

mount royal university
department of mathematics and computing

COMP 4513-001: WEB 3

WINTER 2024 COURSE OUTLINE

credits	3 (3 hours lecture + 1 hour lab per week)
prerequisites	COMP 3512 or COMP 3612 with C- grade min
instructor	Randy Connolly rconnolly@mtroyal.ca B175-L 440-6061 https://github.com/rconnolly https://github.com/mru-comp4513-archive http://www.randyconnolly.com
office hours	As posted in my schedule. I am also available whenever I am in my office and the door is open. I can also meet remotely via appointment.
lectures	MW 2:30 – 3:50 p.m. T-231
required text	Connolly and Hoar, <i>Fundamentals of Web Development</i> (Pearson, 2022). <i>We will be using the new Third Edition of this book, which, unfortunately, is only available as an ebook or as a rental via:</i> https://www.vitalsource.com/en-ca/products/fundamentals-of-web-development-subscription-randy-connolly-ricardo-hoar-v9780135863497 Labs are available for \$15 from: https://funwebdev3rd.gumroad.com/l/lab-set-one <i>The money from sales is used to hire a student to revise and support the labs.</i>
masking	Appropriate mask use (covering your nose and mouth) is currently listed as optional in all classrooms and laboratories. If you are unwell or exhibiting any symptoms of Covid-19, please stay home and DO NOT come to campus.
land acknowledgement	Mount Royal University is located within the traditional territories of the Niitsitapi (Blackfoot) and the people of the Treaty 7 region in Southern Alberta, which includes the Siksika, the Piikani, the Kainai, the Tsuu t'ina, and the Iyarhe Nakoda. We are situated on land where the Bow River meets the Elbow River, and the traditional Blackfoot name of this place is "Mohkinstsis," which we now call the City of Calgary. The City of Calgary is also home to the Métis Nation.
discord	There is a Discord server for the course. It is an optional resource for the course that will require creating an account on Discord. The invite code for the course is: https://discord.gg/cEA9bjnpbm .

description

This course covers advanced web development topics, applying the foundation client and server techniques learned in Web I and II (or COMP3612). In addition to practical theory, students will learn more about asynchronous JavaScript on both the client and server, TypeScript, noSQL database systems, authentication systems, search engines, and caching. Students will use a modern API to develop an application using maps, social network plug-ins, or other advanced web technology.

This is a fourth-year university course. As such, there are no assigned lab times in this course. While I will provide some labs for you to complete on your own, a substantial amount of learning is going to come from your own self-directed learning. **The field of web development is one characterized by frequent change and constant self-directed learning; you might as well get used to it here in this course.**

If it's any consolation, this year will be the fourth year in a row in which I have to substantially revise my React labs for this course. You would think by version 18 it would be pretty stable, but nope!

grading

The final grade for this course will be determined as follows:**

Assignment 1 (due week of Feb 26-Mar 1)	20%
Assignment 2 (due Apr 8)	14%
Lab Exercises (4 worth 2.5% each)	15%
Quizzes (2 @ 3% each)	6%
Midterm (In class Feb 14)	15%
Final Exam (2 hours)	35%

** - assignment weightings or due dates might change

Percentage grades will be converted to letter grades as follows:

95-100	A+	67-69	C+
85-94	A	63-66	C
80-84	A-	60-62	C-
77-79	B+	55-59	D+
73-76	B	50-54	D
70-72	B-	<50	F

course format There are three hours of scheduled instruction per week. New material will be formally presented in lectures. You will have to do a fair amount of work outside of scheduled class time.

labs Students will work on labs on their own. The labs are step-by-step exercises that are an essential mechanism for learning the content. Each lab has one or more Test Your Knowledge exercises as well: these require the student to solve a problem on their own by applying the material covered in the step-by-step exercises. **These will be assigned as homework: you will have a week or so to complete it and be assigned a pass/half/fail mark.**

I will likely do the lab marking at the beginning/end of lectures or during office hours. This will be done by demoing your work on your own laptop or on a lab computer. You will find it very difficult to successfully complete this course without doing these lab exercises.

examinations The midterm date will be Feb 14. Students will not normally be permitted to write a missed test at a later date. If alternative arrangements can be arranged, they must be made with the instructor at least a week prior to the date of the test.

educational outcomes At the end of the course, the acquired knowledge and skills will enable the student to successfully accomplish many of these outcomes:

- Plan and implement cross-browser and cross-platform solutions.
- Write an asynchronous web application using programming best practices.
- Integrate third-party web services/APIs into an application.
- Understand and utilise sessions across multiple site visits.
- Apply a variety of password protection techniques.
- Understand and apply search engine optimization techniques.
- Understand principals of search engines
- Understand virtualization
- Understand the basic principles of LAMP administration
- Understand web usage statistics.
- Utilise software versioning systems in a web development context

**artificial
intelligence tools**

[PLEASE READ]

Over the past 18 months, many students have found themselves very tempted to use AI tools like ChatGPT or GitHub Copilot to write their assignments. While these tools can improve experienced programmer productivity (especially by eliminating simple boiler-plate code), I do think that in a university context using these tools is a mistake.

First off, for labs it is often easy for me to discern this. If I think your code has been AI generated, I will ask you to explain in detail what your code is doing; failure to do so may result in the student receiving a zero grade and a record added to their account with the University's Office of Student Conduct.

Second, the functional value of a course such as this is to help you develop your capability to problem-solve and create non-trivial web applications. Remember: employers are not interested in hiring people for their copy and paste abilities.

Third, the instructors in both the Web 2 and Web 3 courses (i.e., Jordan and me) have noticed a dramatic drop off in exam and quiz performance over the past 18 months (i.e., midterms and exams declining from the long-time average of 75% to 60%, with a dramatic growth in the number of Ds and Fs). We suspect that this is a consequence of students relying on these tools instead of putting in the hard-work to actually learn the material by doing the coding themselves.

Fourth, using these AI tools can be a real slippery slope: for instance, you may think you will use them just this once for the first lab, but because you didn't figure out how to do the first lab, you are behind, and need to use them for the next one, and so on. Before you know it, you'll be getting a 35% on the midterm.

Fifth, without knowledge and experience, adapting AI-provided code is very difficult and time-consuming.

Sixth, it is common in the software development field for prospective employees to pass a so-called technical interview, which typically involves solving coding problems on paper or a white board. With the spread of these AI tools, this will likely be even more common in any type of development job interview.

**Requirements for
third-party tools
and accounts**

You will need to create accounts with a variety of third-party sites and tools and accept the terms of service for developer access to these services. Students are advised that, if the software allows, to limit personal information entered into the software as the data resides with the software company outside of the University.

In the event you do not wish to use a particular third-party party tool you can opt-out. Do note that this will require you to research and suggest an alternative service, which may be acceptable depending on the characteristics of the tool. You agree that the instructor will have sole authority to decide if an alternative is acceptable.

If any of this gives you pause, you should consider withdrawing from the course now, while you can still have your fees refunded.

**marking and
hosting**

Quizzes will be short (5-10 minute) single questions assessed in class. You will receive advanced notification via D2L and in class whenever a quiz is scheduled.

Labs will be more complicated to mark. I will likely do the lab marking at the beginning/end of lectures or during office hours. This will be done by demoing your work on your own laptop or on a lab computer.

For the two assignments, I am going to ask you to host your assignments on a third-party site. For the first two assignments, you will likely find GitHub Pages (<https://pages.github.com/>), Netlify (<https://www.netlify.com/>), or Firebase Hosting works quite well. For the second assignment, you will need Node, so the hosting requirements are a bit more complicated. There you could use glitch (free), or any type of live publicly-available server, which might require an expenditure in the \$15-\$25 range (though in most years students do not need to spend any money).

Assignments will be considered late if submitted after the time specified on the assignments. Unless an assignment states otherwise, it will be accepted up to two days late; however, 10% will be deducted for being late (even for part of a day late). Assignments will not be accepted more than two days late. This includes weekends, so if an assignment is due Friday at 12:00 p.m. then Sunday at 12:00 p.m. is the latest it can be handed in. Start your work early and schedule adequate time for completion.

software requirements

As far as web development related software, you are initially going to need both Chrome and the FireFox browser.

You are also going to need some type of code editor. Visual Code (<https://code.visualstudio.com/>) has become the go-to editor in the web community, so I'd recommend using it. But you are welcome to use other options (e.g., Brackets, Sublime, WebStorm, etc).

For the assignments and some of the labs, you will be consuming a variety of external APIs. While these APIs can be examined within a browser, it may be easier to use a third-party desktop tool. The most popular are postman (<https://www.postman.com/>) and Insomnia (<https://insomnia.rest/>). Both have free options, so I would recommend installing one or both.

Version control is an important part of the course. You will need to install the command-line Git tool, and create an account on github.com.

You will also need to install Node.

accessibility statement

Disability-Related Accommodations:

Students registered with Access and Inclusion Services and have an Academic Accommodation Letter should schedule a private conversation with me, as soon as possible, to discuss how your accommodations will be implemented into the design of this course. A conversation may take place in-person, online, or over the phone. Requests to discuss accommodations during or between classes will not be accepted. Acknowledgement of your accommodations will then be provided via email or Faculty Acknowledgement Form, depending on course delivery method.

If you are a student who experiences disability-related barriers and has not registered with Access and Inclusion Services, you may wish to inquire about services by contacting the Access and Inclusion Services office directly.

Non-Disability-Related Accommodations:

Students requiring non-disability-related academic accommodations (as related to the protected grounds in Human Rights Legislation) are encouraged to contact Access and Inclusion Services to explore what academic accommodations may be available to assist them in order to participate fully in their academic studies.

Students can contact Access and Inclusion Services by visiting the office in Y201, or by contacting them at (403) 440-6868 or accessibility@mtroyal.ca.

cheating

It is expected that all work handed in by a student will be original work that has been done by the individual. If it is not, then this act of intellectual dishonesty will be dealt with severely. An *Academic Dishonesty Incident Report* will be filed with the Office of Student Conduct. If a record of previous academic dishonesty is established, the case will be forwarded to the Academic Integrity Board. The complete process is described in Mount Royal's *Code of Student Conduct*.

Normally, the sanction for any student accused of cheating will be zero on the assignment (in the case of one student giving part of his/her assignment to another student, both students are considered to be cheating). Sanctions for further incidents of cheating by the same student will be reviewed by the Academic Integrity Board and may ultimately result in expulsion from the University.

While students are expected to work reasonably independently, we do not expect you to work in isolation. Often you learn best when working with others on an assignment. So what degree of collaboration is expected and, indeed, encouraged, and what is deemed to be cheating?

In general, we encourage things like bouncing ideas off one another, discussing which of two alternate solutions might be better (and why), and getting another's ideas on how to resolve a difficulty that you have already spent time on. You should not be working so closely together that someone else's solution becomes incorporated into your product. These general guidelines apply to **any type** of assignment. Some more detailed guidelines for programming assignments follow.

When you get to the point of writing the actual program, you must work independently. It is **NOT** acceptable to be coding with others. Write your program by taking your design, possibly fleshing out some details, and writing the code in the appropriate programming language. If you have difficulty with a certain statement, check your notes or your text. If you are still having problems with the code, then you can ask others for help. However, you must master the language as quickly as possible, and not rely on others. When testing the program, if you are having problems with a section and have spent some time trying to find the problem yourself, it is a good idea to ask others if they can help you. Other people will often see errors that you cannot see because you are too close to your solution – that is, you no longer see it clearly. Don't forget that if you need help at any time, the Instructional Assistant and your Instructor are available.

When complete, two students' programs should be essentially independent of one another. Each student should have attempted each step of the problem-solving method alone before discussing it with others. In this way you can develop your own skills, while still learning from (and helping) others.

4513 TENTATIVE SCHEDULE

Week	Lectures	Work
Jan 8-12	Course Intro + JavaScript Review Additional JavaScript Features	
Jan 15-19	Additional JavaScript Features TypeScript	
Jan 22-26	React React	
Jan 29-Feb 2	React React	
Feb 5-9	React React	
Feb 12-16	React + Review Midterm	
Feb 19-23	Reading Week	
Feb 26-Mar 1	Node Node	<i>Assign #1 due</i> date might change
Mar 4-8	Node Node	
Mar 11-15	NoSQL NoSQL	
Mar 18-22	Security Security	
Mar 25-29	Security Security	
Wed Apr 3, Mon Apr 8	Security Review	<i>Assign #2 due</i> date might change