SAN DIEGO COMMUNITY COLLEGE DISTRICT
CONTINUING EDUCATION
COURSE OUTLINE

SECTION I

SUBJECT AREA AND COURSE NUMBER

COMP 690

COURSE TITLE

WEB PROGRAMMING: JAVASCRIPT

TYPE COURSE

NON-FEE VOCATIONAL

CATALOG COURSE DESCRIPTION

This course incorporates JavaScript into the web development process and serves as a foundation for the Full-Stack Web Development certificate program. Students will learn about modern web architecture and how JavaScript serves as an armature for every component within that ecosystem. More importantly, students will learn how to program for the web using JavaScript. Students will also learn how JavaScript can enhance a webpage, allowing additional interactivity and more precise control of page elements. Techniques used in creating a website and making the content more dynamic will also be taught. (FT)

LECTURE/LABORATORY HOURS

180

ADVISORIES

Successful completion of COMM 641or equivalent prior knowledge or experience.

RECOMMENDED SKILL LEVEL

Possess a 12th grade reading level; ability to communicate effectively in the English language.

INSTITUTIONAL STUDENT LEARNING OUTCOMES

1. Social Responsibility
   SDCE students demonstrate interpersonal skills by learning and working cooperatively in a diverse environment.
2. Effective Communication
   SDCE students demonstrate effective communication skills.

3. Critical Thinking
   SDCE students critically process information, make decisions, and solve problems independently or cooperatively.
4. Personal and Professional Development
   SDCE students pursue short term and life-long learning goals, mastering necessary skills and using resource management and self -advocacy skills to cope with changing situations in their lives.

COURSE GOALS

1. Introduce JavaScript and understand the role of JavaScript in the modern web architecture model
2. Introduce the fundamentals of the JavaScript programming language
3. Introduce advanced concepts in JavaScript
4. Understand the purpose of Document Object Model (DOM) scripting
5. Learn how to retrieve and manipulate web page content
6. Understand how to work with events
7. Learn how to consume and display JavaScript Object Notation (JSON) data
8. Introduce debugging, error handling, and testing concepts in JavaScript
9. Learn about package managers, build tools, and task runners

COURSE OBJECTIVES

Upon successful completion, the student will be able to:

1. Describe the role of JavaScript and how the language ties into the modern web architecture model
2. Use fundamental programming concepts to build a simple program in the web browser
3. Use advanced programming concepts to build more complex programs in the web browser
4. Retrieve and manipulate web page elements, otherwise known as DOM scripting
5. Create events and event handlers to add interactivity to a web page
6. Consume and display the content contained within a JSON file directly within a web page
7. Properly debug and gracefully handle errors within a web page
8. Use testing tools to test the performance of web applications
9. Use package managers, build tools, and task runners to automate tedious JavaScript processes including downloading of third-party tools

SECTION II

COURSE CONTENT AND SCOPE

1. Introduction to Web Programming with JavaScript
    - 1.1. Classic Web architecture
    - 1.2. The Core 3: HTML5, CSS3, and JavaScript
    - 1.3. Modern Web architecture
    - 1.4. Future Web architecture
    - 1.5. ECMAScript and current JavaScript support
    - 1.6. The W3C
    - 1.7. The Mozilla foundation
    - 1.8. Basic structure of a Web page
    - 1.9. The Document Object Model (DOM)
    - 1.10. Including JavaScript in a Web page
    - 1.11. Using a browser for testing code
2. JavaScript Essentials
    - 2.1. JavaScript syntax
        - 2.1.1. Expressions
        - 2.1.2. Punctuation
        - 2.1.3. Case sensitivity
        - 2.1.4. Camel case
        - 2.1.5. Hungarian notation
        - 2.1.6. Comments
        - 2.1.7. Keywords
        - 2.1.8. Reserved keywords
    - 2.2. Variables
        - 2.2.1. Declaration
            - 2.2.1.1. var
            - 2.2.1.2. let
            - 2.2.1.3. const
        - 2.2.2. Assignment
        - 2.2.3. Dynamic vs. strong typing
        - 2.2.4. Coercion
    - 2.3. Types
        - 2.3.1. string
        - 2.3.2. String escape sequences
        - 2.3.3. number
        - 2.3.4. Boolean
        - 2.3.5. null and undefined
        - 2.3.6. object
        - 2.3.7. The typeof operator
        - 2.3.8. Data type conversion
    - 2.4. Operators
3. Program Structure
    - 3.1. Expressions and statements
    - 3.2. Helpful functions
    - 3.3. Control flow
    - 3.4. Conditional statements

- 3.4.1. if else
- 3.4.2. Ternary operator
- 3.4.3. switch

3.5. Looping Statements

- 3.5.1. for
- 3.5.2. for in
- 3.5.3. while
- 3.5.4. do
- 3.5.5. break and continue

4. Functions
   - 4.1. Global functions
   - 4.2. Defining a function
   - 4.3. Functions as values
   - 4.4. Function Declarations
     - 4.4.1. Creating and calling a function
     - 4.4.2. Passing parameters to functions
     - 4.4.3. By value vs by reference
     - 4.4.4. The arguments object
     - 4.4.5. Returning values
     - 4.4.6. Scope and hoisting
     - 4.4.7. Strict mode
   - 4.5. Function Expressions
     - 4.5.1. Creating an anonymous function
     - 4.5.2. Using anonymous functions to return a DOM element
   - 4.6. Advanced Functions
     - 4.6.1. Arrow functions
     - 4.6.2. Optional arguments
     - 4.6.3. Closures
     - 4.6.4. Recursion
     - 4.6.5. Growing functions
5. Data Structures: Objects and Arrays
   - 5.1. Data sets
   - 5.2. Properties
   - 5.3. Methods
   - 5.4. Objects
   - 5.5. Mutability
   - 5.6. Introduction to Arrays
     - 5.6.1. Populating Arrays
     - 5.6.2. Referencing Array elements
     - 5.6.3. Adding elements at specific positions
     - 5.6.4. Finding the length of an Array
     - 5.6.5. Deleting Array elements
     - 5.6.6. Iterating over an Array
     - 5.6.7. Members of the Array object
   - 5.7. Two-Dimensional arrays
   - 5.8. The Web storage Application Programming Interface (API)
6. Strings, Numbers, Math, and Dates
   - 6.1. Working with Strings
   - 6.2. Working with Numbers

- 10.4. Evaluating data as code
- 10.5. CommonJS
- 10.6. ECMAScript modules
- 10.7. Building and bundling
- 10.8. Module design

11. Asynchronous Programming
    - 11.1. Asynchronicity
    - 11.2. Crow tech
    - 11.3. Callbacks
    - 11.4. Promises
    - 11.5. Failure
    - 11.6. Dealing with difficult networks
    - 11.7. Collections of Promises
    - 11.8. Network flooding
    - 11.9. Message routing
    - 11.10. Async functions
    - 11.11. Generators
    - 11.12. The event loop
    - 11.13. Common bugs
    - 11.14. The XMLHttpRequest Object
    - 11.15. Using the .fetch() method
        - 11.15.1. Using the .fetch() method with Promises
        - 11.15.2. Using the .fetch() method with Async/Await
12. JavaScript and the Browser
    - 12.1. The window Object
    - 12.2. window.navigator
    - 12.3. window.location
    - 12.4. window.history
    - 12.5. window.frames
    - 12.6. window.screen
    - 12.7. window.open()
    - 12.8. window.close()
    - 12.9. window.moveTo()
    - 12.10. window.resizeTo()
    - 12.11. window.setTimeout()
    - 12.12. window.setInterval()
    - 12.13. window.document
13. DOM Programming
    - 13.1. Document Structure
    - 13.2. Trees
    - 13.3. The Standard
    - 13.4. Moving through time
    - 13.5. Finding elements
    - 13.6. Changing the document
    - 13.7. Creating nodes
    - 13.8. Attributes
    - 13.9. Layout
    - 13.10. Styling
    - 13.11. Cascading styles

13.12. Query selectors
13.13. Positioning and animating
14. Handling Events
14.1. Registering event listeners
14.2. Common HTML DOM events
14.3. The event object
14.4. Event propagation / event delegation
14.5. Stopping propagation
14.6. Preventing the browser's default behavior

APPROPRIATE READINGS

Reading assignments may include, but are not limited to: assignments from the textbook, supplemental reading assignments, industry-related periodicals or magazines, manuals, online help pages, articles posted on the Internet, and information from Web sites, online libraries and databases. Topics should be related to web programming techniques with JavaScript and include techniques for manipulating elements on a web page using JavaScript.

WRITING ASSIGNMENTS

Writing assignments may include, but are not limited to: completing assigned reports, providing written answers to assigned questions, performing internet research and reporting on that research. An example would include a case study of JavaScript data structures can be used in conjunction with web storage to persist a shopping cart within a web application.

OUTSIDE ASSIGNMENTS

Assignments may include, but are not limited to: appropriate internet research, reading, preparing reports and studying as needed to perform successfully in class. An appropriate assignment for instance, would include the creation of a mortgage calculator web application written in JavaScript.

APPROPRIATE ASSIGNMENTS THAT DEMONSTRATE CRITICAL THINKING

Assignments which demonstrate critical thinking may include, but are not limited to: building a CRUD (create, retrieve, update, and delete) web application with JavaScript using data structures. Students may also be expected to participate in online class discussion posts, in-class discussions and project reviews.

EVALUATION

Evaluation that a student has met the course competencies will include multiple measures of performance related to the course objectives. Evaluation methods may include, but are not limited to performance in a variety of activities and assignments, such as completing a research project individually or in a group, hands-on projects, demonstration of use of the internet, quizzes, class participation, written and practical tests, attendance and punctuality.

Upon successful completion of all courses in the program a Certificate of Program Completion will be issued.

METHOD OF INSTRUCTION

Methods of instruction, may include, but are not limited to, lecture, in-class and online discussions, hands-on demonstrations, computer-assisted instruction, field trips, and laboratory assignments.

This course, or sections of this course, may be offered through distance education.

TEXTS AND SUPPLIES

OER Textbooks

Eloquent JavaScript: A Modern Introduction to Programming,
Marijn Haverbeke, No Starch Press, current edition

Web Resources:
Udemy: Programming with JavaScript, https://www.udemy.com/course/programming-with-javascript/

Supplies:

Pen, journal (composition book), notebook paper and a soft 3-ring binder, or a one-subject 110 sheet college ruled notebook, and appropriate storage media such as a USB Drive, external hard drive, or cloud-based storage.

PREPARED BY Zak Ruvalcaba DATE April 7, 2021

REVISED BY _____ DATE _____

Instructors must meet all requirements stated in Policy 3100 (Student Rights, Responsibilities and Administrative Due Process), and the Attendance Policy set forth in the Continuing Education Catalog.

REFERENCES:

San Diego Community College District Policy 3100
California Community Colleges, Title 5, Section 55002
Continuing Education Catalog