

Computer Science & Web Design

Course Description

This course introduces students to the foundations of computer science with a focus on how computing powers the world. Students will explore computer science terminology and concepts and apply them to a culminating project using programming language to solve a problem. This course also introduces students to basic web design using HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets). Students will be introduced to planning and designing effective web pages; creating web pages by writing HTML and CSS code, and producing a functional, multi-page website.

Course Code: 8675309

Endorsements to teach: Basic Business, BMIT, Math, Science

Standard 1

Develop foundational skills and knowledge in computer science.

- Indicator 1.1** Map a brief overview of computer science history.
- Indicator 1.2** Show how binary bits can be used to create numbers, characters, letters, images, audio and video.
- Indicator 1.3** Consider the pros and cons of the variety of computing languages.
- Indicator 1.4** Appraise the current opportunities available in computer science career fields.
- Indicator 1.5** Articulate common vocabulary in computer science. (algorithms, byte, bit, binary, computational thinking, etc.)
- Indicator 1.6** Learn the steps of a design cycle and how it works and how it works as a practical problem solving method.
- Indicator 1.7** Understand what comprises a computer: input device, memory, output, control unit, arithmetic/logic unit. (Von Neumann architecture.)
- Indicator 1.8** Explore the impact that modern computing has on the world.
- Indicator 1.9** Discuss the world wide web, history, trends, and trajectories.

Standard 2

Analyze the design process in regards to audience, layout, purpose, and current standards of web development.

- Indicator 2.1** Identify the target audience and the need for the website.
- Indicator 2.2** Define the content for a website.
- Indicator 2.3** Review the value of using optimal photos, graphics and text on a website.
- Indicator 2.4** Create a design for a website.

Standard 3

Learn to read a write in the Hypertext Markup Language for a purpose. (HTML)

- Indicator 3.1** Memorize basic tags employed in current HTML best practices.
- Indicator 3.2** Identify tags used on an existing page.
- Indicator 3.3** Modify existing webpage tags to change structure, improve design, or alter the webpage purpose.
- Indicator 3.4** Construct an HTML page composed of a multitude of tags that demonstrates planning and purpose.

Standard 4

Understand and apply CSS to format web page elements.

- Indicator 4.1** Evaluate the importance of CSS.
- Indicator 4.2** Distinguish the attributes of CSS. (padding, border, hexadecimal color codes, div, absolute and relative positioning, etc.)
- Indicator 4.3** Enhance web pages by using text formatting, color, graphics, images, and multimedia.
- Indicator 4.4** Apply CSS elements to a web page.

Standard 5

Construct a computational artifact using a programming language. (computational artifact is the final product of what is programmed)

- Indicator 5.1** Develop language literacy for a programming language. (debugging, variables, sequence, input-output, conditionals, loops, syntax, functions, algorithms)
- Indicator 5.2** Use computational thinking to explain how simple algorithms work and to detect and correct errors in algorithms and programs.
- Indicator 5.3** Design, write and debug simple programs that accomplish specific goals.
- Indicator 5.4** Using a programming language, design and develop an independent program to solve a problem and demonstrate creativity.

Contributors:

Susan Schiltz
 Tamara Zentic
 Ramsey Young

Resources: code.org, W3School, Snap, Scratch, App Inventor, Python, javascript, Java