1. **COURSE ID:** CIS 127  
**TITLE:** HTML5 and CSS  
**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 P/NP)  
**Recommended Preparation:**  
Eligibility for ENGL 838 or ENGL 848  
Eligibility for ESL 400, Completion of CIS 111 or CIS 254.

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

3. **COURSE DESCRIPTIONS:**  
**Catalog Description:**  
Introduction to HTML5 and CSS (Cascading Style Sheets). Covers CSS3, HTML5 elements, HTML5 APIs, forms, audio and video, offline applications, Canvas drawing and animation, communication APIs, Web Sockets, and Web Workers. Introduces HTML5 Geolocation, local and session storage, the Web SQL Database, and advanced topics such as mobile web applications, performance analysis, browser issues, and developer tools. Intended for students with previous programming experience. This course cannot be substituted for DGME 169.

4. **STUDENT LEARNING OUTCOME(S) (SLO'S):**  
Upon successful completion of this course, a student will meet the following outcomes:  
1. Develop HTML5 Web and offline applications using CSS for layout.  
2. Use the HTML5 Canvas element for drawing and animation.  
3. Create HTML5 applications that employ audio and video.  
4. Develop Web applications for various mobile devices.  
5. Use HTML5 APIs for geolocation, communications, sockets, and threads.  
6. Employ HTML5 storage capabilities for local and session storage, and the Web SQL Database.  
7. Create an advanced project using the various HTML5 technologies, with attention to security and performance.

5. **SPECIFIC INSTRUCTIONAL OBJECTIVES:**  
Upon successful completion of this course, a student will be able to:  
1. Develop HTML5 Web and offline applications using CSS for layout.  
2. Use the HTML5 Canvas element for drawing and animation.  
3. Create HTML5 applications that employ audio and video.  
4. Develop Web applications for various mobile devices.  
5. Use HTML5 APIs for geolocation, communications, sockets, and threads.  
6. Employ HTML5 storage capabilities for local and session storage, and the Web SQL Database.  
7. Create an advanced project using the various HTML5 technologies, with attention to security and performance.

6. **COURSE CONTENT:**  
**Lecture Content:**  
1. HTML5 and CSS Overview  
   A. History of HTML5 and CSS (Cascading Style Sheets)  
   B. WHATWG and W3C specifications  
   C. Levels of CSS: CSS1, CSS2, CSS3  
2. Introduction to HTML5 Markup  
   A. HTML5 page structure  
   B. HTML5 DOCTYPE  
   C. HTML5 elements  
      i. Structural elements  
      ii. Semantic elements  
      iii. Deprecated elements
D. HTML5 and CSS
E. Using HTML5 in browsers that do not support it
   i. Detecting native availability of HTML5 features
   ii. Emulation

3. Introduction to CSS
   A. Selectors
   B. Layout
   C. Visual effects
   D. Transformations and animation
   E. Browser issues

4. Overview of HTML5 APIs
   A. Common building blocks
   B. Programming HTML5
      i. JavaScript
      ii. DOM scripting

5. HTML5 Forms
   A. HTML5 form elements
   B. Building and using HTML5 forms

6. HTML5 Offline Applications
   A. Offline manifest files
   B. ApplicationCache events

7. HTML5 Audio and Video
   A. Audio and video elements
   B. Containers and codecs
   C. Audio and video APIs

8. HTML5 Canvas
   A. Canvas coordinates
   B. Canvas APIs
   C. Drawing operations
   D. Canvas transforms
   E. Animation

9. HTML5 Communication APIs
   A. Cross-document messaging
   B. XMLHttpRequest Level 2
      i. Progress events
      ii. Cross-origin resource sharing
   C. Server-sent Events

10. HTML5 Web Sockets
    A. Introduction to sockets
    B. Web Sockets API
    C. Web Sockets Protocol

11. HTML5 Web Workers
    A. Introduction to threads
    B. Web Worker communication
    C. Web Workers APIs

12. HTML5 Geolocation
    A. Geolocation overview
       i. User Privacy
       ii. Location information sources
    B. Geolocation APIs

13. HTML5 Storage
    A. Local Storage
    B. Session Storage
    C. Web SQL Database

14. Advanced Topics
    A. HTML5 on mobile devices
    B. Performance Analysis
    C. Developer Tools
7. REPRESENTATIVE METHODS OF INSTRUCTION:
Typical methods of instruction may include:
   A. Lecture
   B. Directed Study
   C. Activity
   D. Discussion
   E. Observation and Demonstration
   F. Other (Specify): Lecture will be used to introduce new topics; Instructor will model problem-solving techniques; Class will solve a problem together, each person contributing a potential "next step"; Instructor will invite questions AND ANSWERS from students, generating discussion about areas of misunderstanding; Instructor will create and manage an Internet forum 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 programming assignments.

   Reading Assignments:
   Reading assignments accompanied by self-test questions and running code examples.

   Other Outside Assignments:
   Studying posted lecture notes and relevant handouts. Computer programming assignments. The reading assignment frames the concepts covered and provides the basic knowledge necessary to do the self-test questions and understand the sample code. The lecture notes and handouts provide a more in-depth look at topics and distills the information down to what the faculty thinks is most important. The programming assignments provide hands on practice of the concepts covered in the readings.

9. REPRESENTATIVE METHODS OF EVALUATION
Representative methods of evaluation may include:
   A. Exams/Tests
   B. Group Projects
   C. Homework
   D. Lab Activities
   E. Projects
   F. Quizzes
   G. Written examination
   H. 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 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:

   Origination Date: September 2016
   Curriculum Committee Approval Date: October 2016
   Effective Term: Fall 2017
   Course Originator: Melissa Green