UC University of California It starts here

Print Close

Add New / Reinstate Courses

View New Course

Teacher Contact

* **First Name:** Romeo

* **Last Name:** Simionas

* **Position/Title:** Teacher

Phone Number: (510)364-4503

* **E-mail:** rsimionas@yahoo.com

* **Was this course Previously Approved by UC?** No

* **Course Title:** Web Development

* **Transcript Title /Abbreviation:**	**Transcript Title /Abbreviation:**	**Course Code**
	Web	2525

* **Seeking "Honors" Distinction:** No

* **Subject Area:** Elective

* **Category:** Other

* **Grade Level for which this course has been designed:** □ 9 ☑ 10 ☑ 11 ☑ 12

* **Unit Value:** 1.0 (one year, 2 semesters, or 3 trimesters equiv.)

* **Is this course classified as a Career Technical Education:** Yes

Name of Industry Sector: Information Technology

Name of Career Pathway: Information Support and Services

* Brief Course Description

This course is designed to teach the fundamentals of developing Web pages. Students will learn how to create web pages using (X)HTML, CSS, JavaScript. The web pages will have to pass the World Wide Web Consortium (W3C) validation process as XHTML compliant. Also, students will be introduced to Dreamweaver as an alternative to coding.
This course will prepare students to choose a major UC or CSU program in Web Development/Design, or Computer Programming.

Pre-Requisites

Computer Skills Class - Required

Algebra 1 - Required

Geometry - Recommended

Co-Requisites

Graphics Design - Recommended

* Context for Course

Web Development is one of our classes for our Information Technology industry sector. Students start by taking Computer Skills and then take Web Development. In parallel, they can take other computer related courses such as Computer Graphics, Computer Repair, or Robotics. Pittsburg High School is looking to create an IT Academy where students would take a series of IT classes (including Web Development) to earn a certificate.

* History of Course Development

Andreas Kaiser started teaching Web Design 6-7 years ago as an afterschool class. The intent was to have the students create a web site for the school (we had none at that time!). Mr. Kaiser wanted to pursue CAD and Robotics more and Mike Ford took over the web project. He created a class during the day to teach the students the basics of HTML and Dreamweaver.
Currently the class is taught by Romeo Simionas, who wrote this curriculum.

Textbooks

TEXTBOOK 1

* **Title:**	HTML-Comprehensive Concepts and Techniques, Fifth Edition
* **Edition:**	Fifth

TEXTBOOK 1

* **Publication Date:**	2009
* **Publisher:**	Course Technology - Cengage Learning
* **Author (s):**	Gary B. Shelly, Denise M. Woods, William J. Dorin
URL Resource:	www.scsite.com
* **Usage:**	Primary Text Read in entirety or near entirety

TEXTBOOK 2

* **Title:**	Adobe Dreamweaver CS5 Revealed
* **Edition:**	1
* **Publication Date:**	2010
* **Publisher:**	Cengage Learning
* **Author (s):**	Sherry Bishop
URL Resource:	www.cengage.com
* **Usage:**	Supplementary or Secondary Text Read in entirety or near entirety

Supplemental Instructional Materials

HTML Referrence: http://www.w3schools.com/tags/default.asp

CSS Referrence: http://www.w3schools.com/css/css_reference.asp

JavaScript Objects Referrence: http://www.w3schools.com/jsref/default.asp

* Course Purpose

Objectives:

1. Students will learn the Web Development Phases.

2. Students will learn and use the HTML language to create web pages suitable for class work, personal or professional/business purposes.

3. Students will learn how to code and add Cascading Style Sheets (CSS) to the web pages for a more consistent format and look.

4. Students will test each page by uploading them to the W3C validator (http://validator.w3.org/).

5. Students will learn basics about event driven, object-based programming languages in order to be able to write/add JavaScript to the web pages.

6. Students will learn how to add dynamic content to a web page by integrating JavaScript and HTML.

7. Students will be introduced to Dreamweaver as an alternative to coding web pages and will create a small web site using Dreamweaver.

Learning Outcomes:

1. Students use critical thinking and problem-solving skills when completing projects.

- Use evidence and analysis to draw conclusions when making decisions on developing original web pages.
- Students explore World Wide Web resources independently and make decisions on the information they use to create original web pages.
- Use feedback from teacher and peers to improve work.

2. Students show creativity in completing original projects.

- The projects assigned, like the business web site, or personal web site, motivate students to be creative and passionate about the work they do.
- Students design high quality, original work and are connected to the subject assigned.

3. Students use strong communication skills to complete high quality projects.

- Use proper written language on web pages to communicate the intended message.
- Communicate with teacher and peers when help or assistance is needed.
- Communicate with other people to obtain feedback for the web pages developed.

* Course Outline

A. Introduction to Web Development.

(1-3: students will research and write a paper. The W3C site is recommended for research: http://www.w3.org/. The definitions will be discussed and clarified in class with the teacher).

1. What is the World Wide Web? Short history

2. What is the Internet? Short history.

3. Find the meanings and define the following acronyms: HTML, XHTML, HTTP, ISP, CSS, XML. (list open for other meaningful terms/acronyms)

4. Web Development Life Cycle: Planning, Analysis, Design and Development, Testing, Implementation and Maintenance.
   The Web Development Life Cycle and its elements will be presented in detail by teacher in class.

B. Creating and Editing Web Pages with HTML and CSS.

1. The Basics.

a) Using Notepad as a basic text editor for creating web pages using HTML.
b) Other text editors can be used, including third party more specialized editors.
Students are encouraged to do research on other editors. Notepad will be used in this class, since it’s the most largely available.
c) Saving the text file with the .html extension creates a web page.
d) The home page.
e) Basic elements of a web page: title, body, background, headings, text, images, hyperlinks.

f) Web page structure: a basic template for the transitional document type is introduced to students. Each tag is explained in detail. Students will use this template as a start for each of their web pages.
g) Entering web page content, viewing and saving:
- text: use <p> </p> tags to add a paragraph of text.
- introducing tag syntax: <tag attribute=”value”>, ex.: <ul type=”disc”>
- headings <h1> …<h6> with the align attribute and values.
- lists: ordered, unordered, types. Tags used:<ul></ul>, <ol></ol>, <li></li> and attributes, values.
- For the tags learned, students will refer to http://www.w3schools.com/tags/default.asp for a full list of attributes and values.
- Saving and organizing HTML files. Students will create a folder structure with separate folders for each assignment/project and an “Images” subfolder for each web site they create.
- Using the browser to view the web page: open the page with a browser for viewing and with the text editor for editing.
h) Adding images, background, horizontal rules.
For the tags learned, students will refer to http://www.w3schools.com/tags/default.asp for a full list of attributes and values.
- Image types: GIF, JPEG, PNG
- Image tag <img> and attributes: align, alt, border, height, hspace, src, vspace, width.
- Adding an image to the page: minimum of attributes required for W3C validation:
<img src=”image.ext” width=”” height=”” alt=”” />
- Adding background color: use attribute bgcolor to the <body> tag. Values for the attribute are colors in hexadecimal codes, or words for predefined colors. An application displaying the hex codes for colors is available for students to run from the Instructor folder.
- Adding horizontal rules using tag <hr /> with attribute size. Ex.: <hr / size=”5”>
f) Validating and viewing HTML code.
- World Wide Web Consortium (W3C) defines standards for HTML.
- Importance of code validation.
- Validation service used for this course: http://validator.w3.org/
- Viewing HTML code from within the browser.

2. Adding Links, Formatting Images and Text.

a) Adding Links.
- Before adding links, students will refer to the Web Development Life Cycle studied in the beginning of this course. In planning the web site, students will determine where the links are necessary. Planning ahead is very important and helps identifying how to format various elements of the text and graphics. Here are a few hints:
a) Effectively use graphics: ex: use same logo on all pages.
b) Use headings that connect to the graphics.
c) Use default text size when appropriate.
d) Highlight key text with ordered or unordered lists.
- Before creating links, students are introduced to the absolute/relative file path theory

- Link tags:<a></a> with attributes: href, name, rel, rev, type.
- Basic link syntax: <a href="URL">linktext</a>
- Link color attributes: link, vlink, alink. Used with the <body> tag
- Text link to another web page within the same web site: <a href="filename.html">linktext</a>
- Email link: <a href=mailto:email address>linktext</a>
- Links within a web page: <a name="targetname"></a> <a href="#targetname"></a>
- Creating a navigation bar.
- Testing links.

b) Formatting Text
- <font> tag (deprecated – CSS used mostly) can be used with different attributes to format text: color, size, face.
- Other HTML tags used for text formatting: <b></b>, <big></big>, <blockquote></blockquote>, <em></em>, <i></i>, <strong></strong>, <sub></sub>, <sup></sup>, <tt></tt>, <u></u>. Some may be deprecated.

For the tags learned, students will refer to http://www.w3schools.com/tags/default.asp for a full list of attributes and values.

c) Formatting Images
- Alignment is key consideration when inserting an image: using the align attribute with "left" and "right" values will wrap the text to left or right of the image.
- Hspace and vspace attributes control the amount of horizontal and vertical space around the image in pixels.
- The border attribute can be used to add a border around the image. Value is a number representing pixels.
- Thumbnail images: used to improve page loading time.

For the tags learned, students will refer to http://www.w3schools.com/tags/default.asp for a full list of attributes and values.

3. Creating Style Sheets (CSS)

a) About CSS.
- CSS is not HTML, it is a separate language used to define the style for a web page or entire web site.
- CSS was created because HTML is limited in defining appearance, style.
- W3C defines standards for CSS as well.
- CSS is powerful: can establish a standard look for all pages at the same time, avoiding adding repetitive code to individual pages.

b) Types of style sheets.
- Inline style sheets
- Embedded style sheets
- External style sheets
- Style sheet precedence: Inline, Embedded, External

c) Planning Style Sheets.

- When planning, precedence rules must be taken in account first.
- Use external style sheets for styles that apply across the web site.
- Use embedded style sheets for single web page styles.
- Use inline style sheets for individual styles.

d) CSS Syntax

- A CSS rule has two parts: a selector and one or more declarations.
- The selector is the HTML element we want to style.
- Each declaration consists of a property and a value.
- The property is the style attribute we want to change. Each property has a value.
- Refer to http://www.w3schools.com/css/css_reference.asp for CSS properties code.

e) Inline Style Sheets.

- Used to define the style of an individual HTML tag.
- Code: added with the HTML tag.

f) Embedded Style Sheets.

- Used to control the style of a single web page.
- Code: insert the embedded style sheet inside <style></style> tags at the top of the web
page within the <head></head> tags.

g) External Style Sheets.

- Used to control the look of many web pages within a web site.
- Code: two steps:

1. Create an external style sheet with all the style statements and save with a .css extension.

2. For each web page to which we apply the styles, a <link /> tag will be inserted within the <head></head> tags. Attributes for the <link /> tag: rel, type, href.

h) Classes in CSS.

- Used to gain more control over the style on a web page.
- Different variations for any one tag can be created.
- Code:

1. The class is defined with a name used as identifier: class=”classname”

2. The name is used in a selector to define a specific style for the class.

4. Creating tables

a) Table elements. See pages 150-151 of text book for graphic description.
- Row, column, cell
- Headers, border, caption, rules

b) Planning the table.
- Before creating tables, students will refer to the Web Development Life Cycle studied in the beginning of the course.
- Determining if a table is needed.
- Determine how the information will appear in the table.
- Sketch the table on paper.

c) Coding the table.
- HTML tags for the table structure: <table></table> with attributes: align, bgcolor, border, cellspacing, cellpadding, cols, width; <tr></tr> with attributes: align, bgcolor, valign; <th></th> with attributes: align, bgcolor; <td></td> with attributes: colspan, rowspan, valign.

For the tags learned, students will refer to http://www.w3schools.com/tags/default.asp for a full list of attributes and values.
- Using the tags and attributes above, teacher will demonstrate how to create a table with text, pictures, border and borderless.
- Add cellspacing and cellpadding to table.
- Spanning rows and columns.

5. Creating Image Maps

a) Intro to Image Maps
- What are image maps? Refer to page 205 of text book for graphic representation.

- Issues when using image maps: performance.
- How to avoid performance issues by creating text links.
- Server-side vs. Client-side image maps.

b) Planning the Image Maps.
- Determine if an image map is needed.
- Select an appropriate image to use as an image map.
- Sketch the hotspots on the image.
- Map the image coordinates for each hotspot – Paint can be used to locate x and y coordinates. Also there are software tools to identify coordinates. Students will be asked to locate the coordinates manually, using their algebra and geometry knowledge.

c) Coding the image map.
- A client-side image map will be created.
- Tags used to define the map: <map></map> with attributes: name and id; <area /> with attributes: shape, cords, href, alt.
- Attribute usemap=”#mapname” used with the <img> tag to display the image map.

For the tags learned, students will refer to http://www.w3schools.com/tags/default.asp for a full list of attributes and values.

6. Using Frames in a Web Site

a) Intro to Frames.
- What is a Frame? Refer to page 261 of text book for graphic representation.
- Instances when frames can be used.

b) Coding Frames – the three steps.

1. Step 1: Creating a frame definition file to define the layout of the frame in the web site. Tags used: <frameset></frameset> with attributes: cols, rows; <frame /> with attributes: frameborder, marginwidth, marginheight, noresize, name, scrolling, src, <noframes></noframes> . Note: Frames use a different DOCTYPE statement.

2. Step 2: Defining columns and rows in the frameset. Using attributes rows and cols the percentage of the space occupied by rows and columns will be defined.

3. Step 3: Defining frame attributes. The attributes of the <frame /> tag will be used to define borders, margins etc.
   For the tags learned, students will refer to http://www.w3schools.com/tags/default.asp for a full list of attributes and values.

7. Creating Forms

a) Intro to Forms.
- What are forms? Refer to page 303 of the text book for graphic representation.
- The three main components of a form: Input Controls, the <form> tag, the Submit button
- Form Input Controls: text, password, textarea, select, checkbox, radio, submit, reset.

b) Coding Forms.

- HTML tags used to create forms: <fieldset></fieldset>, <form></form> with attributes: action, method, target; <input /> with attributes: type, name, value, checked, disabled, readonly, size, maxlength, src, alt, tabindex; <legend></legend> with attribute align; <select></select> with attributes: name, size, multiple, disabled, tabindex; <option></option> with attributes: selected, disabled, value; <textarea></textarea> with attributes: name, rows, cols, disabled, readonly, tabindex.

For the tags learned, students will refer to http://www.w3schools.com/tags/default.asp for a full list of attributes and values.

- Processing the information entered in a form:

1. Using a Common Gateway Interface (CGI) to write info in a database on a server having script capabilities.

2. Posting to an email address.

- Identifying the form process: the action attribute of the <form> tag specifies the action that will be taken when the form is submitted. The information can be sent by email, or to update a database using CGI scripting.

c) Planning the Form.

- Determine what data to collect.
- Determine what types of controls to use.
- Lay out the input areas effectively.
- Use grouping techniques for clarity.

C. Integrating JavaScript and HTML

1. What is JavaScript?
    - Scripting language
    - Adds interactivity to HTML pages
    - Embedded into HTML page

2. What can JavaScript do?
    - Dynamic text into an HTML page
    - Read and write HTML elements
    - Validate data
    - Reacts to events
    - Create cookies
    - Detect the visitor's browser

3. JavaScript Basic Code.
   - The <script></script> HTML tags are used to insert a script into an HTML page.
   - JavaScript statements and blocks.
   - Variables, declaring variables, assigning values to variables
   - Operators: arithmetic, assignment; adding strings and numbers.
   - Operators: comparison, logical, conditional.
   - If, if…else statements. Switch statement.
   - Functions, the Return statement.
   - Loops: for, while, break, for…in.
   - Events: onLoad, onUnload, onFocus,onBlur, onChange, onSubmit, onMouseOver,
nMouseOut

4. Objects
   - What are objects; properties, methods.
   - JavaScript objects: String, Date, Array, Boolean, Math, RegExp.
   - Browser objects: Window, Navigator, Screen, History, Location.
   - For a complete reference of all built-in JavaScript and browser objects, visit:
http://www.w3schools.com/jsref/default.asp

5. Extracting the Current System Date and Displaying to a Web Page.
   - Use the Date() object to obtain the current system date.
   - Convert date to string using toLocaleString() method.
   - Extract month and day of the week using substring() method.
   - Locate position of the year using indexOf() method.
   - Extract year using substr() method.
   - Use the write() method to write to a web page.

6. User-Defined Functions.
   - General form.
   - User-defined functions are written in the <head> section.
   - Functions execution: calling them with a JavaScript statement.
   - Naming conventions for user-defined functions.
   - Calling JavaScript functions using event handlers.

7. Inserting a Scrolling Message on a Web Page.
   - Scrolling message basic components.
   - Creating a form and text field.
   - Creating a user defined function for scrolling message.
   - Entering an If statement.
   - Creating a recursive call by using the setTimeout() method.
   - Calling the scrolling message user defined function by using an Onload Event Handler.

8. Adding a Pop-Up Window.
   - The open() method.
   - Create a user defined function that contains the open() method.
   - Add the Event Handler to call the user defined function.

D. Using Dreamweaver to Create Web Pages.

1. Discovering Dreamweaver.

   - Defining a web site: create the local root folder and default images folder.

   Refer to Dreamweaver text book page 1-24.

   - Open Dreamweaver on the HTML screen and define a new site using the Site menu and

   he folders created.

   - Explore the Dreamweaver interface and connect to the concepts learned in this course.

   ocate formatting tabs, fonts, link box etc..

   - Use the Split screen option when adding text, pictures or editing to see how HTML or

   SS code is generated.

   - Edit code and see the results in the Design screen.
   - Open the Page Properties screen and explore the available options.
   - Open the CSS panel and explore.
   - Open all the menus and locate web development elements like tables, menus, frames, navigation bar, etc..

2. Edit an Existing Web Site with Dreamweaver.

   - Define a site for the Country Project web site, using the Country Project folder as local root folder and the Images folder as default images folder.
   - Open the Country Project home page. See if it displays correctly.
   - Use the split screen.
   - Dreamweaver as a visual check for valid/invalid code.
   - Edit the Country Project by changing backgrounds, fonts, text, colors. Edit pictures.
   - Add a new page using Dreamweaver and add content to the page.

3. Create New Pages Using Dreamweaver.

   - Create a page from existing templates
   - Use the Insert menu to create a spry menu bar.

- Define a site for the business project that was completed previously.
- Add new pages.
- Add content to the pages and edit. Use the split screen to see the code being generated.

* **Key Assignments**

1. Projects.

a) Country Project

Students will create a ten page web site presenting a country of their choice. Each student will have a different country. The ten pages will cover topics like

education, tourism, government, art, etc.. Students will choose the topics of their preference from a comprehensive list provided by the instructor. The instructor will keep track of each student’s choice.
Students will research the internet for info on the country they chose. They will gather information and save images they will include in the web pages.
This will be a basic web site project that students will complete step by step as directed by the instructor. The instructor will cover a small topic every day, like “adding images to a web page”, “editing text”, etc. and demonstrate in front of the class. Students will build their web pages with the knowledge they gain every day from the instructor’s demonstration. They will practice HTML and CSS coding by creating the ten basic web pages. This project will not cover concepts beyond tables.
Later on, this web site will be loaded in Dreamweaver and edited for practice.

b) High School Web Site Project

Students will create a high school web site that would cover the information on their high school’s published web site. They will create a home page with identical info and at least five more pages of their choice from the published web site.
This project is practice for the concepts learned while creating the Country Project. Students will work independently. Incentive: their page(s) may be published to replace currently posted page(s), if they have better look and layout.

c) Personal Web Site Project

Students will create a personal web site covering topics like family, friends, hobbies, school etc. chosen by each student. A list of possible topics will be made available by the instructor, but students will be able to choose topics not listed if they pertain to the subject.
This project will be assigned to students after more advanced features of web development were covered, before JavaScript. Students will be expected to create web pages containing image maps, forms, frames, CSS and possibly JavaScript.

d) Business Web Site Project

This is the most important project of the course. It will be assigned after all the concepts were covered, except Dreamweaver.
Students will create a web site advertising a business of their choice. The web site will have a minimum of 10 pages. Students will have to discuss their choice with the instructor first. They will be encouraged to choose a business that is familiar to them, like family business, a hobby that can be transformed in a business, a place where they work or would like to work part time, a business that they might want to pursue later in life etc.
Students will conduct an online research to see how web sites covering a similar

topic were developed and designed.
Students will carefully plan their web site first and then begin creating the web pages.
Students are expected to include all the concepts they learned, including JavaScript.
This project will be the end of the year project. It will be also loaded in Dreamweaver, edited and more pages added using Dreamweaver.
At least one page created from a template and one page containing spry menus will be added using Dreamweaver.

2. Papers

a) Use the Internet to write a paper on the following topics: (recommended site: http://www.w3.org/)

What is the World Wide Web? Short History.
What is the Internet? Short history.

b) Using the internet find definitions to the following acronyms and terminology:

HTML, HTTP, etc..

3. Assignments.

a) Realty Web Site – refer to HTML text book page 144.
Presenting information using tables with rows and columns.

b) Southwest Map Web Site – HTML text book page 202.
Creating image maps.

c) Jewelry Store Web Site – HTML text book pages 260, 302.
Using Frames in a web site.
Creating Forms on a web page.

d) Financials Web Site – HTML text book page 342.
Using CSS to enhance the look of a web page.

e) Landscaping Web Site – HTML text book page 386.
Use JavaScript to add interactive tasks to web pages: countdown message, change the color of the scroll bar, drop-down menu object, display last modified date.

f) Realty Web Site #2 – HTML text book page 434.
Use JavaScript to perform the following tasks: create a scrolling message, mortgage loan calculator, validate data entered in a form, format payment as currency, pop-up window, date last modified.

* Instructional Methods and/or Strategies

a) Lecture.

Students will have each theoretical concept presented by the instructor.

b) Demonstrations (and Email).

Students will learn by example. They will receive a short tutorial by email every day with the topic taught. The topic will be demonstrated in class and the students will apply it to their specific assignment.

c) Text Book.

Students will complete some of the more complex assignments using the text book. The concepts will be first presented by the instructor.

d) Online Research.

Students will do a lot of online research to complete the projects.
Students will be encouraged to be independent and use the internet for coding references.

e) Projects.

Students will practice the concepts learned on projects that are designed to motivate them and make them feel connected to the subject.

f) Discovery.

Students are led to discover Dreamweaver after gaining coding knowledge in HTML, CSS and JavaScript.

g) Peer Tutoring.

Students that are more technically inclined and complete the assignments early are encouraged to help their peers who struggle.

* Assessment Methods and/or Tools

a) Text book assignments – intended to see if students assimilate the more difficult concepts.

b) Individual Projects – intended to check if students are able to practice what they learned, if they feel connected and passionate about the topic they chose for the project, if they do extensive online research and use critical thinking when researching and deciding on how to develop their web pages.

c) Individual research assignments – intended to have the students discover web development concepts, terminology, history. Make students independent and develop critical thinking.

d) Unit tests – check if students are learning.

e) Every day participation points are assigned.

Print Close

Send e-mail to: hsupdate@ucop.edu

[back to top]