C S 22A: JAVASCRIPT FOR PROGRAMMERS

Foothill College Course Outline of Record

Heading	Value
Effective Term:	Summer 2023
Units:	4.5
Hours:	4 lecture, 2 laboratory per week (72 total per quarter)
Advisory:	One of the following: C S 1A, 2A, 3A, or equivalent; knowledge of HTML and CSS.
Degree & Credit Status:	Degree-Applicable Credit Course
Foothill GE:	Non-GE
Transferable:	CSU/UC
Grade Type:	Letter Grade (Request for Pass/No Pass)
Repeatability:	Not Repeatable

Student Learning Outcomes

- Use a web application development environment that includes a browser, editor, debugger and code libraries.
- Write modifiable JavaScript programs that modify the DOM, respond to user events and make requests to the server.

Description

Introduction to object oriented programming in JavaScript. Topics include: client and server side programming, Model/View/Controller architecture, current tools and testing methods, interaction with HTML and CSS, Document Object Model, XML, and JSON. Students will have practice writing programs for mobile web browsers and creating dynamic webpages including animation.

Course Objectives

The student will be able to:

1. Install and use a web application development environment
2. Produce clearly written code in an industry standard style appropriate for JavaScript
3. Understand the basic syntax of the JavaScript language
4. Use popular JavaScript libraries and frameworks
5. Write a JavaScript program that responds appropriately to user events
6. Write client-side JavaScript code to modify the Document Object Model
7. Write client-side JavaScript code that makes requests to and handles replies from the server
8. Write server-side JavaScript code for authentication, optimization, and data storage
9. Understand web application architecture
10. Address performance in web applications
11. Use JSON and/or XML to access data for a web application
12. Relate algorithms and computation to real world problems and what software can do (such as graphics, virtual body medical examinations); solving problems in other fields (such as data science, health, etc.); or pro-equity impacts (such as toxicity and bias detection)

Course Content

1. Web application development environment
   a. Current browser with web development tools
   b. Selenium for testing
   c. Webkit
   d. Mobile browsers
   e. Test driven development
2. Coding style
   a. Documentation
   b. Indentation
   c. Object orientation
   d. Names for identifiers
   e. Tradeoffs between thin/fat client
   f. Tradeoffs between using html tags/JavaScript code
   g. Minimize dependencies between modules
   h. Model/view controller architecture
3. JavaScript syntax
   a. Variables and data types
   b. Expressions and operators
   c. Control structures
   d. Arrays and objects
   e. Functions including recursion
   f. Prototypes
   g. Closures
   h. Namespaces
   i. JSON
4. Libraries and frameworks
   a. JQuery
   b. Dojo
5. User events
   a. Mouse, keyboard, on page load, transitions
   b. Model/View/Controller architecture
   c. Interaction with HTML and CSS
   d. Asynchronous request to server
   e. Dynamic page refresh
6. Document Object Model
   a. HTML 5 or above and CSS 3 or above
   b. Traversing a document
   c. Finding elements in a document
   d. Modifying elements in a document
7. Communicating with the server
   a. Making requests to the server
   b. Handling responses from the server
   c. Asynchronous requests and responses
   d. Synchronous requests and responses
8. Server-side JavaScript
   a. Rendering an HTML page
   b. Caching data for optimization
   c. Security

   d. User authentication
   e. Data storage
9. Web application architecture
   a. Tomcat vs. node.js
   b. http and https
   c. Middleware
   d. Servlets
   e. LAMP
   f. Databases
10. Performance issues
   a. Throughput vs. latency
   b. Concurrency
   c. UI issues: bookmarks and back button
   d. Degradable user interface
   e. Issues with mobile platforms: phones and tablets
   f. Memoization
11. XML and JSON
   a. Comparing XML and JSON
   b. Storage of metadata
   c. Storage of user data

Lab Content

1. Install and use a web development environment to create and debug a web application
   a. Use iterative development
   b. Use test driven development
   c. Use good programming style
2. Incorporate JQuery in a web application
3. Write a web application that creates and uses its own namespace
4. Write a recursive JavaScript function
5. Write a resusable class in JavaScript
6. Write a program with a dynamic user interface
7. Make an asynchronous request from the server
8. Write a server side JavaScript program that creates an html page
9. Write an application that has registered users
10. Write an application that stores data:
   a. On the server
   b. On the client
11. Write an application that includes an animation and a transition

Special Facilities and/or Equipment

1. Access to a computer laboratory with internet access and modern browsers.
2. Web server that will host the student work.
3. Website or course management system with an assignment posting component (through which all lab assignments are to be submitted) and a forum component (where students can discuss course material and receive help from the instructor). This applies to all sections, including on-campus (i.e., face-to-face) offerings.
4. When taught via Foothill Global Access on the internet, a fully functional and maintained course management system through which the instructor and students can interact.
5. When taught via Foothill Global Access on the internet, students must have currently existing email accounts and ongoing access to computers with internet capabilities.

Method(s) of Evaluation

Methods of Evaluation may include but are not limited to the following:

Tests and quizzes
Written laboratory assignments which include source code, sample runs, and documentation
Final examination

Method(s) of Instruction

Methods of Instruction may include but are not limited to the following:

Lectures which include motivation for syntax and use of the JavaScript language and OOP concepts, example programs, and analysis of these programs
Online labs (for all sections, including those meeting face-to-face/on campus), consisting of:
1. A programming assignment webpage located on a college-hosted course management system or other department-approved internet environment. Here, the students will review the specification of each programming assignment and submit their completed lab work
2. A discussion webpage located on a college-hosted course management system or other department-approved internet environment. Here, students can request assistance from the instructor and interact publicly with other class members
Detailed review of programming assignments which includes model solutions or quality assurance specifications and specific comments on the student submissions
In-person or online discussion which engages students and instructor in an ongoing dialog pertaining to all aspects of designing, implementing, and analyzing programs
When course is taught fully online:
1. Instructor-authored lecture materials, handouts, syllabus, assignments, tests, and other relevant course material will be delivered through a college-hosted course management system or other department-approved internet environment
2. Additional instructional guidelines for this course are listed in the attached addendum of CS department online practices
Collaborative team assignments: graded work, based on course material, that requires active interaction between students (e.g., using pair programming)
Students do self reflections and/or self evaluations throughout the term
Supervised activity to enable students to showcase their unique accomplishments in the classroom and provide opportunities to showcase beyond the classroom
Students are provided with up-to-date information about instructional services available to them
Learning units or modules are consistently structured and sequenced to reduce cognitive load
Use of formative assignments and/or assessments for applying skills recently experienced in course content

Representative Text(s) and Other Materials

Haverbeke, Marijn. Eloquent JavaScript. 2019.

Boduch, Adam, Jonathan Chaffer, and Karl Swedberg. Learning jQuery 3#: Build Interesting, Interactive Sites Using jQuery by Automating Common Tasks and Simplifying the Complicated Ones. 2017.

The popular jQuery library hasn't substantially changed for years, and the Boduch/Chaffer/Swedberg text still has current content. jQuery is still numbered as version 3.

Both texts are available for free as eBooks via the Foothill College library or the authors' website and thus are OER (Open Educational Resources).

Types and/or Examples of Required Reading, Writing, and Outside of Class Assignments

1. Reading
    a. Textbook assigned reading averaging 30 pages per week
    b. Reading the supplied handouts and modules averaging 10 pages per week
    c. Reading online resources as directed by instructor though links pertinent to programming
    d. Reading library and reference material directed by instructor through course handouts
2. Writing
    a. Writing technical prose documentation that supports and describes the programs that are submitted for grades

Discipline(s)

Computer Science