CptS 489  
Web Development  
Syllabus

# *Last Updated 12 December 2019*

# Basic Facts

|  |  |  |
| --- | --- | --- |
| **UCORE  Designation:** | None |  |
| **Catalog Description** | *Web development using markup languages, style sheet language, and scripting languages; developing and consuming web services; testing web applications.* | |
| **Prerequisites:** | CptS 322 with a C or better and certified CptS, EE, CE or SE major | |
| **Instructors**: | Dr. Chris Hundhausen  [hundhaus@wsu.edu](mailto:hundhaus@wsu.edu)  SPARK 217  Office Hours: Tuesday and Thursday 12-1:00 p.m. and by appointment. Please arrive in first 15 minutes or let me know in advance if you are going to arrive late. Skype (with screen sharing) or phone meetings are welcome. I’m available to talk about web programming, jobs, careers, graduate school, or anything else. | |
| **Teaching Assistants:** | Coral Jain (grader)  Office location: TBA Office Hours: TBA and by appointment  Carla De Lira (Kliks and gradebook updates)  [carla.delira@wsu.edu](mailto:carla.delira@wsu.edu)  Office location: EME 228 Office Hours: by appointment | |
| **Meeting Time & Place:** | Tu. & Th. 9:10 – 10:25 in Spark 335 | |
| **Credits** | 3 | |
| **Online:** | <https://plus.osble.org>  OSBLE Plus is a learning management environment developed by the HELP Lab at Washington State University. An invitation will be sent to your official WSU e-mail address. Please use that invitation to create an account with your preferred e-mail address or go to plus.osble.org and choose “Sign up.” Use your WSU ID in the Student ID field. Once you have an account, you will be **automatically** added to the course as long as you are officially registered for it. | |
| **Required Texts** | * *HTML/CSS/JavaScript*. [w3schools.com](https://www.w3schools.com/) is the comprehensive online reference for this HTML/CSS/JavaScript aspects of this course. It contains easy-to-read reference materials, step-by-step tutorials and a wealth of examples. The instructor’s lecture notes contain many references to materials in the w3schools collection. * *React.js*. This course’s unit on React.js is structured around the 12 “Main concepts” in [the comprehensive documentation](https://reactjs.org/docs/getting-started.html) at [reactjs.org](https://reactjs.org/). We will also delve into the “Advanced Guides” as time allows. * *Node*.js. TBA * *MongoDB*. TBA * *AWS deployment*. TBA. | |
| **Recommended Texts** | The following is book provides a great supplement to the material on React.js:  Wieruch, Robin (2018). *The Road to Learn React*. ISBN 9781720043997. Available online for an optional donation at <https://leanpub.com/the-road-to-learn-react>..  The following two texts are recommended for your career development:   * Laakmann McDowell, Gayle. (2016). *Cracking the Coding Interview* (6th ed.). Palo Alto: CareerCup. * Sonmez, John. (2017). *The Complete Software Developer’s Career Guide.* Sanford, NC: Simple Programmer. | |
| **Required Software and Accounts** | * Web browser of your choice (I recommend [Google Chrome](https://www.google.com/chrome/)) * [Git](https://git-scm.com/) * React.js toolkit * Node.js toolkit * MongoDB Cloud Atlas (details TBA) * Integrated development environment of your choice. I recommend [Visual Studio Code](https://code.visualstudio.com/) * Account on GitHub.com * Account on Mongo Cloud Atlas * Account on Amazon Web Services * A Git client. I recommend [GitHub Desktop](https://desktop.github.com/) or just use the command line. Also note that Git is fully integrated into Visual Studio Code. * Kliks classroom response system developed by Professor Emeritus Carl Hauser. Access it at <http://kliks.eecs.wsu.edu/cs489/>. Log in using your OSBLE credentials. | |
| **Using Devices in Class** | We will use smartphones and laptops throughout the semester to facilitate class activities. However, research and student feedback clearly show that using devices for non-class related activities harms not only your own learning, but other students' learning as well. Therefore, please use devices only during activities that require them. At all other times, please refrain from using your device. | |

# Course Overview

According to recent estimates, [over four billion people use the internet worldwide](https://www.internetworldstats.com/stats.htm)—about 55% of the world population. Most of this access is facilitated by a web browser such as [Google Chrome](https://www.google.com/chrome/), [Mozilla Firefox](https://www.mozilla.org/en-US/firefox/), [Microsoft Edge](https://www.microsoft.com/en-us/windows/microsoft-edge) or [Apple Safari](https://www.apple.com/safari/). While many web pages display *static* (fixed) content, an increasing number of web pages are *dynamic*—that is, they update automatically, often in response to user interaction, live data feeds, and even changing screen sizes. Many also connect with back-end data and services to provide enhanced user experiences.

This course focuses on development of *dynamic* websites, which have come to be known as “[web apps](https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app)” (web applications). In addition to being dynamic, modern web apps are [*responsive*](https://ux.stackexchange.com/questions/50499/what-is-the-meaning-of-responsive-design)—that is, they gracefully adapt to any screen size. Well-constructed web apps running in modern web browsers [are increasingly rivaling the quality of native mobile apps](https://www.theverge.com/circuitbreaker/2018/4/11/17207964/web-apps-quality-pwa-webassembly-houdini) (for Android and iPhone), while also providing a seamless desktop experience that native mobile apps cannot provide. (If you don’t believe me, try using [Facebook](http://facebook.com) on your mobile device; the experience is quite close to what you’ll experience with Facebook’s native app.) An added advantage of web apps is that they can run on all kinds of devices (mobile devices, tablets, laptops, desktops) *without the need to maintain multiple codebase*s.[[1]](#footnote-1) This is a huge win if you need to produce a cross-platform app quickly and on a limited budget.

The [tech stack](https://blog.hubstaff.com/technology-stack/) for web development [is a jungle](https://codingthesmartway.com/the-2018-roadmap-to-fullstack-web-development/), and there is no way a 15-week course could ever hope to cover even a small fraction of it. Our journey will instead focus on the key ideas and concepts of web programming, laying a foundation that should enable you pick up a whatever web development tools are needed for the job at hand. I will teach one way of writing web apps that is popular within industry: using HTML/CSS/JavaScript, [React.js](https://reactjs.org/) and [Bootstrap](https://getbootstrap.com/) on the client side, [Node.js](https://nodejs.org/en/) on the server side, and [MongoDB](https://www.mongodb.com/) as a back-end database. You will learn web deployment using the [Elastic Beanstalk](https://aws.amazon.com/elasticbeanstalk/) framework on [Amazon Web Services](https://aws.amazon.com/). However, with a few exceptions, *you are free to use whatever languages, frameworks and tools you want to complete the individual assignments and team project in this course*. Indeed, this course is as much about *inspiration* as it is about knowledge. I will provide some entry points into web programming, but what I really hope to do is inspire you to engage in deeper explorations on your own. For example, you might want to dive into [Angular](https://angular.io/), [Ruby on Rails](https://rubyonrails.org/), [PHP](http://www.php.net/), or [ASP.NET](https://www.asp.net/), just to name a few popular web languages and frameworks. Whatever technologies you ultimately decide to adopt, the core ideas and concepts you learn in this course should provide a solid foundation.

# Learning Outcomes

After successfully completing this course, you will be able to do the following*:*

|  |  |  |
| --- | --- | --- |
| Course Learning Outcome | [ABET Performance Indicators](https://docs.google.com/document/d/1XnwjgdQdVWMxmHjgKNFzD_syHf1uGddYxUy6BNUEEQk/edit?usp=sharing) Aligned With | |
| 1. Design, implement, and evaluate *responsive* web apps to meet a given set of requirements. | | 2a, 2c, 2d, 2e, 2g |
| 1. Communicate effectively in a simulated job interview for a web developer position. | | 3a, 3b, 3c, 3f |
| 1. Function effectively as a member of a web development team. | | 5a, 5b, 5d, 5f, 5g |
| 1. Design web apps that that enable users to complete supported tasks quickly and easily.[[2]](#footnote-2) | | 6a, 6b, 6c |
| 1. Design web apps that are coded according to existing conventions and standards. | | 6a |
| 1. Use appropriate learning strategies to acquire and apply new knowledge needed to solve programming problems. | | 7a – 7g |

# Materials and Resources

A great place to find resources for learning web programming is on the web, where you will find a daunting number of online guides and tutorials to help you learn *full stack* development (i.e., development of both client- and server-side web technologies). One of the best places to find comprehensive, easy-to-read and authoritative materials on web development is [w3schools.com](https://www.w3schools.com/), which will be used extensively in this course.

In order to participate in class, you are required to use Kliks (<http://kliks.eecs.wsu.edu/cs489>), which you must access via a web browser in class in order to participate. *Note: accessing the Kliks URL must be done from a device connected to the WSU wireless network to receive credit for lecture participation. Using Kliks from outside the WSU wireless network will result in a 0 for that participation score.*

A laptop computer is essential for in-class activities; **please bring yours to each class.**

# Course Schedule

See the separate and evolving [“Schedule” document](https://emailwsu-my.sharepoint.com/:x:/g/personal/hundhaus_wsu_edu/ETXGA3HnqHNPqIgsoEygg6gBt6JCiK4AcbdktfrTGxkNPw?e=qxZwiq) (an Excel spreadsheet).

# Course Activities and Structure

The course meets twice per week for 75 minutes. Course meetings will feature interactive lectures with live coding of relevant examples, Q & A sessions, small programming tasks relevant to lecture topics, small and large group discussions, team meetings, and open lab time for completing course assignments in a supportive environment.

The **Kliks** system will be used to take attendance and will also help engage you more actively in some classroom activities. **If you don’t have access to Kliks, you will not get credit for participating in class. However, you will get three “free” attendance credits, as noted below.**

You are expected to engage with the assigned online learning materials *before* the class meetings for which the readings are assigned. This is because the materials are designed to set you up to get the most out of class activities. To nudge you to engage with the assigned materials, I may have you answer reading quiz questions using Kliks.

# Assessment

Your grade for the course will be based on the following items (weights are in parentheses):

* **Participation (20%)**. You will get the most out of this course if you attend class meetings and participate actively. Your attendance and participation will be monitored through your Kliks responses. To receive credit (all or nothing) for a given class meeting, **you will need to be present and to respond to *75%* of the Kliks questions on that day**. We understand that you may need to miss class occasionally for valid reasons. For this reason, **your three lowest class participation scores will be discarded—that is, you will receive three free attendance/participation credits**.We also understand students may occasionally forget to bring an electronic device capable of web browsing, but still show up for class. The first time this happens, you may message the TA (via OSBLE) to request that you receive credit for the class. After that, you will not receive credit if you fail to use Kliks in class.

Some **online course evaluation surveys** will be administered. These surveys will require around 10 minutes of your time, and your answers will be anonymous. Completion of each survey will count the same as attending a class (one participation credit). Your honest and thoughtful feedback in the survey will help guide and improve the course.

* **Individual Programming Milestones (45%).** This course takes a *project-based* approach to learning web programming. The first part of the course is structured around a series of 9 weekly individual assignments, some of which build upon each other. Each assignment is worth 5% of your overall grade. By the time you complete this series of assignments, you will have engaged in *full stack web development* to create a non-trivial web app.

Because each assignment may build on previous assignment, it is important to complete each milestone before moving on to the next one. If you are unable to complete a given milestone, please contact the instructor right away to discuss your options.

* **Post-Mortem Reflections and Responses** **(5%).** Part of learning to become a good programmer is developing an ability to be self-reflective of your process and progress—a skill also known as *meta-cognition*. To practice this, you are required to write a post-mortem reflection as part of each individual programming milestone. You will post your reflection, which must be at least 300 words, to an OSBLE discussion assignment. In a post of at least 100 words, you are also required to respond to another student’s reflection by identifying areas of agreement and disagreement and providing suggestions or advice.
* **Team Project (30**%). You will dedicate the final four weeks of the course to a team project. For this project, you will partner with one or two other students to develop a web app of your choosing. Prior to beginning the project, you will be required to submit a one-page description of the app you would like to develop, including its basic functional requirements, for instructor approval. During each of the final four weeks of the semester, your instructor will assign each member of your team a “team milestone” (TM) grade based on inspection of your source code repository, team Kanban board and a demo app that showcases what you accomplished during the week. During the final exam period, each team will have the opportunity to demo their app in front of the whole class. Your instructor will use these final demos to help determine each team’s fourth (and final) team milestone grade. The weight of each team milestone (TM) will progressively increase as follows: TM1: 4%; TM2: 6%, TM3: 8%, TM4: 12%.

Extra Credit Opportunities

In addition to participating in a code interview, there *may* be other opportunities to earn extra credit. Any such opportunities will be announced in class, the OSBLE activity feed and/or written into assignment prompts.

# Grading Scale

The following scale will be used to convert your course percentage into a grade.

|  |  |
| --- | --- |
| 90-100 | A |
| 88-89.99 | A- |
| 85-87.99 | B+ |
| 80-84.99 | B |
| 78-80.99 | B- |
| 75-77.99 | C+ |
| 70-74.99 | C |
| 68-70.99 | C- |
| 65-67.99 | D+ |
| 60-64.99 | D |
| <60 | F |

# Suggestions for Getting the Most out of This Course

* **Participate actively in class.** You will get the most out of this class if you show up! Much of your course grade depends, at least in part, on your participation and attendance in class. Therefore, you would be well-served to attend every class session (except for three “freebie” attendance credits), and to participate actively.
* **Learning is up to you.** Through nearly 20 years of teaching, I have learned that some students may bring unhelpful expectations to a college course. Learning does not come "for free"; it is not simply a matter of "being taught"—of information being miraculously transferred into your brain!

This is especially true in a project-based programming course such as this one. I believe you need to *actively engage* in web programming—and to figure out how to solve your own problems—in order to become a good web programmer. Thus, I see my role in this course as three-fold: (a) to set up a series of strategically-chosen assignments that will that will help you to learn web programming by doing web programming; (b) to point you toward online resources, and in some cases provide guidance and examples, that will help you complete these milestones (and thereby learn web programming), and (c) to motivate and inspire you to actively engage in web programming and ultimately determine if it is for you. If you expect something else from me or if you have suggestions for how I can help you, please talk to me individually. I realize that there is no “one size fits all” approach to learning, and that some students may benefit from other kinds of assistance.

* **Spend your time wisely.** To succeed in this course, you must take the time to complete all programming milestones. Since students come to the course with different levels of experience and interest and learn at different paces, I cannot predict how long completing these milestones will take. (I’m hoping you will tell me so that I can make course adjustments as needed!) Here are some tips for optimizing the time you spend on assignments:
  + Carefully read the milestone instructions before starting it. Ask questions if you are unclear on what is expected.
  + If you get stuck, get help! In studies of new hires in the software industry, one of the biggest blunders new hires make is that they spend inordinate amounts of time struggling on a problem without asking for help. There are many ways to get help, including (a) googling your problem or searching online Q&A sites such as Stack Overflow; (b) using OSBLE’s activity feed to ask for help online; (c) contacting trusted students in the course for help; or (d) contacting the instructor or TA for help.

# Policies

Please familiarize yourself with the following course policies. By following them, you will get the most out of this course, and you will not encounter any unwelcome surprises down the road.

* **Add a profile picture to OSBLE**. Uploading a recent picture of yourself to OSBLE will help the instructors, the TA, and other students in the class to associate your name with your face. We would greatly appreciate it if you would do this, as it will help us to learn your name more quickly.
* **Corresponding with the instructor and TA through OSBLE**. Please e-mail the instructor and TA only through OSBLE; *do not e-mail us directly, except in an emergency*. If you think your question is of general interest to the class, consider posting it to the course activity feed in OSBLE. We try to respond to student questions quickly—sometimes in minutes and usually within hours.
* **Accessing course materials**. OSBLE is the online presence for this course. Log in regularly (**every day**) to view course announcements, view the course calendar and schedule, participate in the course feed, access course materials, access your grades, and submit assignments. Note that you can forward your personal notifications, OSBLE e-mail, and course feed items directly to your e-mail. To do so, adjust your "Personal Settings” in OSBLE (the “gear” icon in the upper right).
* **Checking your grades**. To view your current grades, click on the Grades tab in OSBLE. The course TA will update your grades weekly. Please check your grades regularly to ensure that your grades have been entered properly, and please let the TA know as soon as possible if you detect an error.
* **Challenging a grade**. If you believe that the TA has made a grading error, you have *one week* from the time your grade is first posted to the gradebook to bring the matter to the TA’s attention. Please discuss grading issues as soon as possible. Students have often attempted to bargain for points well after their grades have been posted—often at the end of the semester when they have realized that they needed more points to obtain a certain grade. Please do not attempt to do this! The first time you try to do this, I will gently remind you that I do not engage in this practice. Any subsequent attempts to bargain with me will be ignored.
* **Late policy for assignments**. The late policies for individual milestones are clearly explained on the assignment prompts. Be sure you understand each assignment’s late policy before it is due.
* **Academic integrity: Kliks**. 20% Of your course grade is based upon your use of Kliks in class. We will deal harshly with students whom we catch using each other's Kliks access. **Any student caught using Kliks in another student's place will receive an automatic "F" in the class. Likewise, any student who knowingly allows another student to use Kliks in his or her place will receive an automatic "F" for the class**.
* **Academic integrity: Code Plagiarism**. All code that you or your group submits must be solely your own or that of your group. If you use or adapt code from outside (online) sources, you must cite the source of the code in an inline comment (e.g., //Copied from <http://my.source.com/code.html>).

While you are encouraged to discuss individual milestones with others in the class, the code that you hand in must ultimately be written by you. To understand what kind of peer collaboration does and does not constitute code plagiarism (a.k.a. “cheating), please consider the following sample scenarios:

1. You and some friends are working together on the next assignment. You just finished the assignment and head off to work. A friend who isn't finished asks to get a copy of the assignment from you so that he can use it as a guide. Wanting to be a good friend, you email him a copy of your assignment with the promise that he won't copy it. ***This is cheating***. You are not allowed to share copies (analog or digital) of your work with other students. Your friend may end up copying and pasting some of your code into his solution. This will likely be detected by plagiarism detection software. ***This is a very common way to get in trouble. Don’t do it!***
2. You and some friends are getting together after class to discuss the assignment due tomorrow. You're done and want to help your friends out. To do so, you discuss the assignment using high-level descriptions (e.g., "I have class X that does Y, which helps me get task Z done") and even draw some nice pictures and/or flowcharts for them to follow. ***This*** ***is not*** cheating and is encouraged.
3. For one of the assignments, you're required to implement a poker game. A quick Google search reveals several existing implementations. You download the project and modify it to suit the requirements of the assignment. However, you do not attribute the source of the code with an inline comment. ***This is cheating.***
4. Your friend is almost done with his homework assignment but has run into a nasty bug. You look at his program and quickly realize that he isn't correctly deleting a dynamically created pointer. You show him how to fix the bug. ***This is not*** cheating and is encouraged!
5. A student has asked for code help on the OSBLE activity feed. You respond with a partial code snippet. ***This is not*** cheating and is encouraged!
6. You and a friend are working together on an assignment using the same computer. As a result, both of you turn in the same or nearly identical assignments. As I cannot determine who did the work, ***this is cheating***. Remember, you must turn in original work typed out entirely by yourself.
7. A friend of yours has taken the class in a prior semester. You ask him for his prior code solutions, which you use to aid you in completing your assignments. ***This is cheating***.

If you find yourself in a situation that is not listed above and are worried about whether it might be considered cheating, err on the side of caution and don't do it. Instead, email me. I'll give you a definitive answer and add it to the list above.

Code Plagiarism rules will be strictly enforced. On the first violation, you will receive a "0" on the assignment and verbal reprimand. On the second violation, you will receive a grade of “F” for the course and be referred to the Director of the School of EECS and/or Dean for possible academic sanctions.

* **Students with Disabilities**. We are committed to providing assistance to help you be successful in this course. Reasonable accommodations are available for students with a documented disability. Please visit the Disability Resource Center (DRC) during the first two weeks of the semester to seek information or to qualify for accommodations. All accommodations MUST be approved through the DRC (Admin Annex Bldg, Rooms 205). Call 509-335-3417 to make an appointment with a disability counselor.
* **WSU Safety Statement**. Washington State University is committed to maintaining a safe environment for its faculty, staff, and students. Safety is the responsibility of every member of the campus community and individuals should know the appropriate actions to take when an emergency arises. In support of our commitment to the safety of the campus community the University has developed a Campus Safety Plan, [http://safetyplan.wsu.edu](http://safetyplan.wsu.edu/). It is highly recommended that you visit this web site as well as the University emergency management web site at <http://oem.wsu.edu/> to become familiar with the information provided.

1. While frameworks do exist for maintaining native Android and iPhone apps with a single codebase (e.g., [Xamarin](https://visualstudio.microsoft.com/xamarin/), [React Native](https://facebook.github.io/react-native/)), using these frameworks [is not without its issues](https://www.uruit.com/blog/xamarin-vs-react-native/), and often additional native code needs to be written for each device. [↑](#footnote-ref-1)
2. I have a background in user interface design and teach CptS 443 (“Human-Computer Interaction”). While this course does not explicitly focus on theories, principles and processes for designing humanly-usable web apps, this course will emphasize *good design*. To sharpen your design skills, you are encouraged to take CptS 443, which complements this course. [↑](#footnote-ref-2)