Cpt S 489 – Web Development

Class Overview & Policies

Professor: Evan Olds

Email: evanolds@wsu.edu

Course Website (Blackboard): https://learn.wsu.edu/

What this course is/isn't

Is	Isn't
Technical side of web development	Artistic side of web development
Programming/development/projects, problem-solving/critical thinking, data structures/algorithms	Memorize-and-repeat (although there is some of that for the fundamental new concepts and terminology)
Computer-science approach to JavaScript	Copy-and-paste approach to JavaScript
A class where you will do challenging work to learn new concepts and gain new skills	A class where you get a passing grade just by putting in the effort (you will need to demonstrate the relevant skills!)

Materials

- Two **required** books, one of which is 100% digital and not on the shelves in the bookstore
  1. “JavaScript: The Definitive Guide”
  2. zyBooks – “Web Programming”
     - Instructions on how to purchases come later in this set of notes

The “Constitution”

- The **Syllabus** contains a list of course policies that you must familiarize yourself with
  - Consider this set of lecture notes a part of the syllabus, although note that they are two separate documents
- Your first homework assignment is to read and understand everything stated in the syllabus

Point Breakdown

- 10% from zyBooks participation activities
- 50% from homework/projects
- 40% from exams
- Unlike some of the other classes I teach, the number of points from exams and number of points from homework/projects will not be equal
  - 1 exam point != 1 homework/project point
  - Compute percent score in the range [0,100] with the following formula:
    - (obtained_HW_points / possible_HW_points + obtained_exam_points / possible_exam_points) * 50
- Grading scale that maps a percent score to a letter grade is in the syllabus and also in this set of lecture notes, after a discussion of homework/project grading

Homework/Project Grading

- Each piece of a project will be worth 3 points
- Without achieving a considerable chunk of proper functionality, the project gets a 0
- Why and what does this mean?
  - This is 400-level computer science. You must be able to actually write code that does something useful at this point. If you “tried hard” on a project but the end result is code that doesn’t work for any relevant use cases, then this is an indication of not having the ability to do the project. Period.
  - You need to be producing code that would have value in the real world

Homework/Project Grading

- Furthermore, no half-points for the pieces of code. An integer score in the range [0,3] will be assigned.
  - 0 means it doesn't work well enough to be considered valuable in the real-world and therefore you have not demonstrated an ability appropriate for 400-level computer science (as previously discussed)
  - 1 means you have something that works as it should for a lot of general cases, but you only addressed the bare-minimum
  - 2 means you fulfilled the assignment requirements to the extent where you're demonstrating understanding of all parts of the assignment, but you did not satisfy the most difficult "challenge cases"
  - 3 means all required aspects of the assignment were achieved, including the "challenge cases"

Homework/Project Grading

- Example: Sudoku Solver
- This will be an assignment where you write a Sudoku solver in JavaScript
- There will be assignments given prior to this project that may help you, so you can gain points for those separately, but the actual Sudoku Solver project will be worth 3 points

Homework/Project Grading

- Example: Sudoku Solver
- You would be given a score of 1 if your solver solves ALL easy puzzles correctly, but not the more challenging ones. This means you wrote working code that shows that you could at least implement the logic to solve easy puzzles.
- You would not get a 1 if it “narrows-down” easy puzzles and then crashes. That’s not working code that appropriate for senior-level computer science. These puzzles are simple in nature and everyone should be able to write code that works to solve the easy puzzles by the time we get to this project later in the semester.

Homework/Project Grading

- Example: Sudoku Solver
- Scores:
  - 1 point scenario described on previous slide
  - 2 points if your solver solves all easy, moderate and hard puzzles
  - 3 points if your solver solves all puzzles including the “brutal” ones
    - This is quite a bit more difficult than the other two scenarios and you’ll have to figure out some of the more advanced Sudoku strategies to achieve it
  - 0 points for all other scenarios including code that crashes, can’t solve easy puzzles, does a brute-force instead of logic-based approach, etc.

Grading Scale

- Since implementing the “bare-minimum” on a project would get you 1 of 3 points and a grade of 33%, the grading scale is somewhat adjusted to address this
- You will have to have several projects implemented correctly with MORE than the just the “bare-minimum” to pass
  - Hence “bare-minimum” in quotes. It’s a minimum relative to the functionality of the assignment, not relative to a passing grade.
- Grading scale follows on next slide

Grading Scale

Percent Score Range	Letter Grade
[90,100]	A
[85,90)	A-
[79,85)	B+
[73,79)	B
[67,73)	B-
[60,67)	C+
[50,60)	C
[47,50)	C-
[44,47)	D+
[41,44)	D
[0,41)	F

Cutoff point for passing is 50%!

Grading Scale

- This is a very lenient grading scale, considering you can get only half the points in the class and still pass
- For that reason
  - NO CURVE
  - NO “rounding” or bumping-up to next category
  - You get the grade you earn at the end of the semester with absolutely zero adjustments to your score or the grading scale
  - Monitor your progress throughout the semester!

Grading Scale

- Keep in mind:
  - Coding projects are assigned an integer score in the range [0,3]. If you get a score of 1 on all homework assignments that's a 33% on homework. You need to be aiming for a 2 out of 3 or higher on most assignments to secure a passing grade.
  - The ranges for letter grades are not the same for the A's vs B's vs C's. Some more leniency on the lower end, to reflect the reality that some lesser-skilled coders still have a place in the industry. To earn an A you need to show that you can do pretty-much all of the tasks at hand, including the ones that involve working out challenging problems and not just looking for a pre-written answer on stackoverflow.com.

Reminder

- You must read the syllabus and understand things within it
- This set of notes only covers a subset of the course policies

zyBooks?

- Online interactive textbook
- Far superior to non-interactive textbooks
- Instructions to purchase the book ($48):
  1. Sign up at zyBooks.com
  2. Enter zyBook code WSUCPTS489OldsSpring2017
  3. Click Subscribe
- You must have your zyBook purchased by the end of the first week of class

Current Course Plan

- Calendar in syllabus – pay close attention to homework due dates and exam dates
- Every Friday bring your laptops and we will implement a portion of a problem solution in class, with the following goals:
  - Get in-class participation to collectively brainstorm how to solve a particular problem and implement the code
  - Build a piece of code that may be of assistance for upcoming projects