Web Development and Design

Grades 9 – 12 Semester Elective (16 weeks)

By Miranda Moore – CTE Course Construction, Assignment 5, UW Stout

Course Philosophy

Computer science students learn logical reasoning, algorithmic thinking, design and structured problem solving—all concepts and skills that are valuable well beyond the computer science classroom. Students gain awareness of the resources required to implement, test, and deploy a solution and how to deal with real-world constraints. These skills are applicable in many contexts, from science and engineering to the humanities and business, and they have enabled deeper understanding in these and other areas. Computer simulations are essential to the discovery and understanding fundamentals of how the human mind works, one of the great intellectual challenges of all time. Thus, much computer-enabled innovation lies ahead of us and computer science is an essential tool for achieving our vast potential.

Many careers in the 21st century require an understanding of computer science and some of the jobs today's students will have in 10 to 20 years haven't been invented yet, and will likely require more than just an understanding. Professionals in every discipline—from artists and entertainers, to communications and health care professionals, to factory workers, small business owners, and retail store staff—need to understand computing to be productive and competitive in their fields. Studying computer science can prepare a student to enter many career areas, both within and outside of computing. Professionals with computer science training have never been more in demand than they are today. Computing scientists are working with experts in other fields, designing and building computer systems that support the functioning of modern society and are enabling us to tackle the critical challenges that face our world. In addition, computing skills are now preferred, if not required, for work in almost any profession.

Course Outline

Prerequisites: General computer knowledge is required. A strong background in mathematics and science is recommended.

Course Goals:

- Knowledge of the fundamental components that make a website function
- Ability to design multiple websites of varying difficulty levels using a project based approach building a website from beginning to end
- How to launch and promote a fully functional website.
- Educational tools that will allow students to continue learning and building websites outside the classroom

Course Description: Students will develop competent web development skills through hands-on experiences which use real-life aptitudes such as reading, writing, imagination, self-expression, problem solving, attention to detail, work ethic, follow through, and communication skills. Attention will also be given to the design, organization/structure of formal elements, and developmental use in web programming. Ethical and social issues in computing, careers in computing, and responsible computing are woven in throughout this course.

Course Objectives:

- Identify the major components and concepts of computers and the suitability of hardware and software integrations, as well as identify the many different computing fields.
- Define developmentally appropriate, accurate terminology when communicating about technology.
- Identify ethical behavior with respect to intellectual property rights, software piracy, digital media use and explore implications.
- Analyze technological limitations on design.
- Apply a variety of computing solution techniques for real-world problems that use creativity and innovative thinking.
- Develop content-specific models and simulations to support learning and research.
- Exhibit dispositions necessary for collaboration: providing useful feedback, integrating feedback, understanding and accepting multiple perspectives, socialization.
- Design, develop, publish, and present products (e.g., webpages, mobile applications, animations) using editing technology to effectively display, manipulate, and provide information through various types of media (text, images, sound, etc.).
- Perform client-side and server-side manipulation using scripting languages.
- Identify client needs, analyze potential end-user issues, describe data requirements, present documentation, and produce data throughout a project lifecycle.

Course Resources:

Computer/laptop

Internet services

Web Design with HTML, CSS, JavaScript and jQuery Set 1st Edition by Jon Duckett

Site Point HTML5 & CSS3 for the Real World: 2nd Edition

Programming the Web Using XHTML and JavaScript
McGraw-Hill Irwin – Larry Randles Lagerstrom, University of California-Davis

Handouts and materials

MOTIVATION!

Units of Instruction:

Unit: 1 – Communication Systems: Community, Global, and Ethical Impacts

Objective: Gain understanding of how technology affects people and our environment in both positive and negative ways.

Content Outline:

1. Technology evaluations, how we use devices and why – after? (PowerPoint)
2. Determine the reliability of information found on the Internet
3. Ethical use of modern communication devices
4. Explore a variety of careers to which computing is central
5. Describe security and privacy issues that relate to computer networks

Learning Activity: Reading. Technology discussions. Explore a variety of careers to which computing is central. Students will choose a profession and write a 1-2-page summary of how technology influences that career. By purchasing and using technology, we all take on the responsibility to ethically dispose of it when finished.
See Appendix A.

Evaluation Criteria 40 points: No spelling errors, well used grammar, and a clear connection between the career chosen and influential technologies.

Unit: 2 – Computational Thinking and Problem Solving

Objective: Pattern recognition and data analysis. Explore how computing has facilitated new methods of managing and interpreting data. Translate, process and visualize data to find patterns and test hypotheses. Illustrate how widespread access to data and information facilitates identification of problems.

Content Outline:

1. Use the basic steps in problem solving (e.g., problem statement and exploration, examination of sample instances, design, implementing a solution, testing, evaluation)
2. Evaluate how different solutions may be used to solve the same problem
3. Describe and analyze a sequence of instructions (algorithms, OOP, PP)
4. Collaborate in data collection
5. Represent data in a variety of ways including text, sounds, pictures, and numbers
6. Discuss security implications

Learning Activity: Reading. Lab – Group Assignment. Individual Assignment – Write an algorithm. Choose an activity to describe and analyze a sequence of instructions.
See Appendix B.

Evaluation Criteria:

Algorithm 35 points – sequence of instruction is logical and thorough (idea is to have enough information as though you were programming a computer do that activity).

Group Activity 40 points – participation is required. Grading will be based on group effort – your ability to work together and the quality of the presentation.

Unit: 3 – Internet Basics

Objective: Differentiate among open source, freeware, and proprietary software licenses and their applicability to different types of software. Understand developing parameters and the issues various devices and bandwidth availability may pose. Define what is involved when starting up a website.

Content Outline:

1. Bandwidth, Cache, Transfer Rate
2. Domain Names, Naming Directories and Files
3. Browsers, FTP
4. Domain Hosting
5. File Types
6. Explore principles of system design in scaling, efficiency, and security.

Learning Activity: Reading. Lab – Discover file sizes and download times per bandwidth setting; view coequal sites in popular browsers (Chrome, IE, Firefox, Safari).

Evaluation Criteria 50 points: Essay describing three of the topics under Content Outline (Outline is located in appendix C).

Unit: 4 – Web Design and Programming

Objective: Take on the role of a developer, create web pages and documentation for users and equipment. Plan and code web pages using a variety of techniques and check for usability. Describe a variety of programming languages available to solve problems and develop systems.

Content Outline:

1. Design
    a. Topography
        i. Color
        ii. Fonts
    b. Multimedia
        i. Graphic and image formats, video, audio.
    c. SEO
    d. Navigation and Search
2. Programming – (PowerPoint)
    a. HTML, XHTML, HTML5
    b. CSS, CSS3
    c. Explore principles of system design in scaling, efficiency, and security

d. JavaScript, jQuery (animation)
e. Theory: other languages and their appropriate uses

Learning Activity: Reading. Build homepage features as a class; student chosen topic.

- Activity – Design a home page for a made-up product website.
- Conditions – May not look like the existing one. Use at least 2 multimedia forms. Include at least two paragraphs of text. Show navigation placement, copyright, and company's contact information (phone number, address, email). Design a simple logo.
- Standards – Must show good use of white space, placement of graphics. Font is legible and color use is creative. Product does not have to be realistic, but text must provide information on your creation. Site page needs to be organized for quick information gathering and the layout needs to cater to the interactive user.

See appendix D.

Evaluation Criteria 60 points: Participation and full execution of project.

Unit: 5 – Computing and Data Analysis

Objective: Design, develop, publish, and present products (e.g., videos, websites) using technology resources that demonstrate and communicate curriculum concepts. Use various debugging and testing methods to ensure program correctness.

Content Outline: Final project – design your own website. Criteria in Appendix F.

Learning Activity: Reading. Create an outline or flowchart of your website, use design software to do a mock-up of what you would like your site to look like. See final project options in Appendix E.

Evaluation Criteria:

Mock-up 60 points

Final Project 200 points

1. Speed – 10 points
   a. The homepage downloads efficiently.
2. Home page – 30 points
   a. The homepage is attractive, has strong design appeal.
   b. You can tell where you are immediately (title, description, image captions, etc.)
   c. There is a menu giving a clear indication of the site contents.
   d. Site purpose is clearly identified.
   e. Contact information is readily available.
   f. Copyright date or date site was established is easy to determine.
3. Ease of navigation – 40 points
   a. User is able to move around within the site with ease.

b. Directions for using the site are provided if necessary.
c. Directions are clear and easy to follow.
d. Information is easy to find (no more than three clicks, for example).
e. The links to other pages within the site are helpful and appropriate.
f. Internal and external links are working properly (no dead or incorrect links, etc.)

4. Use of multimedia – 30 points
a. Each graphic, audio file, video file, etc., serves a clear purpose and uses alt tag.

5. Browser compatibility – 40 points
a. Site is equally effective with a variety of browsers (Chrome, IE, Firefox, Safari).

6. Content – 40 points
a. There is sufficient information to make the site worth visiting.
b. The information has headings and is organized.
c. The same basic format is used consistently throughout site.
d. Lists of links are well organized and easy to use.

7. Availability of further information – 10 points
a. Links to other useful Web sites or citations are provided.

APPENDIX A

Unit 1 Items:

Computer Based Training on E-Waste PowerPoint presentation: E-Waste.pptx

APPENDIX B

Unit 2 Items:

PowerPoint presentation: algorithms_pp_oop.pptx

In class group assignment: (International Society for Technology in Education (ISTE); Computer Science Teachers Association (CSTA); National Science Foundation (NSF) , 2011)

Problem statement: At a local school there is only one entrance to the driveway that leads to the front of the school. All vehicles are required to drive through this entrance to drop the students off at school in the morning. There are 1200 students at the school and the majority of them are driven to school with only one student in each car. Traffic enters the driveway from both directions, which forms a T intersection with the school. Vehicles make a right or a left turn to the school driveway and there is no traffic light, stop sign, or crossing guard at the entrance. This causes excessive traffic back up and safety issues during peak hours.

1. Break up into small work groups and identifying variables that have caused the problem. (e.g. a lack of car-pooling, student behavior, etc.)
2. Create a map or diagram that illustrates the properties that match the variables in the problem. Using the variables identified, mark up the map with the various variables causing the problem.
3. Collect data and do research on the variables identified as causing the traffic problem. This research could involve [data collection] at the source of the problem, surveys, interviews, and web-based research.
4. Define relationships among all of the variables. Each group will create a model that shows the interdependency of the identified variables. The model may be diagram, map, graphs, or 3D, and could be computer-generated.
5. Put together a portfolio of graphical representations of the relationships of their variables. Using a variety of types of graphs and charts (e.g., pie charts, bar charts, graphs, maps, etc.).
6. Estimate the influence of each variable by giving it a magnitude scale. Likely variables for system: number of students vs. number of cars, age of drivers, time waiting in traffic, schedule, numbers of entrances, local traffic codes, etc.
7. Each group will show and describe how the model will change if variables change (e.g., adding an entrance to the school).
8. Research and study which variables can be changed and how difficult it is to change them. Form a hypothesis of what the effects of changing some of the variables might be, and how the outcome will change.
9. As a group write up and create a presentation of the findings.

APPENDIX C

Internet Basics Essay – Unit 3

Introduction

A. Capture reader's interest
B. Build case through logic
C. Topic sentence/thesis statement

First main point (strongest)

A. Support
    1. Example
    2. Example
B. Support
    1. Example
    2. Example

Second main point

A. Support
    1. Example
    2. Example
B. Support
    1. Example
    2. Example

Third main point

A. Support
    a. Example
    b. Example
B. Support
    a. Example
    b. Example

Conclusion

A. Restate topic
B. Summarize three main points
C. Revisit introduction or tie all ideas together

APPENDIX D

Introduction to Web Design – Unit 4

PowerPoint presentation: intro2webDesign.pptx

(Handout)
Skeleton Page:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset=utf-8" />
<meta name=”viewport” content=”width=device-width, initial-scale=1” />
    <title>Page Title Goes Here</title>
  </head>
<body>
    <!-- Page content goes here -->
</body>
</html>
```

<!DOCTYPE html> The first thing on an HTML page, tells browser which version of the markup language the page is using. Here we are telling it that this is an HTML5 page.

<html lang=”en”> After the decloration, we state the html tag and tell the browser we are using the English language in our content. Everything in an HTML web page, must begin with the opening <html> tag and end with the HTML closing tag </html>.

<head> Information about the page, like its title and linked css or js files and the meta information are contained in the head. Must use closing tag </head>.

<meta /> It’s best practice to include meta tags that tell the browser the charset and viewport.

<body> The actual content of the page – what the browser displays. This must also end the page with its closing tag </body>.

Syntax Rules:

- All elements must be in lowercase.
- Each element must have an opening and closing tag.
- Self-closing elements must have trailing slash.
- Elements must be properly nested.
- White space doesn’t matter.
- Attributes must be in lowercase and values must be enclosed by quotes.

Comment Syntax:

```
<!-- This is a comment -->
```

Global Attributes:

class - assigns one of more CSS classes to an element for styling purposes.
id - assigns a unique ID to an element for CSS styling or other purposes.
style - defines an inline CSS style for a specific element.

Images

Display an image in your web page:

src	where the file is located
alt	description that shows when visitor hovers over image
name	naming the image is important if you add any sort of animation
class	styling linked to css

Image Formats (Matthews, n.d.)

PNG is also a lossless storage format. However, in contrast with common TIFF usage, it looks for patterns in the image that it can use to compress file size. The compression is exactly reversible, so the image is recovered exactly. Supports images without a full background.

GIF creates a table of up to 256 colors. Sometimes GIF uses the nearest color to represent each pixel, and sometimes it uses "error diffusion" to adjust the color of nearby pixels to correct for the error in each pixel. GIF is "lossless" only for images with 256 colors or less. For a rich, true color image, GIF may "lose" 99.998% of the colors. Can be animated.

JPG is optimized for photographs and similar continuous tone images that contain many, many colors. It can achieve astounding compression ratios even while maintaining very high image quality. GIF compression is unkind to such images. It stores information as 24 bit color. Important: the degree of compression of JPG is adjustable. At moderate compression levels of photographic images, it is very difficult for the eye to discern any difference from the original, even at extreme magnification. Compression factors of more than 20 are often quite acceptable.

BMP is an uncompressed proprietary format invented by Microsoft. There is really no reason to ever use this format.

APPENDIX E

Final Project Options: Ideas contributed by (Morgan, n.d.)

1. An autobiographical website project fuses high school students' passion for self-expression with the basics of Web publishing, allowing them to practice designing a layout, inserting graphics and formatting text. Students can produce websites that showcase their interests and important elements of their lives. For example, they might create pages that discuss their school activities, share their favorite sports, music and movies and describe their college and career plans. A personal website can provide simple subject matter for students to practice writing their own HTML coding.

2. Having a website for a high school club can draw new members to the group and inform current participants and their parents of upcoming events. Talk to club advisers at your school to find out what organizations don't have their own home pages. Then, put students in groups to design a website for one of those clubs. Students in each group can interview the adviser and participants as research, collect pictures for a photo album of their activities and post a calendar of upcoming events. Having students write about unfamiliar subject matter can introduce them to the skill of writing clear, direct copy to inform readers about these organizations.

3. Like argumentative speeches and essays, websites can persuade readers to change their position on debatable topics. Students can design websites that take specific positions on issues that affect teenagers, such as eating disorders, bullying, cellphones in schools and depression. After researching their topic, they can design sections describing the issue, its consequences and possible solutions and actions readers can take to help. Because strong Web copy related to persuasive issues avoids extreme language or biases, students will practice presenting the facts in a way that gives an even portrayal of the subject, as well as selecting reliable, credible sources for references.

4. Your students' passion for music, movies, books and other entertainment can become fertile subject matter for their Web design projects. Have students create websites centered around their favorite celebrities. For example, a website about the Beatles might include a brief biography of the band, a list of their albums, links to YouTube videos of their performances and a timeline of significant moments in their career. Because students will be linking their sites to music files and video clips, they also can practice writing clear links by having viewers click on keywords in sentences, rather than vague phrases like "Click here."

APPENDIX F

Final Project Criteria

1. 4 page site (minimum)
2. All pages must have an IDENTICAL navigation bar - users should be able to get TO all pages FROM all pages (use a table for your nav-bar)
3. Make sure all pages have titles ("labels" would be good as well)
4. The "Homepage" must be saved as "index.html"
5. Make the Homepage cool and interesting - this is where your users/viewers "meet" your site - make people want to stay and check things out; let the viewer know where he/she is, what the site is about, and why he/she would want to browse
6. Include a clear navigation menu with appropriately named links – use <header><nav>
7. One of your pages MUST include a photo gallery: use image thumbnails as links to save space (6-8 pics would be good) - make sure the thumbnails link to LARGER versions of the thumbnail image – Use LightBox
8. Show off your table-making skills (colspan, rowspan, nested tables, etc...)
9. Include "footer information" at the bottom of each page: name of site, contact info, etc...
10. Contain at least 1 email link used for contact/more information
11. Use bold, underlined, and italicized text
12. Use different font faces, sizes, and colors
13. Use of blockquote tags
14. Show examples of different link colors. For example: visited, hover, etc.
15. Use Heading tags where appropriate
16. Make use of appropriate paragraph tags and line breaks
17. Provide examples of things aligned to the left, center, right
18. at least 1 animated GIF (it should be RELEVANT to your site)
19. Has some external links - cool sites you like to visit (can be text links or image links)
20. External links should open in a new window using target="_blank"
21. Use of alt text with all images, videos, etc.

*Be sure to put some thought into both the DESIGN aspect of your site (how things look and feel: colors, spacing, alignments, fonts, placements, etc...) and also to USER FRIENDLINESS: (easy to navigate, easy to read/view, people know where they are at any given time - "labeled" pages, no unnecessary scrolling and/or clicking to get to info, etc...)

Upload all files to class folder “Final Projects/your last name”

CLASS INFORMATION

Course Grading:

Total points possible 500.

500 – 450 = A 450 – 390 = B 390 – 320 = C 320 – 250 = D 250 – 0 = F

Attendance:
Students are expected to be on time for each class period. If a student is absent, it is the student's responsibility to get learning materials and make up any work missed. Multiple absences will affect student performance. Please notify the instructor if you are unable to complete or participate in any part of this class. Special accommodations will be made on an individual basis.

Submitting computer files: In order for assignments submitted electronically to be readable and graded according to the appropriate assignment criteria, please follow these guidelines:

- Save the document either in MS Word or as a .pdf file
- Your filename should include your last name, first initial, and the assignment name
- Upload assignment to appropriate class folder

(Simoneau, 2017)

Email communications: To receive adequate attention, please follow these guidelines:

- Include the course number in the subject line, e.g. CTE 302, followed by a meaningful title to identify the topic of your message
- Be specific and direct in your message.
- Be friendly and cordial, yet cautious in using humor or sarcasm. I appreciate a sense of humor but just as with written discussions, it can be misinterpreted.
- Use standard spelling, punctuation, and capitalization. NO ALL CAPS even if you feel like screaming about something.
- I will try to respond as quickly as possible, but my schedule may prevent me from doing so. Please allow 24 hours so that I can be thoughtful in my response to you. If you send an email any time between Friday evening and Sunday evening or during a holiday/vacation, the response time may be longer.
- Additional email guidelines and examples can be found at http://owl.english.purdue.edu/owl/resource/636/01/

(Simoneau, 2017)

Student based learning – does not mean that the learning is centered around the student, it means that the student is responsible for their own learning.

References

Build A Module. (2013, November 7). *The difference between procedural and object-oriented programming* . Retrieved from YouTube: https://www.youtube.com/watch?v=OEfSFrk_KEI

Computer Science Teachers Association, Association for Computing Machinery. (2011-2016). *CSTA K-12 Computer Science Standards: Mapped to Common Core State Standards* . Retrieved from CS Teachers.org: http://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/Docs/Standards/CSTA_Standards_Mapped_to_Com.pdf

CTSA Standards Task Force. (2011). *CSTA K-12 Computer Science Standards.* Retrieved from Computer Science Teachers Association: http://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/Docs/Standards/CSTA_K-12_CSS.pdf

ECS Team. (2017). *Scope and Sequence*. Retrieved from Exploring Computer Science: http://www.exploringcs.org/curriculum

Gupta, D. (n.d.). *Why every programmer should learn to optimize algorithms*. Retrieved from Programiz: https://www.programiz.com/blog/scalability-algorithms

High School Web Design Team. (n.d.). *Introduction to Web Design.* Retrieved from High School Web Design: http://hswd-wpengine.netdna-ssl.com/files/Introduction-Web-Design-Curriculum-Inventory.pdf

International Society for Technology in Education (ISTE); Computer Science Teachers Association (CSTA); National Science Foundation (NSF) . (2011). *Introduction to Computational Thinking*. Retrieved from Teachers Resources: https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/472.11CTTeacherResources_2ed.pdf

Jasso, J. (n.d.). *Algorithm Examples - Fundamentals You Should Know*. Retrieved from MATLAB: http://www.matrixlab-examples.com/algorithm-examples.html

Matthews, R. (n.d.). *Digital Image File Types Explained*. Retrieved from Wake Forest Student, Faculty and Staff Web Pages: http://users.wfu.edu/matthews/misc/graphics/formats/formats.html

Morgan, K. (n.d.). *Web Page Design Project Ideas for High School Students*. Retrieved from Classroom: http://classroom.synonym.com/design-ideas-high-school-students-6567640.html

Simoneau, M. (2017, January 19). *CTE 438/638 Course Construction in Career and Technical Education Syllabus.* Retrieved from UW Stout D2L Course Construction: https://uwstout.courses.wisconsin.edu/d2l/le/content/3645318/viewContent/22351878/View