COURSE SYLLABUS

PROGRAMMING FOR WEB APPLICATIONS I

FULL SAIL UNIVERSITY

CONTACT INFORMATION

Catalog Course Code: WDD 244

Three-Letter Course Abbreviation: PWA1

Instructor: Fialishia O'Loughlin

Telephone: 786.505.2904 or 407.679.0100 ext. 3276

Email: foloughlin@fullsail.com

iChat: foloughlin@fullsail.com (AIM)

Hours: Campus Office hours: Immediately after lectures
Online Office hours (EST):
AIM before 5:00 p.m.
Email after 5:00 p.m.
Phone before 12:00 p.m.
Lab assistance: Contact the Lab Specialist or use the FSO portal to ask general questions

COURSE DESCRIPTION

The Programming for Web Applications I Course trains students in the technologies used to create dynamic content for the web using client-side programming. This course builds upon the coding and logic concepts learned in the Web Programming Fundamentals course, continuing the use of JavaScript. Students will also be shown more advanced concepts, such as data structures and key algorithms.

COURSE MATERIALS

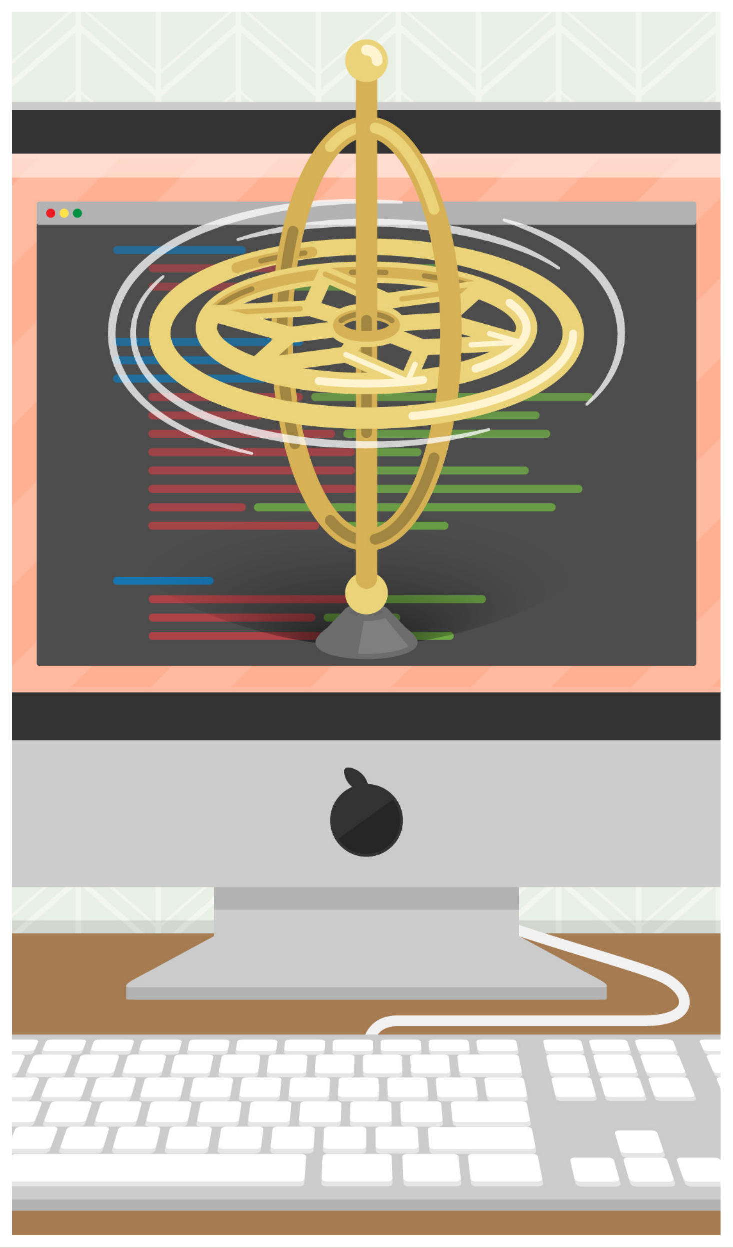
- Set up a domain and web-hosting service by Week One
- Install file transfer protocol (FTP) software
- Purchase a flash drive or recordable media by Week One
- Laptop with approved software
- Install *AIM* and set up an account by Week One
- Set up a new *GitHub* account by Week One
- Install *WebStorm* or a favorite code editor by Week One
- Install *SmartGit* GUI client by Week One
- Install a flowcharting tool of choice (e.g., *OmniGraffle*, *Google Drawings*, *Microsoft Visio*)
- Required textbook for every class session: *Head First JavaScript Programming: A Brain-Friendly Guide*, Eric T. Freeman and Elisabeth Robson, O'Reilly Media (ISBN 10: 144934013X; ISBN 13: 9781449340131)

  Head First JavaScript Programming supplemental resources: http://wickedlysmart.com/hfjs/
- Optional resource: *Modern JavaScript: Develop and Design*, Larry Ullman, Peachpit Press (ISBN-10: 0321812522; ISBN-13: 9780321812520)

COURSE OBJECTIVES

Students will accomplish these course objectives by completing the tasks specified beneath each one:

- Understand the ADDIO web-development process
  - Use critical-thinking and problem-solving skills to solve logic problems
  - Analyze *JavaScript* code
  - Design a flowchart of coding logic
  - Develop and debug programming issues
  - Integrate, test, and optimize a web application
- Apply and use native objects
  - Understand how the Math object is used
  - Demonstrate an understanding of the Date object
  - Use indexOf, parseInt, or other native objects to enhance web applications
- Enhance HTML forms with client-side functionality and validation
  - Debug and fix issues with an HTML form requiring the validation of user input
  - Enhance the performance of a web application with HTML form functionality
- Access and manipulate web-browser document object model (DOM) elements
  - Employ dot notation to access DOM objects and elements
  - Use *JavaScript* to define and manipulate elements from an HTML page



- Understand and implement synchronous and asynchronous events
  - Set up event handlers for page loading
  - Create event handlers for user input
- Understand constructor functions, objects, and prototypes
  - Create both a constructor function and an instance of an object from a constructor
  - Use the method call to invoke a function using the "this" keyword
  - Assess default prototype properties of a constructor function
  - Add new prototype properties to a constructor function
  - Use *JavaScript JSON* to parse a string and/or convert a string into an object

COURSE OUTCOMES

By the end of this course, students will be able to:

- Use the ADDIO web-development process to analyze and enhance web applications
- Improve critical-thinking and problem-solving skills to resolve logic problems
- Use the *JavaScript* programming language to:
  - Recognize the different types of data structures, such as objects and arrays
  - Solve problems by branching code using conditional logic
  - Create loops to iterate through elements of an array or properties of an object
  - Modify the contents of an array object via array access notation and array methods
  - Create and manipulate object literals and constructor objects
  - Control HTML (add, update, and remove HTML elements)
  - Display dynamic content
  - Work with forms and events
  - Develop closure and scope levels

GENERAL EDUCATION COMPONENT

In the real world, the ability to collaborate and work well within a team of web-application developers requires strong team-building, critical-thinking, and problem-solving skills. Students enter this course after the Psychology of Play course, in which they are given an overview of the theories and concepts of play. In addition, they are introduced to the ways in which the action of play shapes the brain, develops critical-thinking skills, and strengthens the ability to collaborate with others in social and professional settings. Because the use of the creative, technical, and analytical qualities of the brain when working within a team environment is important, further sharpening these skills will benefit students throughout the Web Design and Development program and in their careers.

DEGREE CONNECTION

Programming for Web Applications I is a continuation of the Interactive Programming courses as part of the Web Design and Development degree program. This course builds upon the critical-thinking, logical problem-solving, and coding skills/concepts learned in the Web Programming Fundamentals course as students continue to use the *JavaScript* language. This course is designed to teach students new techniques that will enhance their client-side programming skills and prepare them for the advanced programming concepts in the upcoming Programming for Web Applications II course.

INDUSTRY CONNECTION

JavaScript is an incredibly diverse language and much larger in application than what it was known for ten years ago. In addition, a large number of new APIs are now in constant development. Having *JavaScript* skills is a must for any modern web developer, and knowing *JavaScript* well is probably one of the most challenging and rewarding things a student can do to become a better web developer.

After completing this class, students are ready and qualified to pursue a career as a front-end web designer or entry-level web programmer and developer. Course portfolios will provide recruiters with a demonstrable means to assess student competencies. To this end, students will have the opportunity to set up assignments online as part of a professional portfolio.

RESEARCH COMPONENT

Information literacy is defined by the American Library Association as the ability to "recognize when information is needed and have the ability to locate, evaluate, and use effectively the needed information." Students will research advanced topics of their choice using the Full Sail Library on Connect to enhance their critical-thinking and research skills. Students in this course will be required to define the need for information required to solve problems, critically evaluate information and its sources, and adhere to copyright policies and standards for citation. Through specific assignments and activities, the instructor will determine opportunities for the development of skills in information literacy and the use of the Full Sail Connect Library.

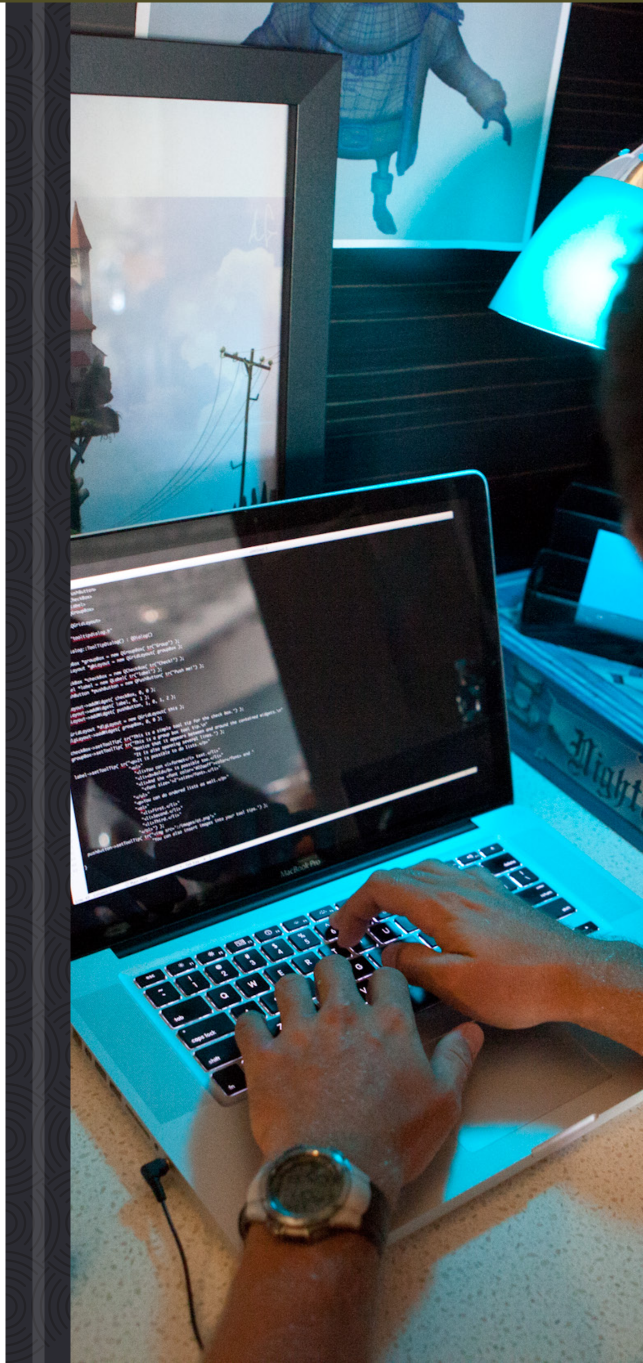
ADDITIONAL RESOURCES

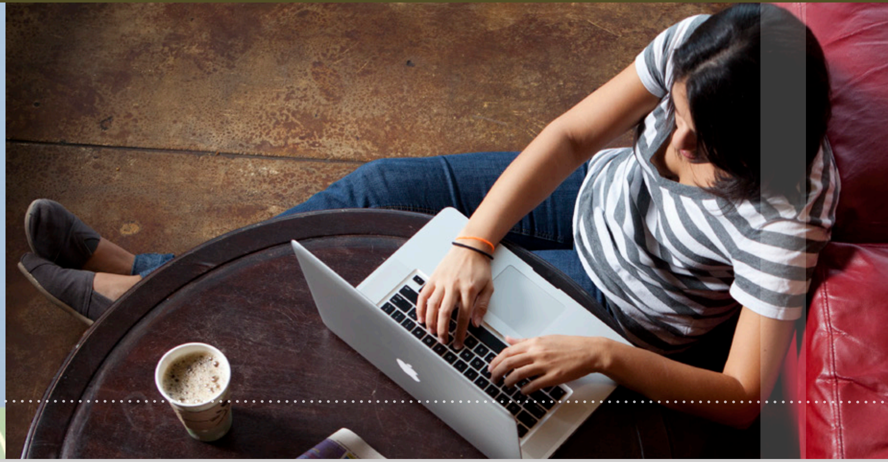
Students may use the following links for extra help:

- *Head First JavaScript Programming* resources: http://wickedlysmart.com/hfjs/
- *lynda* video tutorials: http://lynda.fullsail.edu
- *W3Schools*: http://www.w3schools.com/
- *Yahoo*: http://developer.yahoo.com/performance/
- *Mozilla*: http://developer.mozilla.org
- *Stack Overflow*: http://stackoverflow.com
- *DocHub*: http://www.dochub.io

TOPICS COVERED

- ADDIO Web-Development Process
- Critical Thinking and Problem Solving
- Native Objects
- HTML Forms with Client-Side Functionality and Validation
- Document Object Model (DOM) Elements
- Synchronous and Asynchronous Events
- Constructor Functions, Objects, and Prototypes





LEARNING ACTIVITIES

FSO Activities

Throughout the course, students will complete several online and in-class exercises relating to the various weekly topics or objectives being discussed. These activities are not graded, though students are still responsible for the materials covered and are expected to review each online activity. If they do not, they may lose GPS participation points. Most of the activities are short and direct and contain small amounts of actual code; however, some of the activities build on each other. As a result, it is imperative that students understand each activity before moving on to the next.

FSO Homework

Students will be developing a wide variety of small-scale programming projects to validate the knowledge they have gained throughout the course. Most assignments will focus on the core learning objectives taught in the previous learning material and are usually due before the end of each week (or in-class lecture or lab—if applicable). This will allow students to have some time at home to complete the assignment if necessary. Although the grade weight of the course assignments is fairly light, understanding and completing them is a very crucial part of preparing for the written and practical exams.

Group Tasks

Students will participate in at least four group presentations worth approximately 2 percent of the overall grade during the term. Although students are individually responsible for submitting their own programming work, they will develop most of their group project work as part of a team of two to four people. A handout will be provided during the first lecture for group assignments.

Group assignments will focus on the creation of team deliverables and will give students some practice in team building and developing a personal portfolio. Beginning with Lecture One, each team will set up an initial client sales pitch and presentation for the first team demo. During Lectures Two through Eleven, each team will continue to present a new and improved version of the client's project and the team's progress (using *PowerPoint*). Handouts will be distributed via the online portal explaining the team charter setup and additional guidelines. Assignment details and due dates are subject to change.

Practical Examinations and Quizzes

Programming for Web Applications I places a strong emphasis on using practical exams to assess each student's knowledge and understanding of the objectives. This is reflected in the overall grade weights of the course. As a result, students will have a choice between two types of exams: true/false assessment or a coding challenge. Weekly quizzes will also be given.

GRADE WEIGHTS

Get Organized	12%
Web App Exercises	18%
Class Activities/Quizzes	11%
Midterm Exam	12%
Team Demos	24%
Research Topic/Summary	13%
GPS/Professionalism	10%
Total	**100%**

STRATEGIES FOR SUCCESSFUL LEARNING

Please keep the following strategies in mind over the next four weeks:

- Ask questions and don't get behind. Again, the importance of doing your homework cannot be stressed enough.
- Get and stay organized! Get started on the right track by completing Steps One through Six under the Important Announcements tab on FSO to download the required files so that you can get and stay organized.
- Client satisfaction and professionalism are important! In the real world, being on time and meeting your client or manager's deadlines are important to having a successful career. Logging in to FSO is not sufficient enough to obtain a good grade. For each week of class, your grade will be determined by class attendance, completion of your *JavaScript* assignments and activities (submitted via FSO and/or *GitHub.com* when required), and participation in discussions for that week's class by a set due date.
- See FSO for a complete list of course grading rubrics, homework submission guidelines, and the late work policy.
- Schedule your study time. You cannot understand the topics presented without making time for careful and timely study sessions. If you are having difficulty with a particular topic, please contact the instructor, but first make sure you have taken the time to do your part. Dedicate and schedule some time to complete and work through all the nongraded activities so that you are prepared for the upcoming graded assignments.
- Learning through repetition and struggle is normal. Coding and troubleshooting errors will not come easy for some, so perseverance is key. You may get frustrated when error messages or bugs occur in your program. This happens to everyone—even the most experienced of coders. Stay focused and code in a modular fashion. Use good debugging techniques, fix one thing at a time, and don't give up!
- Consult three sources before approaching the instructor with an issue. Don't be afraid to ask questions, but learn to be resourceful by researching the problem first. Start by consulting the Internet (see the "Additional Resources" section), your textbook, a classmate, or the Lab Specialist before contacting the instructor. Training yourself to be resourceful will benefit you greatly in the real world. Once you've tried three other resources, feel free to send an IM or use email and FSO to send the instructor questions.
- Have fun!

COURSE-SPECIFIC RUBRICS

Please check FSO's Course Materials and References for course-specific grading rubrics and the weekly calendar of assignments. This syllabus, all grading rubrics, and weekly calendar activities may be altered (at the instructor's discretion) during the course of this term. It is the responsibility of the student to make any adjustments as announced.