

## CSCE 102 - General Applications Programming

1. Course number and name: CSCE 102 - General Applications Programming
2. Credit: 3-hrs; Contact: 2 lectures of 50 minutes and a lab per week
3. Instructor: Spring 2012: Catherine T. Matthews
4. Text books: Terry Felke-Morris, *Basics of web design HTML5 & CSS (2<sup>nd</sup> Edition)*, ISBN 978-0133128918 and Jeremy Keith with Jeffery Sambells, *DOM Scripting, Web Design with JavaScript and the Document Object Model (2<sup>nd</sup> Edition)*, ISBN 978-1430233893
5. Specific course information
  - a. Catalog description: Introduction to systematic computer problem-solving and programming for a variety of applications. Open to all majors.
  - b. Prerequisites: Two years of college preparatory mathematics or equivalent
7. Learning outcomes. Specific outcomes of instruction are that students will be able to:
  1. Demonstrate the ability to find a solution and write an algorithm when given an English description of a task to be accomplished (that is, a problem statement)
  2. Demonstrate the ability write, execute, test, and debug computer programs in a high-level language
  3. Demonstrate the ability to modify a computer program written in a high-level language
  4. Demonstrate mastery and use of concepts and terminology related to computer programming and the web.
7. Topics covered and approximate weight (14 weeks, 3 hours/week, 42 hours total)
  - a. Introduction to the Web/Internet/Browsers/Search engines/Web pages (2 hours)
  - b. Writing code to create and apply style to web pages (4 hours)
  - c. Programming languages/Introduction to programming (2 hours)
  - d. Problem solving/Algorithms/Data/Information (4 hours)
  - e. Problem solving and programming for the web to create Interactive web pages (6 hours)
  - f. Variables/data types, numbers (integers, floating point), strings, Boolean, arrays (3 hours)
  - g. Logical flow (3 hours)
  - h. Debugging/Testing (3 hours)
  - i. Conditional logic (if/else, not, and/or) (4 hours)
  - j. Loops (3 hours)
  - k. Functions (2 hours)
  - l. Reviews and Examinations (6 hours)
8. Grade consists of:
  - Tests 30%
  - Final Exam 20%
  - Lab 30%
  - Project 5%
  - Quizzes/Homework 15%

Lab Grades

The grade is calculated using the standard curve:

A: 90-100,  
B+: 87-89,  
B: 80-86,  
C+: 77-79,  
C: 70-76,  
D+: 67-69,  
D: 60-66,  
F: <60

You must pass (average on must be  $\geq 60$ ) the lab portion of the course in order to pass the class.

## 9. Specific topics covered.

**Week 1:** Introduction, web browsers, search engines, computer languages, algorithms, pseudo-code, logical flow, testing, creating a web page with a text editor, HTML5, structure, elements, paragraph, break, blockquote, lists (ordered, unordered, description) phrase elements, block-level, special entity characters, testing, validation

**Week 2:** Hyperlinks (absolute, relative), email hyperlinks, fragment identifiers, web graphics, using images, an image as a link, introduction to Cascading Style Sheets, syntax

**Week 3:** CSS (inline/local, embedded, external), using color hexadecimal value representation, layout and design, classes, inheritance, id, contextual selectors, span, div, validation, transform-rotate, The Box Model, absolute/ relative positioning, magazine style layout, pseudo-classes, interactivity with CSS, creating image rollovers, writing JavaScript functions

**Week 4:** CSS continued multiple background images on one page, icon in the URL box, customizing bullets. Computer languages, compilers/interpreters, errors syntax/logical, programming constructs (sequential, conditional, looping, go to, functions), logical flow, introduction to objects/methods, data representation, input data, test data, information, interactive web pages, introduction to JavaScript, W3C, using built in methods (alert/prompt/confirm)

**Week 5:** Variables (declaring, naming, and values), data types, strings, number (integer, floating point) Boolean, arrays, strongly/weakly typed, assigning values to variables and properties, operations, concatenation, mathematical operations, interactive web page through user input, event handlers, changing the CSS properties on a web page after it has loaded, assigning values to CSS properties, using: `this.style.`, `getElementById().style.`, and `document.body.style`, debugging and testing

**Week 6:** Test review, test 1

**Week 7:** Return test, Problems requiring decisions, conditional flow, if/else, nesting if statements, complex conditions, and/or, `toUpperCase()`

**Week 8:** Image rollovers in JavaScript, creating tables, data in tables as text or images, data in online tables accessible to the blind, configuring data into sections for a more informative display of information, using thumbnails, structural pseudo-classes

**Week 9:** Writing code to create online forms to input data, text boxes, using the `Math.` methods, creating online specialized calculators, converting strings to floating point, `parseFloat`, formatting an input data form using a table, `fieldset` and `legend`, code for helping people with vision and mobility handicaps with

input data forms, writing code for improved data input forms using CSS, testing data input forms, text form controls, server-side processing, the datalist element (Firefox), create slider and spinner controls, date control /calendar date selection, creating check boxes

**Week 10:** Review for test 2, test 2

**Week 11:** Return test 2. Using loops, writing code to password protect (looping); for loops, do loops, do while loops

**Week 12:** Using loops to create an online slideshow, coin flip problem, including audio and video in your web page

**Week 13:** Creating radio buttons, creating a pull down menu, optimization for search engines, accessibility testing, usability testing

**Week 14:** Exam Review