Sophia Learning

CS1005: Introduction to Web Development (ACE recommends 3 credits)

COURSE DESCRIPTION

Introduction to Web Development introduces students to fundamental concepts of web design and development through real-world examples and hands-on practice. Lessons combine practical skills along with broader conceptual knowledge, introducing topics in order of traditional project development methods. This progression follows the sequence from initial conceptualization and design through front-end development, briefly covering back-end concepts relevant to the final project. Topics include the history of the Internet, planning, development, document markup, scripting languages & web storage. Additional concepts like accessibility, security, and responsive design are also covered. Students will be introduced to various tools used in the industry and are encouraged to build their own personal 'toolbox' throughout the course. The browser-based programming environment is utilized to actively practice web development skills and principles. By the end of the course, students will gain a foundational understanding of basic programming that will allow for easier transfer to other languages.

COURSE EFFECTIVE DATES: June 2024 - Present

PREREQUISITES: No prerequisites

LENGTH OF COURSE: This is a self-paced course. Students may use as much or as little time as needed to complete the course.

ACE LEARNING EVALUATIONS RECOMMENDATION: In the lower-division baccalaureate category, 3 semester hours in Introduction to Web Development (ACE ID: SOPH-0043).

GRADING: This is a pass/fail course. Students must complete 10 Challenges (formative assessments), 3 Milestones (summative assessments), and 1 Touchstone (project-based or written assessment) with an overall score of 70% or better.

Challenges	Points Possible
Challenge 1.1: Introduction to Web Development	8

Challenge 1.2: Web Design	5
Challenge 1.3: Web Development	5
Challenge 2.1: Markup Languages	6
Challenge 2.2: Introduction to HTML	6
Challenge 2.3: Graphics & Media	5
Challenge 2.4: Cascading Style Sheets (CSS)	7
Challenge 3.1: JavaScript	6
Challenge 3.2: Server-Side Development	5
Challenge 3.3: Web Data Storage	6
Total	**59**

Milestones	Points Possible
Milestone 1	54
Milestone 2	72
Milestone 3	51
Total	**177**

Touchstones	Points Possible
Touchstone 3: Final Report	125
Total	**125**

Grand Total	**361**

Touchstone Tasks are required assignments that give you an opportunity to explore the course material and develop skills that you'll use throughout the course. Although Touchstone Tasks are ungraded and you will not receive feedback, the work required on them will prepare you for success on any graded Touchstone(s).

- **Touchstone Task 1: Selecting Your Client and Planning Your Website Design:** In this first Touchstone Task, students will select a client for their project, begin building wireframes, and make design decisions based on client need and web design best practices. They will document their progress on the provided template file, providing an analysis of design choices.
- **Touchstone Task 2.1: Creating HTML Pages:** In this second Touchstone Task, students will create 4 HTML web pages for their selected client. They will begin by reviewing information about their selected client to ensure all expectations are met. They will finish by documenting any issues that prevented the code from functioning as intended.

- **Touchstone Task 2.2: Adding Style to the Webpages Using CSS:** In this third Touchstone Task, students will create an external CSS file to include CSS styling on the web pages for their selected client. They will write a brief description of CSS styling choices applied to each web page and provide an analysis on accessibility practices they followed in the website. Finally, they will document any issues you had that prevented the code from functioning as intended.
- **Touchstone Task 3.1: Implementing Dynamic Features with JavaScript:** In this fourth Touchstone Task, students will add dynamic features to their web pages using JavaScript. They will begin adding a functional alert window for the subscribe feature in the footer of every web. For the Gallery page, they will create alert windows for the add-to-cart feature. For the About Us/Contact page, they will create a functional alert window for the web form. Additionally, they will provide an analysis on the JavaScript functionality and document any issues they had that prevented the code from functioning as intended.
- **Touchstone Task 3.2: Storing Data with Web Storage:** In this final Touchstone Task, students will store and retrieve data using web storage. They will use session storage to add items to the shopping cart as a part of the implementation of the “add to cart” feature on the Gallery web page. They will retrieve information about added items in the shopping cart using sessionStorage as a part of the implementation of the “view cart” feature on the Gallery web page. They will add custom order information to the localStorage from the web form in the About Us/Contact web page. Additionally, they will provide an analysis on the functionality and document any issues they had that prevented the code from functioning as intended.

Touchstones are projects that illustrate comprehension of the course material, help refine skills, and demonstrate application of knowledge. Read further for information on the touchstone in this course:

- **Touchstone 3: Final Report (125 points):** The final project provides 3 client scenarios to choose from and guides learners along the process of designing and developing a website to meet the client’s needs. The submission will include the following details: the student’s design plan and preparation steps before creating the site, their final project code, their record of testing and debugging efforts for any code that doesn't function properly, and rationale for how their specific design choices meet current accessibility standards. Learners will be evaluated on their understanding and accomplishment of the development process, rather than the particular details of their code itself or its sophistication.

For more general information on assessments, please visit the Student Guide located on your course dashboard.

LEARNING OUTCOMES

Upon completion of the course, the student will be able to:

1. Demonstrate how web development has changed over the years.
2. Demonstrate how the document markup can impact a website.
3. Use scripting languages to build a functional website that meets client needs.

COURSE COMPETENCIES AND TOPICS

Unit	Competencies	Major Topics
1. Web Development	• Understand the origins of the Internet and its impact on web development. • Identify fundamental skills needed in Web Design. • Identify fundamental skills needed in Web Development.	• Brief History of the Internet • The OSI Model and Internet Protocols • Web Development Resources • Wireframes, Storyboarding, and UI/UX Design Concepts • The Fold • Typography & Web Fonts • Search Engine Optimization (SEO) & Data Analytics • Project Management Methods • Structure Patterns and Responsive Design • Best Practices for Web Development
2. Document Markup	• Define the roles of different markup languages in web development. • Identify the purpose of different tags in an HTML file. • Identify different types of graphics and media files that can be used in a website. • Demonstrate the different ways that CSS can be added to a website.	• Introduction to Markup Languages • Best Practices for Web Development • HyperText Markup Language (HTML) • Cascading Style Sheets (CSS) • Document Object Model (DOM) • Video, Audio, and Graphics Formats • The Box Model, Rule Structure, and Order of Precedence • Font & Text Decoration • Responsive Styling • Accessibility
3. Scripting Languages & Storage	• Identify how JavaScript is used in web development. • Identify how PHP is used in web development. • Use data manipulation for web development.	• Introduction to Server-Side and Client-Side Scripting • JavaScript, JavaScript Object Notation (JSON), and Extensible Markup Language (XML) • Asynchronous JavaScript (AJAX) and the jQuery Library • PHP • Cookies, Sessions, & File Interaction

		• Database Management Basics • Security Risks