) (SLO'S):**
   Upon successful completion of this course, a student will meet the following outcomes:
   1. Develop interactive Web applications that integrate HTML5 with JavaScript using event handlers.
   2. Explain object-based programming and the Document Object Model (DOM).
   3. Develop interactive Web applications that integrate client- and server-side programming using JavaScript and a server-side language.
   4. Employ Ajax technologies to fetch XML, RSS, or JSON data asynchronously from the server.
   5. Explain JavaScript design patterns and illustrate how they are used to create various applications.
   6. Create an advanced project using various libraries, with attention to security and performance.

5. **SPECIFIC INSTRUCTIONAL OBJECTIVES:**
   Upon successful completion of this course, a student will be able to:
   1. Develop interactive Web applications that integrate HTML5 with JavaScript using event handlers.
   2. Explain object-based programming and the Document Object Model (DOM).
   3. Develop interactive Web applications that integrate client- and server-side programming using JavaScript and a server-side language.
   4. Employ Ajax technologies to fetch XML, RSS, or JSON data asynchronously from the server.
   5. Explain JavaScript design patterns and illustrate how they are used to create various Ajax applications.
   6. Create an advanced project using various libraries, with attention to security and performance.

6. **COURSE CONTENT:**
   Lecture Content:
   1. Introduction
      A. History of JavaScript
      B. HTML5
      C. CSS
      D. Editors/IDEs
      E. Debuggers
      F. ECMAScript Versions
   2. JavaScript Basics
      A. Data and Operations
      B. Conditional Statements and Control Structures
      C. Functions
         a. Methods
         b. Closures
         c. Recursion
         d. Scope

        e. Event Handlers
    D. Arrays
        a. Array Object
        b. Array Methods
        c. Associative Arrays
        d. Multidimensional Arrays

3. Objects
    A. Native Object Types
    B. Prototype Inheritance
    C. Custom Object Types
    D. Browser Objects
    E. Document Object Model (DOM)
        a. DOM Scripting
        b. DOM Nodes
        c. HTML Nodes
4. Input/Output
    A. Validating Form Input
        a. Regular Expressions
        b. Exception Handling
    B. Reading/Writing Cookies
    C. Browser Detection
    D. Keystroke Detection
5. Introduction to Ajax
    A. Ajax Technologies
        a. XMLHttpRequest
    B. History of Ajax
    C. Real-World Examples
6. Ajax Design Patterns
    A. Observer Pattern
    B. Singleton Pattern
    C. Model-View-Controller
7. Ajax and the Server
    A. Ajax Libraries
    B. Server-Side Designs
    C. Exchanging Data
    D. Writing to the Server
        a. Using HTML Forms
        b. Using XHR Objects
        c. Managing User Updates
8. Integrating the Client and Server
    A. Client-Side Code
    B. Server-Side Code
    C. Browser Issues
9. Building Stand-Alone Ajax Applications
    A. Rich User Interface
    B. Loading XML and RSS Feeds
    C. Web Services
    D. Open-Source Libraries
    E. JSON
10. Ajax Security and Performance
    A. JavaScript and Browser Security
    B. Communicating with Remote Services
    C. Protecting Confidential Data
    D. Restricting Access to Web Data

Lab Content:

1. Programming project using a loop, a switch statement, and event handlers.

2. Programming project employing functions and arrays for data storage.

3. Programming project with object creation and manipulation.

4. Programming project with form submission, string and image processing.

5. Task management Web application with form submission.

6. Programming project with popup windows and cookies.

7. Programming project with select menus, checkboxes and/or radio buttons.

8. Programming project using regular expressions for input validation.

9. Programming project to fetch JSON data from a Web service.

10. Programming project using Ajax for server-side XML data retrieval.

11. Programming project using exception handling for error management.

12. Programming project using Ajax libraries such as jQuery.

13. Programming project using prototypes and prototypical inheritance.

14. Programming project using a public Web service (such as movies, iTunes or weather reports).

15. Programming project using Web APIs such as Google Maps, social media, language translators, or similar Web services.

16. Final project using Ajax libraries, frameworks, Web services, with attention to security and performance.

7. **REPRESENTATIVE METHODS OF INSTRUCTION:**
Typical methods of instruction may include:
   A. Lecture
   B. Lab
   C. Discussion
   D. Other (Specify): Teacher will model problem-solving techniques. Class will solve a problem together, each person contributing a potential "next step". Teacher will create and manage an Internet conference for discussion of course topics. 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 programming assignments Programming project using a loop, a switch statement, and event handlers. Programming project employing functions and arrays for data storage. Programming project with object creation and manipulation. Programming project with form submission, string and image processing. Task management Web application with form submission. Programming project with popup windows and cookies. Programming project with select menus, checkboxes and/or radio buttons. Programming project using regular expressions for input validation. Programming project to fetch JSON data from a Web Service. Programming project using Ajax for server-side XML data retrieval. Programming project using exception handling for error management. Programming project using Ajax libraries such as jQuery. Programming project using prototypes and prototypical inheritance. Programming project using a public Web service (such as movies, iTunes or weather reports). Programming project using Web APIs such as Google Maps, language translators, or similar Web services. Final project using Ajax libraries, frameworks, Web services, with attention to security and performance.
Reading Assignments:
Reading assignments accompanied by self-test questions and coding samples.

9. **REPRESENTATIVE METHODS OF EVALUATION**
Representative methods of evaluation may include:
   A. Class Work
   B. Exams/Tests
   C. Group Projects
   D. Homework

E. Lab Activities
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. Haverbeke. *Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming*, 3rd ed. No Starch Press, 2018
B. Ruvalcaba & Delamater. *Murach's JavaScript and jQuery*, 4th ed. Mike Murach & Associates, 2020
C. Flanagan, D. *JavaScript The Definitive Guide*, 7th ed. O'Reilly, 2020
D. Horstmann, C. *Modern JavaScript for the Impatient*, 1st ed. Pearson, 2020

Origination Date: October 2020
Curriculum Committee Approval Date: November 2020
Effective Term: Fall 2021
Course Originator: Melissa Green