Course Syllabus

Home Department

WDDBS Web Design and Development – Bachelor of Science

Course Name

Web Programming Fundamentals Online

Contact Information

Catalog Course Code:	WDD 144-O
Three-Letter Course Abbreviation:	WPF-O
Instructor:	Jessica Garlic
Telephone:	407.679.0100 ext 2408
Email:	jgarlic@fullsail.com
iChat:	jcurry@fullsail.com (AIM)
Hours: Indicate office hours, chat hours, and preferred method of contact.	Office hours are by appointment. Please email me with your schedule and let me know when you are free.

Course Description

The **Web Programming Fundamentals** course introduces basic programming and logic skills. Students learn how to break down processes and tasks into smaller, programmable parts and translate them into source code. JavaScript, the language available in every Web browser, is used to teach these cornerstone concepts.

Course Materials

- Beginning JavaScript (4th Edition): by Paul Wilton, Jeremy McPeak
- Notebook or other device for note-taking
- Code editor of your choice
- Lynda.com subscription (provided)
- Github.com account

Course Objectives

- Learn basic programming components, logic and terminology
    - Create and use values and variables for storing those values.
    - Understand the difference between compiled and interpreted coding languages.
- Learn to debug code.
    - Understand the relationship between HTML and JavaScript files.
    - Use commenting to describe and document JavaScript code
    - Identify errors, what causes them and how to solve them.
- Use conditional statements for dynamic decision-making.
    - Utilize if, else if and else to create branches for logic.
    - Use Boolean and relational operators to create conditions
- Learn to create functions and use existing functions to execute blocks of code.
    - Utilize variable scope
    - Create and apply arguments and parameters for transferring data

Course Outcomes

Upon successful completion of this course, students will be able to:

- Understand and use basic programming components and terminology
- Use critical thinking to solve problems
- Use loops and arrays to track sets of objects
- Create modular code using functions

General Education Component

Web Programming Fundamentals is greatly supported by College Mathematics, Geometry & Measurement and Fundamentals of Physics general education courses. Programming in any language relies on a base understanding of manipulating variables as provided in the algebra of College Mathematics. Understanding of graphing techniques, solving for distances and angles as developed in Geometry & Measurement is essential in placing, animating and manipulating visual objects programmatically. Scientific formulas and equations students explore in Fundamentals of Physics are required in one of the lab assignments students will submit in the WPF coursework.

Degree Connection

Just over halfway through the Web Design and Development Degree program (month 4) Web Programming Fundamentals serves as the student's introduction to programming. The class is the base of the series of JavaScript-based classes in the program including Programming for Web Applications I and II, Web Interaction and Animation, Web Standards Project, and Front-end Frameworks . The basic programming concepts learned are essential as the base for the server-side language classes near the end of the degree program such as Design Patterns for Web Programming, Server-Side Languages,

Advanced Server-Side Languages and Advanced Topics in Web Programming.

Industry Connection

This class is designed to introduce the basics of programming to the student. While using JavaScript as a learning tool, the elements of language learned in the class can be applied to all of the 20 most popular programming languages in use today. In addition to concepts, industry standards and conventions will be covered so that students can produce strict, clean, efficient and well-documented code. The class will be essential to developers seeking to learn this and all other programming languages used in the industry and important to designers who will have to interface their artwork with the dynamically driven sites used today.

Research Component

While learning programming is a difficult step on the road to becoming a web developer, it is a necessary one. Students look at diverse tools and models of thought for logic and learning to program by researching M.I.T's learning application for programmers called Scratch. Students must research, use and create an application in the Scratch environment as part of this course.

Additional Resources

Here is a list of some of the online guides, code documentation and references we will be using throughout the course:

- Lynda.com
- Beginning JavaScript (4th Edition): by Paul Wilton, Jeremy McPeak
- Net.tutsplus.com
- CodeAcademy.com
- LightBot 2: http://www.silvergames.com/light-bot-2

Topics Covered

- Basic programming terminology
- Values, variables, arrays and expressions and basic JavaScript syntax
- Set up and use of an HTML file, JavaScript file and how to associate them.
- Using functions for timing and compartmentalization of code
- Conditional logic for dynamic decision-making
- Loops for repetition and control of code

LEARNING ACTIVITIES

- Screen casts
    - There are many instructor-guided activities using screen casts that walk you through the concepts and techniques covered in every unit of the course.
- Instructional Activities/ Worksheets
    - These are designed to be practical exercises to use concepts and techniques covered in lecture and follow along videos.
    - Students will also keep a log of errors encountered, their causes and solutions as part of learning to troubleshoot code in the class.
- Assignments
    - These are progressive activities or projects where students apply a combination of concepts and techniques learned during lecture. Any art or graphics used in these assignments are provided for you by the instructor or must be created by the student.
- Discussions
    - Debugging Together – Students will post errors discovered during the course of their activities
    - Scratch – Students will research MIT's learning tool and implement it for themselves.
- Reflection Videos
    - Students will share their experiences in the class with the instructor and fellow classmates. These videos should include an analysis of the work completed for the respective week.

Grade Weights

Assignments 80%

Reflection Videos 10%

GPS/Discussion 10%

Total 100%

Many assignments will have a craftsmanship component to assignments grades, not exceeding 25% in weight. Requirements for good craftsmanship include (but are not limited to) following industry conventions discussed in class, good commenting and formatting of code for readability and reusability.

Strategies for Successful Learning

- Keep a well-organized set of files with all activities and demonstrations so you can refer back to them as you would with your notes.
- Students are encouraged to ask questions during throughout the course. There is no such thing as a "stupid question."
- Memorizing code is not the goal of this class, but you should be familiar with its concepts and comfortable using documentation to look up the various components as you need to use them.
- If you find you are having trouble with any part of the class material do not hesitate to get help and get help early!