

ENVR 4000 T06 / GEOG 4670 T04 / GEOG 7010 T16
Programming for Environmental Science Data Analysis
GEOG 2xxx / ENVR 2xxx: Programming for Environment and
Geography
Course Syllabus
Winter 2026

Instructor:	Dr. Juliana M. Marson (Juliana.MariniMarson@umanitoba.ca)
Class schedule:	TR 1:00-2:15 pm in 321 Wallace (please arrive at least 5min earlier)
Tea is for Teaching:	Every weekday from 15:30 to 16:00 (longer meetings can be scheduled as needed – send me an email!)

Course objectives

This course's main goal is for students to start developing a life-long skill that can be applied to any field: programming. It does not matter if you want to produce nice figures with your data for a paper, if you want to build a new videogame or a website, or if you need to perform tedious and repetitive file processing – you need to know how to program. Students will learn about problem-solving with computers, will become familiar with programming languages (the current choice is MATLAB), and will experiment with manipulating different types of data related to Environment & Geography.

Disclaimer: parts of this syllabus and this course were adapted from David J. Malan's course (CS50) with permission (Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license).

Suggested readings

- Menke, William., and Joshua E. Menke (2012). *Environmental Data Analysis with MatLab*. 1st ed., Elsevier.
https://search.lib.umanitoba.ca/permalink/01UMB_INST/1p55dqn/alma99149107720001651
- Spraul, V. A. (2012). *Think like a programmer: An introduction to creative problem solving*. No starch press.
- Johnson, R. (2010). *The Elements of MATLAB Style*. Cambridge: Cambridge University Press.
doi:10.1017/CBO9780511842290
(https://search.lib.umanitoba.ca/permalink/01UMB_INST/1p55dqn/alma99149503281001651)
- Harvard's CS50x course, by David Malan (<https://cs50.harvard.edu/x/2022/>)
- Harvard's CS50P course, by David Malan (<https://cs50.harvard.edu/python/2022/>)

Content (subject to changes)

1. Introduction to programming
 - 1.1. What is programming?
 - 1.2. Representing numbers
 - 1.3. Representing text, images, videos, and sounds
 - 1.4. Algorithms
 - 1.5. Pseudocode
2. MATLAB overview
 - 2.1. MATLAB GUI
 - 2.2. Navigating folders
 - 2.3. Simple arithmetic
 - 2.4. Variables
 - 2.5. Arrays, element access, and matrix operations
 - 2.6. Writing scripts
3. Essentials of programming (applied to MATLAB) – PART I
 - 3.1. Functions
 - 3.2. File I/O
4. Visualizing data
 - 4.1. Line plots
 - 4.2. Scatter plots
 - 4.3. Histograms
 - 4.4. Bar and whiskers
 - 4.5. 2-D and 3-D fields
 - 4.6. The m_map package (maps)
5. Essentials of programming (applied to MATLAB) – PART II
 - 5.1. Conditionals
 - 5.2. Loops
 - 5.3. Debugging
 - 5.4. Exceptions
 - 5.5. Regular expressions
6. Data analysis in MATLAB
 - 6.1. Basic statistical functions
 - 6.2. Other useful functions to perform analyses

Assessment

Daily Checks	30%
Coding Project	30%
Quiz 1	20%
Quiz 2	20%

Assessment information

Daily Checks

Daily Checks are used to review content, scaffold other assignments, and so I can gauge your understanding of the material. They will consist of printed questions which you'll need to answer with pencil/pen instead of using your computer. The activity will take place on the last 10 min of class, and you will hand in your paper at the end. On the following class, I will give your paper back and we will discuss the solution together. You are encouraged to take notes and use that to study for the quizzes.

Each Daily Check is worth 1 point. This is the rubric I will use:

Score	Description
1.0	Shows reasonable to good understanding of the problem. Most or all concepts are applied correctly. Solution is functional or close to functional, even if it requires minor fixes.

0.5	Shows effort but lacks meaningful progress toward solving the problem. Attempts to apply concepts learned in class but does so incorrectly or incompletely. Solution contains major errors or does not achieve the intended outcome.
0	Absent or shows negligible effort to solve the problem.

Coding project

The coding project is your opportunity to take your newfound savvy with programming out for a spin and develop your very own piece of software. So long as your project draws upon this course's lessons and relates to Environment and Geography, the specific nature of your project is entirely up to you, albeit subject to my approval. If you are a grad student, this is your chance to build something that will help you with data analysis for your thesis, but also that might be useful for other people working in your area of research. If not, just write something that can solve an actual problem: strive to create something that outlives this course.

This project should be built progressively throughout the course, not left to be done one day before the deadline! To ensure that, you will submit your project in **five stages**. This will give me the opportunity to give you feedback, check if you are on track, remove the temptation of letting AI write your project for you, and help when necessary.

Stage I: Project idea (Due **January 29**)

In this stage, you will identify a problem related to Environment and/or Geography that you would like to address through programming. The goal is to think about how code can be used to solve a real, meaningful problem. I am happy to discuss your idea before you submit it in writing by email. Your idea does not need to be fully developed yet, but it should:

- Clearly relate to course themes
- Describe the problem you want to solve
- Indicate who might benefit from the solution

Feedback at this stage is meant to help you refine and focus your project early on.

Stage II: Pseudocode (Due **February 24**)

In this stage, you will plan your program by writing detailed pseudocode. This is a crucial step where you translate your idea into a clear, step-by-step logical plan — similar to the PB&J sandwich exercise we did in class. Your pseudocode should:

- Be written in plain language
- Describe each step of the program in order
- Include comments explaining the purpose of each section

At this stage, the focus is on clarity and planning, not syntax or efficiency.

To support your own learning, **AI tools are not permitted for this stage.**

Stages I and II will be evaluated together and emphasize careful thinking and preparation before coding. They are worth 10% of your final grade. Here is the tentative rubric:

CATEGORY	Developing	Proficient	Strong
Originality & Relevance (2%)	The project idea is still forming and would benefit from clearer alignment with the course scope or a more specific problem statement (0%)	The project idea aligns with the course scope and identifies a relevant problem, though its purpose or broader significance could be further clarified (1%)	The project idea clearly aligns with the course scope, addresses a meaningful problem in Environment or Geography, and shows potential usefulness beyond the course (2%)
Pseudocode Style & Readability (2%)	Comments and structure are present but would benefit from clearer organization and closer adherence to style guidelines (0%)	Comments generally follow style guidelines, and the pseudocode is readable, with minor opportunities for improved clarity or consistency (1%)	Comments consistently follow style guidelines, and the pseudocode is clean, well-organized, and easy to follow (2%)
Pseudocode Design & Detail (6%)	The pseudocode outlines the general idea of the program and would benefit from additional detail or clearer step-by-step logic (0-2%)	The pseudocode includes many key steps and demonstrates solid planning, though some steps could be expanded or clarified (3-4%)	The pseudocode is detailed, clear, and thoughtfully structured, addressing all steps needed to reach the program's intended goal (5-6%)

Stage III: Initial code implementation (Due **March 17**)

In this stage, you will write the first working version of your code based on your pseudocode. This is expected to be a work in progress—bugs, incomplete features, and rough edges are normal and encouraged at this point. Your submission should include:

- Code that attempts to implement all major components of your program
- Comments explaining what the code is intended to do
- Description of bugs you were not able to fix by yourself, limitations, or unfinished features that are not essential to the program's execution
- References for any external resources or functions you used

You may search online for help with specific functions or syntax, but **AI tools are not permitted for writing the code itself.**

The goal of this stage is to demonstrate progress, effort, and problem-solving—not perfection.

Stage III is worth 10% of your final grade. Here is the tentative rubric:

CATEGORY	0%	1%	2%	3%
Correctness	The code framework is emerging and would benefit from additional components to reach the program's end goal.	The code includes several key components and shows progress toward the program's goal.	Most components needed to reach the program's goal are present, even if some parts are not yet fully functional.	All major components required to reach the program's goal are present, demonstrating a complete first implementation (even if with bugs)

Comments	Some comments or references are present; expanding documentation would further support readability and understanding.	The code includes comments and bug descriptions, with room for more consistency or detail.	The code is reasonably commented, bugs are described, and external references are appropriately cited.	The code is thoroughly documented, with clear comments, well-described bugs, and complete, well-integrated references.
Design	The program structure is emerging and would benefit from clearer organization or modularization.	The code shows some organizational structure, with opportunities for modularization and improved efficiency.	The code is well organized, reasonably efficient, and makes use of functions or modular components.	The code is cleanly organized, efficient, and thoughtfully modularized, supporting readability and future development.

The final 1% will be awarded if you include a short reflection ($\approx 150\text{--}250$ words) that demonstrates thoughtful engagement with your code. The reflection may address:

- One design decision you made and why
- A challenge or bug they encountered and how you solved it
- What part of the code you feel most confident about
- What you plan to improve or add before the final submission

Scoring:

1% – Reflection is complete, specific, and clearly connected to the submitted code

0.5% – Reflection is present but brief or somewhat general

0% – Reflection is missing or not connected to the code

NOTE: Submissions should reflect the student's own development over time. Code that shows a sudden and unexplained leap in complexity may be reviewed more closely.

Stage IV: Final code refinement (Due **April 02**)

In this stage, you will refine and improve your program by debugging, cleaning up, and polishing your code. You are encouraged to:

- Discuss issues with classmates
- Research solutions online
- Use AI tools to assist with debugging or understanding errors

However, the core logic and structure of the program should remain **your own work**.

By the end of this stage, your program should:

- Run successfully

- Perform its intended task
- Be clearly organized and readable

This stage emphasizes learning how to improve existing code using available resources responsibly.

Stage V: In-class presentation (April 07 or 09)

In the final stage, you will present your project in 5-10 minutes to the class. Slides are not mandatory but bringing a flowchart that visually shows how your code works is encouraged. This is an opportunity to showcase your work, run your code live, and demonstrate your understanding of the program. During your presentation, you should:

- Explain the problem your program addresses
- Demonstrate how the program works
- Describe key parts of your code
- Discuss challenges you encountered and how you addressed them

The focus is on clear communication and understanding, not on having a flawless or overly complex program.

Stages IV and V will be evaluated together and are worth 10% of your final grade. Here is the tentative rubric:

CATEGORY	Developing	Proficient	Strong
Correctness (1%)	The program runs partially or encounters errors that prevent full execution (0%)	The program executes successfully in many cases, with minor issues or limitations (0.5%)	The program executes successfully and performs as intended (1%)
Design (1%)	The program's structure is still evolving and would benefit from clearer logic or efficiency (0%)	The program demonstrates a logical design, with some opportunities for refinement (0.5%)	The program is clearly structured, efficient, and thoughtfully designed, including error handling (1%)
Style (1%)	Code readability is emerging and would benefit from clearer formatting or commenting (0%)	The code is generally readable, with minor improvements possible (0.5%)	The code is easy to read, well formatted, and consistently commented (1%)
Quality of output (2%)	Output is incomplete or difficult to interpret (0%)	Output contains correct information, with opportunities to improve presentation or usability (1%)	Output is accurate, meaningful, and presented in a clear, practical, and engaging way (2%)
Familiarity with the program (3%)	Student is beginning to explain how the code works and would benefit from deeper familiarity (0%)	Student explains key parts of the code and demonstrates partial understanding (1%)	Student demonstrates strong familiarity with the program and is comfortable explaining, using, and making small adjustments to the code (2-3%)
Demonstration (2%)	The demonstration is incomplete or unclear,	The demonstration is successful and shows the	The demonstration is clear and effective, and

with limited explanation of the program (0%)	program in action, though the purpose or usage could be explained more clearly (1%)	the student communicates the program's purpose, functionality, and relevance with confidence (2%)
--	---	---

This project is designed to support steady learning and experimentation. Each stage builds on the previous one, and progress over time is valued more than producing a perfect final product.

Quizzes

The format of quizzes will be discussed in class.

Lateness

Late submissions (of coding project stages) will be penalized at a rate of 0.1% per minute.

- If you submit 10 minutes late, your score will be penalized 1%. Your score will thus be 99% of what it would have been if submitted on time.
- If you submit 60 minutes late, your score will be penalized 6%. Your score will thus be 94% of what it would have been if submitted on time.
- If you submit 1,000 minutes (just over 16 hours) late, your score will be penalized 100%. Your score will thus be effectively zeroed.

Academic honesty

The course's philosophy on academic honesty is best stated as "be reasonable." The course recognizes that interactions with classmates and others can facilitate mastery of the course's material. However, there remains a line between enlisting the help of another and submitting the work of another (**including of AI**). The essence of all work that you submit to this course must be your own. In Stage Generally speaking, you can ask for help when asking for help, you may show your code to others, but you may not view theirs, so long as you and they respect this policy's other constraints. Collaboration on the course's Daily Checks is not permitted. Collaboration/AI consultation on quizzes and the course's coding project is permitted **to the extent prescribed by its specification**. If I suspect any violation of academic integrity, including the use of AI or external help, your assignment may be nullified.

Regret clause. If you commit some act that is not reasonable but bring it to my attention within 72 hours, I may allow re-submission with potentially different conditions.

In cases of repeated acts, I will refer the matter for further disciplinary action.

Below are rules of thumb that (inexhaustively) characterize acts that the course considers reasonable and not reasonable. If in doubt as to whether some act is reasonable, do not commit it until you solicit and receive approval in writing from me. Acts considered not reasonable by the course are handled harshly (see [Academic Integrity](#) section).

Reasonable

- Discussing the course's material with others in order to understand it better.

- Finding resources at MATLAB Help Center (not AI) that help you build your coding project in aspects not covered in class (e.g., a specific function) or with non-trivial errors, as long as those are properly cited (including the link to the source).
- Incorporating **a few** lines of code that you find online or elsewhere into your own code, **provided that those lines are not at the core of your code functionality and that you cite the lines' origins.**
- Sending or showing code that you've written to someone at Stage IV of your coding project, possibly a classmate, so that they might help you identify and fix a bug.
- Turning to the course's instructor for help or receiving help from the course's instructor during the exam.
- Whiteboarding solutions with others using diagrams or pseudocode **but not actual code.**
- Working with (and even paying) a tutor to help you with the course, provided the tutor does not do your work for you.

Not Reasonable

- **Using ChatGPT (or other AI) to generate parts or the entirety of your code.**
- Failing to cite (as with comments) the origins of code or techniques that you discover outside of the course's own lessons and integrate into your own work, even while respecting this policy's other constraints.
- Looking at another individual's work during Daily Checks or Quizzes, unless otherwise specified.
- Paying, offering to pay, or simply asking an individual for work that you may submit as (part of) your own.
- Submitting an existing code/idea as your own coding project.
- Submitting (after possibly modifying) the work of another individual beyond the few lines allowed herein.
- Submitting the same or similar work to this course that you have submitted or will submit to another.
- Turning to anything or anyone (besides the course's instructor) for help or receiving help from anything or anyone (besides the course's instructor) during Daily Checks and Quizzes (unless otherwise specified).
- Viewing another's coding project and basing your own project on it.

Grading scale

LETTER GRADE	PERCENTAGE RANGE	DESCRIPTION
A+	90-100	Exceptional
A	80-89.9	Excellent
B+	75-79.9	Very Good
B	70-74.9	Good
C+	65-69.9	Satisfactory
C	60-64.9	Adequate
D	50-59.9	Marginal
F	0-49.9	Failure

NOTE 1: All final grades are subject to departmental review.

NOTE 2: It is very likely that this course draws quite the spectrum of students, including "those less comfortable" with programming, "those more comfortable," and those somewhere in between. However,

what ultimately matters in this course is not so much where you end up relative to your classmates but **where you end up relative to yourself when you began.**

Each student's final grade is individually determined at term's end. Remarkable effort and upward trending are considered. The course is not graded on a curve. Those less comfortable and somewhere in between are not at a disadvantage vis-à-vis those more comfortable.

Tentative schedule

	TUESDAY	THURSDAY
WEEK 00	Jan 06 L00 – Welcome	Jan 08 L01 – Introduction to Programming
WEEK 01	Jan 13 L02 – MATLAB Overview (2.1-2.3)	Jan 15 L03 – Variables, Arrays, Element Access (2.4-2.5)
WEEK 02	Jan 20 L04 – Element access, matrix operations, scripts (2.5-2.6)	Jan 22 L05 – Functions (I)
WEEK 03	Jan 27 (Luiz) L06 – Functions (II)	Jan 29 (Luiz) L07 – Functions (III) CP SI: <i>submit idea</i>
WEEK 04	Feb 03 L08 – File I/O (I)	Feb 05 L09 – File I/O (II)
WEEK 05	Feb 10 L10 – Data Visualization (I)	Feb 12 L11 – Data Visualization (II)
WINTER TERM BREAK	Feb 17 NO CLASS	Feb 19 NO CLASS
WEEK 06	Feb 24 (Paulo) Quiz 1 CP SII: <i>pseudocode</i>	Feb 26 (Paulo) L12 – Conditionals (I)
WEEK 07	Mar 03 L13 – Conditionals (II)	Mar 05 L14 – Loops (I)
WEEK 08	Mar 10 L15 – Loops (II)	Mar 12 L16 – Exceptions
WEEK 09	Mar 17 L17 – Debugging CP SIII: <i>first draft</i>	Mar 19 L18 – Regular Expressions
WEEK 10	Mar 24 L19 – Data Analysis in Matlab I	Mar 26 L20 – Data Analysis in Matlab II
WEEK 11	Mar 31 TBD	Apr 02 Quiz 2 CP SIV: <i>final submission</i>
WEEK 12	Apr 07 Presentations (CP SV)	Apr 09 Presentations (CP SV)

Assignment extensions

Reasons for granting an extension (assignments): a death in your immediate family, an illness in either yourself or in a dependent (requires email notification BEFORE the due date), and requirement to travel for work. The instructor will not accept an email dated AFTER the due date.

Reasons for not granting an extension: having another assignment or midterm on the same day, being away from the university for a personal reason (i.e. holiday or personal vacation), being too busy with other course work (i.e. having a midterm that same day or week), not attending the lectures due to personal or compassionate reasons (or other reasons), car broke down and could not submit assignment on time,

computer is not working properly and you lost the assignment, or any other reason deemed inappropriate by the instructor. This is not an exhaustive list. Please don't ask for an extension if any of these or similar reasons apply. If you know that you will be away, you **MUST** submit the assignment before the due date.

Communication

Effective September 1, 2013, the University requires all students to activate an official University email account. The U of M will only use your university email account for official communications, including messages from your instructors, department or faculty, academic advisors, and other administrative offices (http://umanitoba.ca/registrar/email_policy/).

Please use your UManitoba email address for all email communications with me (**Juliana.MariniMarson@umanitoba.ca**). You are welcome to send course-related questions about the course structure (**provided you check the syllabus first**) or content to my email address, or to request a meeting outside of office hours. I expect all emails to be professional and courteous, and will respond within 48 hours during the work week.

Tea is for Teaching

Every weekday, from 15:30-16:00, you are invited to my office to have some tea! Bring your questions from class, general wonderments, or any topic of your academic (or otherwise) journey that you wish to share/brainstorm with me. Or come in just to say hi, take a break from your day, and warm up! We can totally drink tea in silence if that's what you need. If you can't make it during this time, or need a private/longer meeting, please feel free to send me an email so that we can schedule a time slot for you.

Voluntary withdrawal date

The voluntary withdrawal date is the last date for withdrawing from this course without academic penalty. The voluntary withdrawal date for this course is **March 19, 2026**. Evaluative feedback will be provided prior to this date.

Expectations

I expect you to:

- Read and regularly consult the course syllabus. Ask for clarification on any part of the course syllabus that is unclear
- Take notes during lectures
- Arrive on time and listen actively during lectures
- Regularly check your UM email to read course announcements
- Complete all quizzes and exams without consulting any other person
- Follow university policies on plagiarism, cheating, and exam impersonation

- Let me know as soon as possible if you encounter obstacles that will prevent you from completing any aspect of the course on-time
- Be courteous and respectful to students and the instructor, and follow university policies on Respectful Work and Learning Environment and Inappropriate and Disruptive Student Behavior
- Ask questions
- Communicate with me as much as possible about any extra difficulties you encounter during the semester

You can expect me to:

- Do my best to create a welcoming, inclusive, and positive learning environment
- Treat you fairly and respectfully
- Listen to and consider all feedback you give about the course, and implement changes where possible
- Respond to questions and concerns you have sent to my University of Manitoba email account within 48 hours during the work week
- Hold regular office hours
- Make appointments outside of office hours as needed

Missing course work for medical reasons

Students who are unable to meet a course requirement due to medical circumstances are currently not required to submit medical notes. However, students are required to contact their instructor or academic advisor by email to inform of the missed work and to make arrangements for extensions, deferrals, or make-up assignments. Please follow these guidelines if you are unable to meet an academic requirement for your courses.

- Contact your instructor for term work such as a class, quiz, midterm/test, assignment, lab;
- Contact an advisor in your faculty/college/school of registration for a missed final exam (scheduled in the final examination period);
- Inform your instructor/advisor as soon as possible do not delay. Note for final exams, students must contact within 48 hours of the date of the final exam; and
- Email your instructor/advisor from a U of M email address, and include your full name, student number, course number, and academic work that was missed.

Student accessibility

The University of Manitoba is committed to providing an accessible academic community. Students Accessibility Services (SAS) offers academic accommodation supports and services such as note-taking, interpreting, assistive technology and exam accommodations. Students who have, or think they may have, a disability (e.g., mental illness, learning, medical, hearing, injury-related, visual) are invited to contact SAS to arrange a confidential consultation.

Student Accessibility Services
520 University Centre
Phone: (204) 474-7423
Email: Student_accessibility@umanitoba.ca

Copyright

All course material is copyrighted by Juliana Marson, 2026. No audio or video recording of this material, lectures, or presentations is allowed in any format, openly or surreptitiously, in whole or in part without written permission from Juliana Marson. Course materials (both paper and digital) are for the participant's private study and research, and must not be shared. Violation of these and other Academic Integrity principles, will lead to serious disciplinary action.

Policy on Respectful Work and Learning Environment (RWLE)

- Each individual has the right to participate, learn, and work in an environment that promotes equal opportunities and prohibits discriminatory practices.
- RWLE [2016], Section 2.2 The University wishes to promote and support a community which embraces diversity and inclusion, provides for equality of opportunity, and recognizes the dignity of all people.
- RWLE [2016], Section 2.3 Members of the University Community, including every student and employee, are entitled to a respectful work and learning environment that is:
 - (a) Free from Discrimination and provides for Reasonable Accommodation;
 - (b) Free from Harassment; and
 - (c) Collegial and conducive to early resolution of conflict between members of the University Community.
- Respectful Work and Learning Environment Policy: <http://bit.ly/2b63HQO>
- Student Discipline Bylaw: <http://bit.ly/2b3RL3p>

Inappropriate and disruptive student behaviour

- Inappropriate and disruptive behavior that interferes with the learning of other students, or the instructors' ability to teach will not be tolerated. Such behaviours would include but would not be limited to the following:
 - (a) threats to the physical safety of the individuals or others;
 - (b) verbal threats to or abuse of students or University personnel;
 - (c) recurring and willful damage of University property;
 - (d) inappropriate or disruptive behaviour as a result of misuse of drugs or alcohol on University property; and
 - (e) actions which habitually interfere with the learning environment or requires the inordinate time and attention of faculty and staff.
- Inappropriate or Disruptive Student Behaviour Policy: <http://bit.ly/2aI5au5>

- Student Discipline Bylaw: <http://bit.ly/2b3RL3p>

Academic integrity

Students are responsible for ensuring they understand the University of Manitoba's policy on Academic Integrity (plagiarism, cheating, and examination impersonation). These policies are available in the University Catalog 2016-2017, General Academic Regulations <http://bit.ly/2asrlZN>

The penalties for plagiarism and cheating are severe and range from receiving a grade of zero on an assignment, to academic suspension. For more information on Cheating, Plagiarism, and Fraud: <http://bit.ly/2b63fBP> & <http://bit.ly/2b63ywR>

All work is to be completed independently unless otherwise specified.

Schedule "A"

Schedule "A" is the new requirement passed by the University of Manitoba Senate in 2016 that requires that a mandated list of supports for services plus contact information is provided to students. While this information is important and useful, should you require advice and support, the instructor is happy to meet with you and discuss issues and concerns

Student Resources

Writing and Study Skills Support

The Academic Learning Centre (ALC) offers services that may be helpful to you as you fulfill the requirements for this course. Through the ALC, you may meet with a study skills specialist to discuss concerns such as time management, reading and note-taking strategies, and test-taking strategies. You may also meet one-on-one with a writing tutor who can give you feedback at any stage of the writing process, whether you are just beginning to work on a written assignment or already have a draft. Writing tutors can also give you feedback if you submit a draft of your paper online. (Please note that the online tutors require 48 hours, from Mondays to Fridays, to return your paper with comments.)

All Academic Learning Centre services are free for U of M students. For more information, please visit the Academic Learning Centre website at: umanitoba.ca/student/academiclearning/

You can also talk to a member of the Academic Learning staff by calling 480-1481 or by dropping in at 201 Tier Building.

University of Manitoba Libraries (UML)

As the primary contact for all research needs, your liaison librarian can play a vital role when completing academic papers and assignments. Liaisons can answer questions about managing citations, or locating appropriate resources, and will address any other concerns you may have, regarding the research process. Liaisons can be contacted by email or phone, and are also available to meet with you in-person. A complete list of liaison librarians can be found by subject: <http://bit.ly/WcEbA1> or name: <http://bit.ly/1tJ0bB4>. In addition, general library assistance is provided in person at 19 University Libraries, located on both the Fort Garry and Bannatyne campuses, as well as in many Winnipeg hospitals. For a listing of all libraries, please

consult the following: <http://bit.ly/1sXe6RA>. When working remotely, students can also receive help online, via the Ask-a- Librarian chat found on the Libraries' homepage: www.umanitoba.ca/libraries.

The English Language Centre

The English Language Centre has workshop and programs in advanced academic and health-sciences English (located at 520, University Centre).

Homepage: <http://umanitoba.ca/student/staffdir/elc.html>

Student Accessibility Services

Student Accessibility Services (SAS) provides support and advocacy for students with disabilities of all kinds: hearing, learning, injury-related, mental health, medical, physical or visual. Students with temporary disabilities such as sprains and breaks are also eligible to use our services. SAS acts as a liaison between students and the faculty and staff of the University of Manitoba as well as support agencies within the province of Manitoba. Please phone: 474-6213 (voice) or 474-9690 (TTY) for service. For more information, please visit the Student Accessibility website at: <http://umanitoba.ca/student/saa/accessibility/>

Mental Health

For 24/7 mental health support, contact the Mobile Crisis Service at 204-940-1781.

Student Counselling Centre (SCC)

Contact SCC if you are concerned about any aspect of your mental health, including anxiety, stress, or depression, or for help with relationships or other life concerns. SCC offers crisis services as well as individual, couple, and group counselling. Student Counselling Centre: <http://umanitoba.ca/student/counselling/index.html>

474 University Centre or S207 Medical Services (204) 474-8592

Student Support Case Management (SSCM)

Contact the Student Support Case Management team if you are concerned about yourself or another student and don't know where to turn. SSCM helps connect students with on and off campus resources, provides safety planning, and offers other supports, including consultation, educational workshops, and referral to the STATIS threat assessment team. Student Support Intake Assistant <http://umanitoba.ca/student/case-manager/index.html>

520 University Centre (204) 474-7423

University Health Service

Contact UHS for any medical concerns, including mental health problems. UHS offers a full range of medical services to students, including psychiatric consultation. University Health Service <http://umanitoba.ca/student/health/>

104 University Centre, Fort Garry Campus (204) 474-8411 (Business hours or after hours/urgent calls)

Health and Wellness

Contact our Health and Wellness Educator if you are interested in information on a broad range of health topics, including physical and mental health concerns, alcohol and substance use harms, and sexual assault.

Health and Wellness Educator <http://umanitoba.ca/student/health-wellness/welcome.html>
Katie.Kutryk@umanitoba.ca

469 University Centre (204) 295-9032

Live Well @ UofM

For comprehensive information about the full range of health and wellness resources available on campus, visit the Live Well @ UofM site: <http://umanitoba.ca/student/livewell/index.html>

Copyright

All students are required to respect copyright as per Canada's Copyright Act. Staff and students play a key role in the University's copyright compliance as we balance user rights for educational purposes with the rights of content creators from around the world. The Copyright Office provides copyright resources and support for all members of the University of Manitoba

Visit <http://umanitoba.ca/copyright> for more information.

University and Unit Policies, Procedures, and Supplemental Information

Your rights and responsibilities

As a student of the University of Manitoba you have rights and responsibilities. It is important for you to know what you can expect from the University as a student and to understand what the University expects from you. Become familiar with the policies and procedures of the University and the regulations that are specific to your faculty, college or school.

The Academic Calendar

<http://umanitoba.ca/student/records/academiccalendar.html> is one important source of information. View the sections University Policies and Procedures and General Academic Regulations. While all of the information contained in these two sections is important, the following information is highlighted.

If you have questions about your grades, talk to your instructor. There is a process for term work and final grade appeals. Note that you have the right to access your final examination scripts. See the Registrar's Office website for more information including appeal deadline dates and the appeal form <http://umanitoba.ca/registrar/>

You are expected to view the General Academic Regulation section within the Academic Calendar and specifically read the Academic Integrity regulation. Consult the course syllabus or ask your instructor for additional information about demonstrating academic integrity in your academic work. Visit the Academic Integrity Site for tools and support <http://umanitoba.ca/academicintegrity/> View the Student Academic Misconduct procedure for more information.

Sexual Assault

- If you experience Sexual Assault or know a member of the University community who has, it is important to know there is a policy that provides information about the supports available to those who disclose and outlines a process for reporting.
- The Sexual Assault policy may be found at:
http://umanitoba.ca/admin/governance/governing_documents/community/230.html
- More information and resources can be found by reviewing the Sexual Assault site
<http://umanitoba.ca/student/sexual-assault/>
- For information about rights and responsibilities regarding Intellectual Property view the policy
http://umanitoba.ca/admin/governance/media/Intellectual_Property_Policy_-_2013_10_01.pdf

Academic Programs and Regulations

- For information on regulations that are specific to your academic program, read the section in the Academic Calendar and on the respective faculty/college/school web site
<http://umanitoba.ca/faculties/>
- Contact an Academic Advisor within our faculty/college or school for questions about your academic program and regulations <http://umanitoba.ca/academic-advisors/>

Student Advocacy

Contact Student Advocacy if you want to know more about your rights and responsibilities as a student, have questions about policies and procedures, and/or want support in dealing with academic or discipline concerns.

<http://umanitoba.ca/student/advocacy/>

520 University Centre 204 474 7423 student_advocacy@umanitoba.ca