mount royal university
department of mathematics and computing

COMP 3512: Web 2

Fall 2023 course outline

credits

3 (3 hours lecture + 1 hour lab per week)

prerequisites

COMP 2511, COMP 2503, and COMP 2521 with C- grade min

instructor

Randy Connolly
rconnolly@mtroyal.ca
B175-L
440-6061
https://github.com/rconnolly
https://github.com/mru-comp3512-archive
http://www.randyconnolly.com
http://www.funwebdev.com

office hours

I'll run office hours via Google Meet but the time will likely vary from week-to-week. I can also meet remotely via appointment.

lectures

MW	11:30 – 12:50 p.m.	B220

tutorials

R	11:30 – 12:20 p.m.	E141
R	12:30 – 13:20 p.m.	E141

required text

Connolly and Hoar, *Fundamentals of Web Development* (Pearson, 2023).

We will be using the new Third Edition of this book, which, unfortunately, is only available as an ebook. Link for book is: 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

The money from sales is used to hire a student to revise and support the labs.

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.

description

This course focuses on the concepts and technologies needed to develop web-centric applications. The main topics are:

- The overall architecture of contemporary Internet applications
- PHP
- JavaScript
- Security.

Most students find the lab and assignment work in this course to be quite time consuming. The second assignment (which is a group assignments) typically requires the investment of 50-90 hours of work. Be sure to plan your life accordingly.

This is a third-year university course in a sub-discipline in which SELF-DIRECTED LEARNING IS AN ESSENTIAL CHARACTERISTIC. While there are labs and lectures to help you learn the content, a substantial amount of learning is going to come from your own explorations, from reading the textbook, and with labs and assignments.

grading

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

Assignment 1 (due Oct 14)	13%
Assignment 2 (due Dec 11)	21%
Lab Exercises (7 worth 1% or 2% or 3% each)	10%
Tutorial Exercises (worth 1% each)	7%
Quizzes (3 @ 3% each)	9%
Midterm (Oct 16)	15%
Final Exam	25%

** - assignment weightings 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 four hours of scheduled instruction per week. New material will be formally presented in lectures. Tutorials will be used to reinforce the lectures by providing an opportunity to study examples and practice the application of concepts. Tutorials are not just for students who require extra help; they are an integral part of the course and attendance is expected. You will also have to do a lot of work outside of scheduled class time.

It is possible that some videos ***might*** be available for viewing at your own leisure and schedule, but there is no guarantee that this will be the case. If you want that kind of experience, you'll have to look into Athabasca University options instead.

In order to provide a searchable record of comments between students and myself, I have created a Discord server for the course. There are channels for lectures, tutorials, and assignments. This will require creating an account on Discord. The invite code for the course is: https://discord.gg/EnKBvag.

tutorials

During the scheduled tutorials, you will be doing small coding exercises in a group setting. **You will receive a pass/half/fail mark each week for participating.**

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 to complete it and be assigned a pass/half/fail mark. I will likely mark them in Google Meet.**

You will find it very difficult to successfully complete this course without doing these lab exercises.

examinations

The midterm date will be Oct 18. 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.

course expectations

Learning web development is a “hands-on” learning. The only way you can learn it is by doing it. The materials, discussions, and other activities provide the foundation of the learning process. However, all these are useless unless the student actually does the work using either paper and pencil or programming using the computer. As such, there are a number of lab and tutorial exercises which the student is expected to complete and submit. In addition, there will be graded assignments that will help determine the progress the student is making.

Most job interviews within development typically consists of examinations in which the interviewee must solve a variety of programming problems on paper in front of the interviewer. You might as well learn how to work and think on your own!

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

In the past, I marked weekly lab exercises during the scheduled labs. But last year, during our switch to remote learning, I found the time investment required to do that same marking was quite onerous, as I had to download student’s work, load it in browser, then briefly look at the code, and then do this for all 40 students every week.

To make marking of lab exercises easier, I may ask students to share a screen with me using Google Meet and show me their work.

For the two major assignments, I am going to ask you to host your assignments on a third-party site. For the first assignment, 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 PHP and maybe MySQL, so the hosting requirements are a bit more complicated. There you could use heroku, 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).

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 first assignment 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.

While we won't be using Node in this course, the Node Packager Manager (npm) is an essential tool for most practicing JavaScript developers. To get npm, you will need to install Node. Both Node and npm are command-line tools.

By November, you are going to need to run PHP and MySQL. There are multiple options here. You can install PHP and MySQL natively in Windows and Mac, though it is much easier to do this with a Mac. Alternately, you can make use of XAMPP for either Windows or Mac. This is a much easier alternative, and is the one I would recommend unless you really enjoy configuring software. In the past, some students have enjoyed using PhpStorm as an IDE instead of Visual Code for the PHP part of the course. I've never used it myself, but you may want to experiment with this program as well. As well, while not at all necessary, you may find MySQL Workbench (https://www.mysql.com/products/workbench/) a helpful tool for examining a MySQL database and running queries.

In November, in addition (or as an alternative) to MySQL, you will also be using SQLite. The advantage of SQLite is that it doesn't require a running database server or any other tools. However, you might find it helpful to install and use either SQLiteStudio (https://sqlitestudio.pl/) or the DB Browser tool for SQLite (https://sqlitebrowser.org/), both of which let you examine tables and run queries.

assignments

Assignments will consist of larger applications of the topics covered in lectures. The assignments will normally require several weeks of work and may be broken down into more than one component, e.g. an interface, client scripts, and server scripts.

NOTE: In order to obtain a grade of C or better in the course, you must obtain a minimum of 50% in the overall mark for assignments. Failure to pass the assignment portion of the course may result in a D+ grade or lower for the course, regardless of the marks in the other components.

Problem solving techniques and skills can only be acquired through practice and through the study of increasingly more difficult problems. The assignments all involve problem solving. It is very important that you understand how you solved the problem, and not just be happy with handing in an assignment that produces the requested results.

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.

technology in lectures

Staying focused on the professor in lectures has always been an issue. This semester, with remote lectures, this problem is going to be particularly acute. The temptation to multi-task and watch videos, answer emails, play games, etc will likely be quite strong at times.

I'll try my best to be engaging, but I find that I'm much less interesting and enthusiastic sitting in front of a camera than when I'm standing and teaching in a room full of people. So, it is going to be partly your responsibility to try and stay focused in class. **I recommend strongly that you avoid engaging in attention-sapping activities such as watching videos while trying to listen to my lectures!**

Remember I won't have the luxury of seeing the dawn of understanding or the fog of bewilderment on your faces this semester in our remote setting. That visual feedback is normally essential for me in knowing when I need to move along quicker or take some additional time re-covering content.

accessibility statement

If you are a student with an Academic Accommodation Memo and Professor Acknowledgement form from Accessibility Services, please make an appointment with me as soon as possible to discuss your Accommodations in a private and confidential setting. Requests to review and sign Academic Accommodation documents should not be made during or between classes. If you are a Student Experiencing a Disability who may require Academic Accommodation and have not yet registered with Accessibility Services, please contact their office at accessibility@mtroyal.ca or find additional information at www.mtroyal.ca/accessibility . You must be registered with Accessibility Services to access Academic Accommodations. If you require Academic Accommodations for a reason other than Disability, please make an appointment with me to discuss or contact Diversity and Human Rights Services at 403-440- 5956.

recording of lectures

Mount Royal University has contracted Google to provide educational technology software for its courses; where, the instructor will record lectures using Google Meet and make them available to registered students on Blackboard for up until 10 days after the course is completed. Users, including any recorded student participants, are advised that the personal information collected during the recordings will only be used for educational purposes and is collected under the authority of the FOIP Act – section 33(c) and the Post-Secondary Learning Act in the Province of Alberta.

Users are further advised that the downloading of posted videos other than for the purposes of student personal learning through Blackboard, may violate the copyright of the course instructor or others. For additional questions regarding the collection, use, disclosure and protection of personal information please contact: Amy McCarthy, Department Administrative Assistant, Faculty of Science and Technology, 4825 Mount Royal Gate SW - Calgary, AB - T3E 6K6 - amccarthy@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.

educational outcomes

Mount Royal University had identified six university-wide learning outcomes that it believes are critical in order to prepare its graduates for workplace success and a life of continuous learning. Generally speaking, “outcomes” are goals, results, objectives that you should derive from the University, from a program of study, and from a particular course. This course will emphasize the following outcomes:

University-Wide Learning Outcomes

Computer Literacy

- ♦ Knowledge of basic computer concepts and terminology (lectures, exams).
- ♦ Competence in basic computer operations and hardware (labs).

Thinking Skills

- ♦ Problem-solving (labs, assignments, exams).
- ♦ Analytic thinking appropriate to discipline of information systems (labs).

Communication

- ♦ Convey ideas and information through writing and images (labs, assignments).
- ♦ Use available technology as an aid to effective written, verbal, and visual communication (labs, assignments).

Course and Discipline Outcomes

- How the internet works.
- Developing with Javascript.
- Consuming REST web services.
- Using databases in web applications.
- Developing with server-side technology (PHP).
- Software design in web applications. This includes layered software architectures as well as tiered designs for scalability and reliability.
- Mechanisms for maintaining state in web applications. This is one the most topics in the course since it is the principal difference between web application development and non-web application development.

3512 TENTATIVE SCHEDULE

Week	Lectures	Work
Sep 11-15	PHP Language Fundamentals PHP Language Fundamentals	
Sep 18-22	PHP Arrays and Superglobals PHP Arrays and Superglobals	
Sep 25-Sept 29	Working with Databases Working with Databases	
Oct 2-6	State Management State Management + review	
Oct 9-13	**Reading Week**	*Assign #1 due* date might change
Oct 16-20	**Midterm** JavaScript Language Fundamentals	
Oct 23-27	JavaScript Language Fundamentals JavaScript Language Fundamentals	
Oct 30-Nov 3	Arrays + Objects Arrays + Objects	
Nov 6-10	Objects + Functions Functions	
Nov 13-17	DOM DOM	
Nov 20-24	DOM DOM	
Nov 27-Dec 1	Asynchronous Coding Asynchronous Coding	
Dec 4-8	Asynchronous Coding Asynchronous Coding	
Dec 11	Review	*Assign #2 due* date might change