

CSCI 445 – Web Programming

Fall 2020

Syllabus

Instructor:

Dr. Cyndi Rader

Email: crader@mines.edu

Office Hours (Zoom <https://mines.zoom.us/j/7808271120>):
Monday 4-5, or by appointment

Teaching Assistant:

TBD

Website:

<https://cs-courses.mines.edu/csci445/index.html>

Learning Outcomes:

By the end of the semester, students will be able to:

- Design and develop a modern website using HTML5 + CSS3 while applying software engineering principles.
- Discuss and apply proper design methodologies to create visually appealing and accessible websites.
- Describe the basic structure of the Internet and World Wide Web. Discuss how a URL determines what webpage is returned to a browser from this structure.
- Explain and compare the differences between (1) client-side and server-side processing (2) front-end development and back-end development.
- Create a responsive, dynamic, and interactive website using JavaScript.
- Create a responsive, dynamic, and database backed website using PHP.
- Create a responsive, dynamic, and interactive website using AJAX.
- Locate information and construct a solution to learn new technologies/languages/tools/libraries on your own.

Individual Assignments:

- There will be approximately 12 **individual** programming assignments throughout the semester. Due dates will be specified and are firm. Individual programming assignments **must** represent individual student work.
- All programs must be submitted in electronic form through Canvas.
- Your goal should be to make it easy for us to see how great you did. If we make a couple of good faith attempts but fail to get your program running, we will try once to contact you to help us. If we still cannot get your work to execute, it will receive a zero grade.

Team Project:

- There will be one course project, to be completed in teams of three.
- All code and files required for the projects must be submitted to Canvas.

Exams and Quizzes:

- There will be two exams during the semester and approximately 8 quizzes.
- All exams and quizzes must represent your own work. Unlike assignments, these must not be discussed at all with your classmates.
- Make-up exams will be allowed only in accordance with University policy. Make-up exams due to legitimate travel require advance notice and instructor's approval.

Late Policy:

- All assignments, projects, quizzes, and exams are due at the date and time specified on the item handout.
- Items turned in late will receive the following deductions:
 - 1 day late: -20%
 - 2 days late: -40%
- NO work may be submitted more than 2 days, except for extreme circumstances (must email instructor to explain/receive approval)
- Items not turned in or received more than 48 hours past the due date will receive a grade of zero.

Grading:

The final course grade will be computed from the following course percentage breakdown:

- 60% Homework Assignments
- 10% Team Projects
- 10% Module Quizzes
- 10% Exams
- 5% Module Surveys
- 5% Participation (piazza, etc.)

Final grades will be determined using a *straight scale*. The straight scale assigns letter grades as follows:

- | | | | | | |
|-------------|----|----|----------|----|---|
| • [93, 100] | -- | A | [63, 70] | -- | D |
| • [90, 93] | -- | A- | [0, 63) | -- | F |
| • [87, 90] | -- | B+ | | | |
| • [83, 87] | -- | B | | | |
| • [80, 83] | -- | B- | | | |
| • [77, 80] | -- | C+ | | | |
| • [73, 77] | -- | C | | | |
| • [70, 73] | -- | C- | | | |

Discrepancies:

- If you have any questions regarding how any assignment, project, or exam is graded and you think you deserve more points than you received, you must contact the grader within **one week** of the day the item was returned to you. No claims, justifiable or not, will be considered after this deadline. If you and the grader cannot reach a consensus, then contact the instructor.
- Any assignment returned to the instructor is subject to total re-grading which may result in a grader higher or lower than your original grade.

Participation:

- A portion of the student's grade will be comprised of online participation.
- Students are expected to participate by making regular forum posts, either asking a question or

responding to an existing topic. From time to time, there may be specific discussion topics.

Getting Help

Students are encouraged to ask for help on assignments and projects as needed, but please follow these guidelines:

- You must first try to post your question on piazza. Why?
 - If you're struggling with an issue, or need clarification about a requirement, chances are other students will as well.
 - By posting on piazza, other students get a chance to provide assistance (really good skill to practice, as it can increase your reputation as a coder).
 - Students may answer the question more quickly than the instructor, and may even have *better answers* (e.g., they may have encountered and solved a problem the instructor didn't experience).
 - Before asking, you should see if others have already posted the same issue. You've probably seen some snarky comments on stack overflow etc. when posters don't do their legwork first. We will not post snarky comments!! But it's still good to get in the habit of looking for answers before posting.
 - In order to make a piazza post, you need to think about how to ask the question. Sometimes just this process will be enough for you to figure out your issue (similar to explaining a bug to a stuffed animal; this has been shown to be an effective debugging technique).
- If you've tried piazza with no success, feel free to email the instructor. I typically provide quite a bit of email support when the course is programming intensive (like CSCI445). Some guidelines:
 - Be as concise and precise as possible when describing the issue.
 - Most often you'll need to zip/send your code. Do NOT just include a really small code snippet in your message. Unless it's a really simple syntax error, a small code snippet is rarely enough to figure out the issue.
 - When sending your code, *leave in all debugging output*. This shows that you attempted to solve the problem on your own, and may be helpful for the instructor as well.
 - Before sending a large program, try to replicate the problem in the smallest possible context. For example, if pressing a button doesn't call a JavaScript function, create the smallest program (maybe copying from online examples) that does. As part of this process it's likely you'll figure out the issue.

Piazza

- Be polite. This also applies to assignment clarifications (e.g. writing "This requirement makes no sense" may not be the best phrasing. Try something like: "I'm not clear what requirement X means. Should I do [x] or [y]?")
- A Piazza post is not a text message; use complete sentences and correct spelling, punctuation, and grammar.
- When asking a question, do not post large blocks of code. A single line of code, to clarify your question, may be appropriate. Before posting, ask yourself: would this be giving most of the answer to another student? Thinking about how to phrase the question may help you solve the problem.
- When answering a question, do not post the exact code from your homework solution. Possible exception would be something that takes one line and is primarily a syntax question. E.g., to a question like "How do I set the color of my rectangle" you might answer with something like "You need to set the color before drawing. If g is a Graphics object, you can do `g.setColor(Color.CYAN);`".
- Using pseudocode is an **excellent** way to answer questions.

Collaboration Policy for Programming Projects in CS Courses:

The following policy exists for all CS courses in the CS department. This policy is a minimum standard; your instructor may decide to augment this policy.

- If the project is an individual effort project, you are not allowed to give code you have developed to another student or use code provided by another student. If the project is a group project, you are only allowed to share code with your group members.
- You are encouraged to discuss programming projects with other students in the class, as long as the following rules are followed:
 - You view another student's code only for the purpose of offering/receiving debugging assistance. Students can only give advice on what problems to look for; they cannot debug your code for you. All changes to your code must be made by you.
 - Your discussion is subject to the empty hands policy, which means you leave the discussion without any record [electronic, mechanical, or otherwise] of the discussion.
- Any material from any outside source such as books, projects, and in particular, from the Web, should be properly referenced and should only be used if specifically allowed for the assignment.
- To prevent unintended sharing, any code stored in a hosted repository (e.g. on github) must be private. For group projects, your team members may, of course, be collaborators.
- If you are aware of students violating this policy, you are encouraged to inform the professor of the course. Violating this policy will be treated as an academic misconduct for all students involved. See the Student Handbook for details on academic dishonesty.

Academic Code of Honor:

- All students are expected to follow the University's Academic Code of Honor.
- A student or assigned team working on a program may discuss high-level ideas with other students or teams. However, at time of submission all work submitted **must be his/her/their own work**.
- Use of the Internet as a reference is allowed but directly copying code or other information is **cheating**. It is cheating to copy, allow another person to copy, all or part of an exam or a project, or to fake program output. It is also a violation of the Code of Honor to observe and then fail to report academic dishonesty. *You* are responsible for the security of your own work.
- We will provide, as part of the course, functional code examples for most of the topics covered. While you are encouraged to examine these examples, your submissions must represent a good-faith effort to complete the assignment. Merely copying and pasting code from the examples will result in a failing grade. Furthermore, relying too heavily on the given examples will fail to prepare you for the much more open-ended midterm and final projects.

Disability Accommodations:

The following website <http://disabilities.mines.edu> outlines Mines's disability services. Any student requiring accommodations contact the instructor via email within the first two weeks of class to coordinate accommodations.

Discrimination and Harassment:

This course and all learning opportunities at Mines require a safe environment for everyone to be productive, develop professional practices, and to be able to share and learn without fear of discrimination or harassment. Discrimination or harassment of any type will not be tolerated. Sometimes harassment is unintentional, but regardless of intent the instructor will address any language or behaviors that might discriminate, stereotype, or promote harassment. If you witness discrimination or harassment of others, please bring it to the attention of Mines faculty so it can be addressed immediately.