

College of San Mateo
Official Course Outline

1. **COURSE ID:** CIS 111 **TITLE:** Introduction to Web Programming

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)

2. **COURSE DESIGNATION:**

Degree Credit

Transfer credit: CSU; UC

3. **COURSE DESCRIPTIONS:**

Catalog Description:

This course provides an introduction to the World Wide Web and Web programming. Basic HTML and CSS will be introduced, as well as client-side scripting in JavaScript using variables, functions, and objects. Introduces programming with HTML, CSS, JavaScript, and the DOM (Document Object Model). Server-side scripting languages such as Python and PHP will be demonstrated. Introduces Web 2.0 topics, such as Ajax, Web Services, and HTML5.

4. **STUDENT LEARNING OUTCOME(S) (SLO'S):**

Upon successful completion of this course, a student will meet the following outcomes:

1. Explain basic internet concepts and technologies.
2. Create HTML and HTML5 documents.
3. Create Cascading Style Sheets (CSS) to format HTML and HTML5 documents.
4. Write simple client-side JavaScript programs employing variables, conditional statements, and control structures.
5. Develop HTML and HTML5 Web applications employing the Document Object Model (DOM), CSS, and JavaScript.
6. Explain server-side scripting concepts and languages.
7. Create a Web 2.0 application employing Ajax.

5. **SPECIFIC INSTRUCTIONAL OBJECTIVES:**

Upon successful completion of this course, a student will be able to:

1. Explain basic internet concepts and technologies.
2. Create HTML and HTML5 documents.
3. Create Cascading Style Sheets (CSS) to format HTML and HTML5 documents.
4. Write simple client-side JavaScript programs employing variables, conditional statements, and control structures.
5. Develop HTML and HTML5 Web applications employing the Document Object Model (DOM), CSS, and JavaScript.
6. Explain server-side scripting concepts and languages.
7. Create a Web 2.0 application employing Ajax.

6. **COURSE CONTENT:**

Lecture Content:

1. Introduction to internet and WWW concepts and protocols.
2. HTML and HTML5 elements.
3. Cascading Style Sheets (CSS) (external and internal).
4. JavaScript data types, variables, constants, and operators.
5. JavaScript conditional statements and control structures.
6. JavaScript functions, arrays, and objects.
7. JavaScript events and event handlers.
8. Introduction to the Document Object Model (DOM).
9. Introduction to server-side scripting languages.
10. Web 2.0 concepts, including Ajax libraries and frameworks, and Web Services.

Lab Content:

Students will complete lab exercises and write programs that reinforce the lecture material.

1. Utilize HTML and JavaScript to create client-side programs.
2. Employ variables, arrays and control structures in JavaScript programs.

3. Write JavaScript programs that respond to events.
4. Develop HTML and HTML5 Web applications employing the Document Object Model (DOM), CSS, and JavaScript.
5. Test server-side programs that use languages such as ASP, JSP, Python, Perl and PHP.
6. Create a Web-based Ajax program using a framework such as jQuery.

7. REPRESENTATIVE METHODS OF INSTRUCTION:

Typical methods of instruction may include:

- A. Lecture
- B. Lab
- C. Directed Study
- D. Discussion
- E. Observation and Demonstration
- F. Other (Specify): The course will include the following instructional methods as determined appropriate by the instructor, in approximately the following order: Lecture will be used to introduce new topics; Instructor will create and manage an internet forum for discussion of course topics; Instructor will model problem-solving techniques; Instructor will provide demonstrations of current internet programming techniques; Class will apply methodologies in class to solve a problem together; Students will participate in an ongoing large project in instructor-organized small groups; Instructor will invite questions AND ANSWERS from students, generating discussion about areas of misunderstanding; Students will give individual presentations of outside readings or final project area.

8. REPRESENTATIVE ASSIGNMENTS

Representative assignments in this course may include, but are not limited to the following:

Writing Assignments:

Students will complete and submit exercises and programming assignments on a weekly basis.

Assignments will be related to course content, including HTML, HTML5, CSS, JavaScript variables, control structures, methods/functions, arrays, DOM scripting, server-side scripting, Ajax and jQuery.

Reading Assignments:

Students are required to read assigned chapters in the textbook.

9. REPRESENTATIVE METHODS OF EVALUATION

Representative methods of evaluation may include:

- A. Class Participation
- B. Exams/Tests
- C. Group Projects
- D. Homework
- E. Lab Activities
- F. Projects
- G. Quizzes
- H. Written examination
- I. Weekly textbook readings and textbook exercises comprise one portion of the out-of-class assignments. Assessment of student contributions during discussion time. Hands on lab assignments utilize and reinforce programming techniques. Midterm and final exams provide additional assessment of learning outcomes. Online discussion boards can be assessed using rubrics, instructor comments, peer review, and/or other methods of evaluation.

10. REPRESENTATIVE TEXT(S):

Possible textbooks include:

- A. Carey, P. *New Perspectives on HTML 5 and CSS: Comprehensive*, 8th ed. Cengage, 2020
- B. Negrino, T. *Javascript: Visual Quickstart Guide*, 8th ed. Pearson, 2018
- C. Simpson, K. *You Don't Know JS Yet: Get Started*, 1st ed. Independently published, 2020

Origination Date: October 2020

Curriculum Committee Approval Date: November 2020

Effective Term: Fall 2021

Course Originator: Melissa Green