Fundamentals of Web Applications
COMP 2406A – Fall 2020

Fundamentals of Web Applications

Dave McKenney
david.mckenney@carleton.ca

Welcome

Welcome Back!

(sort of…)

A bit about me…

Dave McKenney

david.mckenney@carleton.ca

BCS, MCS, PhD all from Carleton

A bit about the course…

Lab Coordinator: Sean Benjamin

seanbenjamin@cmail.carleton.ca

A bit about the course…

Introduction to Internet application development; emphasis on computer science fundamentals of technologies underlying web applications. Topics include: scripting and functional languages, language-based virtual machines, database query languages, remote procedure calls over the Internet, and performance and security concerns in modern distributed applications.

A bit about the course…

The course covers the principles involved in the design and implementation of web-based applications.

Focus is on understanding the fundamental CS concepts underlying web applications

Our primary programming language in this offering will be Javascript (on both client and server side).

A bit about the course…

A significant part of this course deals with organizing data and creating protocols to allow that data to be accessed and manipulated

We will also talk a lot about system scalability and robustness, which are important concepts within the web domain

A bit about the course…

Our aim is for the course to be as OS agnostic as possible so that you can choose your OS: Windows, Mac OS, or Linux. The technologies in this course are intended to work on all those platforms, though slight variations may occur.

A bit about the course…

Some topics we may cover:

Web concepts, HTTP, HTTPS

Markup languages (HTML, CSS, Bootstrap)

Javascript

Functional programming and closures

Synchronous vs asynchronous function calls.

Javascript execution environments: Browser, Node.js

A bit about the course…

More Topics:

Client- and Server-side coding (in Javascript)

Node.js and NPM

Client-Server data exchange with JSON and/or XML

JSON and/or relational databases (MongoDB/SQLite)

RESTful web API's

Server-side templating (using PUG, EJS, etc.)

Session storage and cookies, AJAX, web sockets

A bit about the course…

https://cdn2.hubspot.net/hubfs/5178372/fh%20hydrant.jpg

A bit about the course…

https://giphy.com/gifs/lil-wayne-make-it-rain-fat-joe-l41lFvtuqmey9QTAY

A bit about the scheduled lectures…

The course design this term is a bit experimental

Pre-recorded lectures from last year used for general content delivery

Scheduled lectures are supplementary and will include lots of discussion, question/answer, example problems, etc.

My intention is to record the lecture sessions too

A bit about the scheduled lectures…

Each week I will share a list of things you should complete (recordings, readings, problems, etc.) and an outline of what we will do in each lecture

I will also recap what you should have done in the last week

A bit about the scheduled lectures…

I will post in the current-events channel on Discord before starting each lecture with instructions to join

For now, we will use Zoom for the scheduled lectures

A bit about the scheduled tutorials…

Scheduled tutorial times will also be used for supplementary material and extra help

Three TAs will be available during each time slot

We will offer a mix of workshops, group discussion, and one-on-one assistance

(starting next week)

A bit about the course office hours…

Office hours will also start next week

Office hours will provide one-on-one discussion

Office hour times are posted on cuLearn/Discord

A bit about the course office hours…

TAs are there to discuss concepts, critique your design choices, and clarify misunderstandings

TAs are not there to debug your code – you need to understand your own code

Come with meaningful questions

A bit about the course grades…

Breakdown of course grades:

Project check-ins – 3 x 10%

Project – 50%

Participation – 20%

A bit about the course project...

Most of your grade will come from the project

Term-long effort that covers all concepts in the course

There will be three project options – specifications and other details will be available later (most likely next week)

A bit about the course project…

Projects can be completed individually or in pairs
(pairs can be split across both sections)

You will be able to re-evaluate your decision after the first project check-in

Pairs will have higher expectations and/or additional requirements
(communicated when the projects are posted)

A bit about the course project…

Project implementation is only one part

There is also a project report, an oral defense, and peer evaluation

A bit about the course project…

Do not share any of your project code with others, do not copy any code from any source (students, lectures, TAs, online, etc.)

Do not reuse any code you shared with others (e.g., from practice problems)

A bit about the course project...

Discuss at a high level.

Things like this are acceptable:

Clarification of the document/requirements
Overall design decisions
Useful tools/modules
Helpful readings or related practice problems
Visual demonstrations
Possible feature additions

A bit about participation…

The rest of your grade will come from participation in course activities and contributions to learning material

There are several reasons why we are including this

A bit about participation…

1) Asynchronous online nature of course
2) Web development is extraordinarily broad
3) Web development changes rapidly
4) It gives you flexibility to learn how you want
5) It gives you freedom to learn more
6) It takes advantage of our numbers
7) It encourages some key components of academia
8) It improves the course overall

A bit about participation…

Details of participation grades will be shared soon

Some planned methods of participation:

Lecture notes and summaries
Written tutorials
Practice problem tutorials
Practice problem creation
Peer study sessions
Lecture and workshop participation
Workshop creation/hosting
CWDC participation

A bit about participation…

As with all experiments, things may not go perfectly

Many things are still uncertain

We will adapt as necessary throughout the term

Please provide feedback as we go (good and bad)

Academic Integrity

There is absolutely no collaboration allowed on the projects (other than with your partner, if applicable)

Collaboration is encouraged for tutorials, practice problems, workshops, review sessions, etc.

Electronic tools are in use to detect plagiarism

Academic Integrity

New minimum penalties for plagiarism:

First offence, first-year students (< 4.0 cr): Final grade reduction of one full grade (e.g., A- becomes a B-), and additionally no credit for the assignment

First offence (everyone else): F in the course

Academic Integrity

New minimum penalties for plagiarism:

Second offence: One-year suspension from program

Third offence: Expulsion from the University

Note: these are minimum penalties. More severe penalties will be applied in cases of egregious offences

Textbooks and Other Resources

No required textbook

Some suggested textbooks on cuLearn

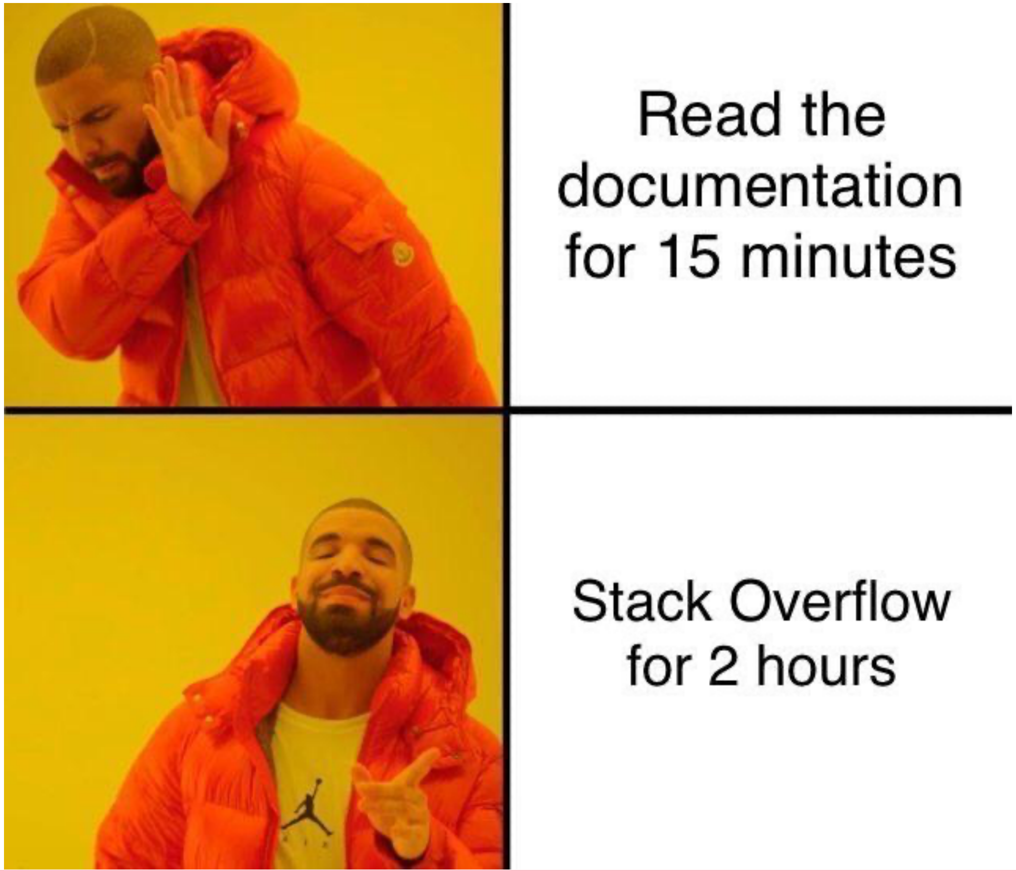
Also have free zyBook – instructions to get it are on cuLearn

Textbooks and Other Resources

A LOT of online resources and documentation

Textbooks and Other Resources

A LOT of online resources and documentation


Read the documentation for 15 minutes
Stack Overflow for 2 hours

Textbooks and Other Resources

A LOT of online resources and documentation

https://www.w3schools.com/

https://nodejs.org/en/

Some will be posted by me

Share anything useful you find!

Course Software

The course will use a variety of software

Goal is to be OS agnostic

Client-side: web browsers (Chrome, etc.)

Server-side: Node.js, NPM, MongoDB

Development: Whatever you want

Version numbers and download links are provided under Software section in cuLearn

Discord

Most of my communication will happen through the course Discord server

Office hours and workshops will also occur through Discord

See cuLearn announcements for instructions to join

Discord

Discord rules:

1. Be respectful of others.
2. This is an academic server. Try to keep conversations generally on topic. An off-topic channel is provided for general chat. Be professional.
3. Do not DM or tag TAs outside of their scheduled hours.
4. Don't share code for projects or anything that will be formally graded. Sharing practice problem code and code for other ungraded activities is encouraged.

Discord

Discord rules:

5. If your question is not of a personal nature, use one of the public channels.
6. Grades and other personal information should not be shared on Discord. All grade related queries should be handled through Carleton email or cuLearn.
7. Both sections of 2406 are sharing the same server but are being taught by separate instructors. You should ensure you clarify any questions about section-specific policies with your instructor.

Email

If you have a personal question (medical issue, etc.), email me with COMP 2406A in the subject line

Some Tips

Work on your time management skills

This is probably the most important thing in university

Lay out your schedule, set goals, meet them

Some Tips

Stop guessing while writing code.

Programming is logic-based, there should be no guessing

You can grind your way through first year with trial and error, but it gets increasingly harder later

Some Tips

Focus on learning the concepts and developing your skills

Don’t get caught up trying to get the ‘right’ answer, there are many ways to solve the same problem

Once you have something working, discuss it with somebody else and improve on it

Some Tips

Read the documentation!

This will clarify what different modules/methods do, what data they expect, what data they produce, etc.

Being able to learn this way is incredibly useful

Questions?

Any questions?

Getting Started!

This week's goals (before next Tuesday):

Head to https://www.w3schools.com/

Work through HTML/CSS/Javascript tutorials there

Make a few static web pages, add some Javascript

Watch the "Intro to Javascript" lectures on cuLearn