McGill

Faculty of Science

COMP 208

Computer Programming for Physical Sciences and Engineering

Course Outline
Fall 2025

Sec. 001, Tues/Thurs 14:35-15:55, SADB M-1
Instructor: Prof. Michael Langer

Sec. 002, Mon/Wed 11:35-12:55, STBIO S1/3
Instructor: Dr. Chad Zammar

1 Overview

Welcome to COMP 208! Please read this document carefully and refer to it when you have questions about course policy.

This course is an introduction to computer programming for those with no background in the subject. We will use the Python programming language, which is one of the most popular languages around. The course will teach you to program in Python, and it will also teach you foundational programming skills that can be transferred to other popular languages such as Java, C/C++.

Learning how to program is not easy. It is not a set of facts that you can memorize. A computer program is a sequence of instructions that tells a computer how to perform a task. To write computer programs, you do know the words and "grammar" of the programming language. But you also need to practice how to put the words together to solve your problems. For example, sometimes you have to break down larger problems into smaller problems, solve each of these smaller problems, and put the parts together. This course will teach you how to do this.

COMP 208 is very similar to two other introductory progamming courses offered by the School of Computer Science, namely COMP 202 and COMP 204. The main differences are that COMP 208 requires you to have a slightly stronger math background than 202 does,

as 202 can be taken by any McGill student. COMP 204 is similar, but it is meant for Life Science students. The three courses have essentially the same lecture content, except for a few topics. The courses also have different assignments, tuned for the particular audiences.

1.1 Learning Objectives

By the end of this course, you will be able to:

- Describe instructions that can be used by a computer to solve a problem or perform a task. What are the steps and parts of the problem or task? What are the inputs and outputs? What test cases that could be used to check if the solution is correct?
- Express these instructions in the Python programming language. What are the appropriate data types or data structures to use? What libraries should or could be used?
- Apply style conventions to make programs easy to understand and modify.
- Debug programs, tracing the program's execution and detecting and correcting errors.
- Learn independently about programming language features and libraries by reading documentation and by experimenting.

1.2 Course prerequisites

There are no prerequisites for this course, but there are two co-requisites. (A co-requisite is a course that you are taking in the current term or have taken in a prior term.)

- MATH 133 Linear Algebra and Geometry
- MATH 141 Calculus 2

These co-requisites will come up for a few of the topics that we will cover in the lectures and also in the programming assignments.

2 Course Topics and Schedule

#	Topics	Assessments
0	Course introduction	
1	binary numbers binary representations in computers (text, numbers, etc) Thonny: shell vs editor	
2	basic Python: values and types, comments, printing, strings, variables, assignment statements, input prompt, naming rules and conventions objects: references, swapping, garbage collection	
3	mathematical, relational and logical operators expressions	
4	conditional statements (if-then-else) functions 1: header, parameter, body nesting and flow of execution `math` module	Quiz 1
5	functions 2: non-void, return statements scope: local vs global variables docstrings `random` module	A1 posted
6	while loops bug types (syntax, runtime, logic), infinite loops debugging, raising exceptions, doctest	Quiz 2
7	for loops, `range, break, continue`	
8	loop examples: numerical methods for sqrt, integration, root finding	Quiz 3
9	strings: indexing and slicing, methods, traversal, f-strings	A1 due
10	lists: indexing and slicing, methods, traversal	Quiz 4 A2 posted
11	list algorithms: search and sort	

#	Topics	Assessments
12	object identity, `id` vs. `==` optimization and garbage collection lists and strings (mutable vs immutable)	
	Fall Reading Break (Oct. 14-18)	
13	nested lists shallow vs deep copies list comprehensions	Midterm A2 due
14	file I/0: read, write, csv format Matplotlib	Quiz 5
15	tuples, sets	A3 posted
16	dictionaries, `enumerate`	Quiz 6
17	modules and import documentation, help, libraries	
18	OOP 1: objects and classes turtles	Quiz 7
19	classes, instantionating objects, attributes, `__init__`, `self`	A3 due
20	OOP 3: instance methods, `__str__`, `__eq__` shallow and deep copy	Quiz 8 A4 posted
21	`numpy` 1: vectorization, broadcasting	
22	`numpy` 2: boolean indexing, image analysis	Quiz 9
23	`numpy` 3: random walks, Monte Carlo integration	
24	Final Exam Review 1	
25	Final Exam Review 2	A4 due

3 Course Design and Materials

3.1 Video Lectures and Lecture Slides

We will use an flipped classroom approach. Rather than introducing you to the course material with live lectures, we will instead provide video lectures which you are expected to watch prior to each scheduled class. These videos are recorded lectures from the Fall 2024 semester which have been edited and shortened – each is typically around 60 min rather than 80 minutes – by removing the Q&A. The video(s) and the corresponding slides for each class will made available to you well in advance.

3.2 In class Activities

The in-class activities will begin with a summary of the material from the videos to refresh you. We will also take a few minutes to address any critical issues.

The rest of the class will be devoted to exercises where you will work through examples, either individually or in small groups. This may involve writing code on a laptop or writing code on paper/tablet.

The classtime activities will not be recorded, but the exercises themselves will be made available in PDF form and/or with solution code after class, for those who did not attend.

We may also provide exercises that go beyond what was covered in the in-class activities, and example solutions to these exercises will also be posted. Note that your solutions may differ from the posted solutions which in some cases may be fine since often there is more than one way to solve a problem.

3.3 Other Resources

There is no textbook for this course, but you may find the following useful as a reference:

- *Think Python 2e*, by Allen B. Downey. It is available here at no cost under the terms of the Creative Commons Attribution-NonCommercial 3.0 Unported License.

There are many excellent web sites for learning Python. Here are a few examples:

- W3 Schools Python tutorial
- Real Python (use the search bar)

3.4 Required software

Programmers typically use an integrated development environment (IDE) to write computer programs. IDEs provide an editor that allows you to type in and edit your programs,

commands to run your program, and many other useful tools.

We will use an IDE called Thonny which you can download from https://thonny.org. Thonny supports Mac, Windows and Linux, and comes with Python 3.10 built in. This means you only have one thing to install and you are ready to learn how to program!

There are many other IDEs out there such as Spyder or IDLE. You are welcome to use these instead. However, the teaching staff may not be familiar with your IDE and therefore might not be of as much help with any issues stemming from that editor.

3.5 Computer Lab in Trottier

All students registered in COMP 208 may use the School of Computer Science (SOCS) computer laboratory facilities, which are located on the third floor of the Trottier building. You can refer to here for more information.

4 Communication, Getting Help

4.1 Office hours

The instructors and the Teaching Assistants (T.A.s) will be available for office hours each week to help you with your assignments and answer questions about the course material. Office hours will be held both in person and through Zoom and will be individual (one-on-one). The office hour schedule will be shared with you on mycourses.

The Computer Science Undergraduate Society (CSUS) also offers a Help Desk for peer-to-peer tutoring.

4.2 Ed Discussion

4.2.1 Course announcements

Important course information will be announced on Ed Discussion. *You are expected to monitor these Announcements.* You should also monitor your McGill email as sometime announcements are sent via email as well.

The Ed Discussion link is available from the COMP 208 mycourses website.

4.2.2 Discussion board

Please also use Ed Discussion to post all your questions related to the course content and the assignments on Ed. Do not email the teaching staff with such questions.

Ed Discussion allows you to post privately, if you are unsure that your post should be read by other students e.g. because it might give away a solution to an assignment. It also allows you to post anonymously if you do not wish to be identified to the other students (but note that instructors and T.A.'s can see who posted).

We encourage you to answer other students' questions, with one important exception: do not provide tips for solving problems on an assessment that is meant to be done on your own (quiz or assignment). Leave that to the TAs and instructor.

When posting on Ed Discussion, please follow these guidelines. Postings that do not conform may be edited or deleted.

- Use the search feature to see if your question has been asked before. Duplicate posts will be flagged and/or deleted.
- Choose a suitable subject line, so that others know what the post is about, for example, "Exercises 3 Question 2".
- Choose the appropriate Topic for your question (Assignment, Exercises, etc).
- If you have multiple questions that are unrelated to each other, then use multiple postings so that the threads do not become entangled.
- Proofread before posting. Your posting may be read by dozens of others, so please take an extra minute to ensure that what you write makes sense.
- Be polite and respectful.

4.3 Email policy

Please only email the instructors for private matters, and do so only if you don't want the TA's to see the correspondence. If you don't mind the TA's seeing the message, then please just make a private post on Ed Discussion. Keeping corresondences on Ed makes it easier for us to keep track.

If you do send email to the instructors, then please use comp208.cs@mcgill.ca and include 'COMP 208' in the subject header. Be sure to send this email from your `@mail.mcgill.ca` address, and include your student ID at the bottom of your email. Email sent from non-McGill accounts may be spam filtered or simply ignored.

When emailing an instructor, please follow the guidelines on etiquette described in the video here and on this website.

It should not be necessary to email a T.A. but if you do then please cc the instructors at the above address.

4.4 Personal Tutoring

Several undergraduates who have taken COMP 208 in previous years are available for personal tutoring – no charge! Details will be announced in the first lecture.

4.5 Tutorials

The course has several "tutorial" time slots and rooms. We will use these slots on a *ad hoc* basis, for tutorials, help with assignments or exercises, or review for exams, etc. This will be determined both by the available of the T.A.s and the demand for help.

5 Method of Evaluation (Grades)

Your course grade will be calculated as follows:

- **Quizzes:** 10%
- **Assignments:** 28% (4×7 % each)
- **Midterm Exam** 22 %
- **Final Exam:** 40%

If you miss the Midterm Exam or you do worse on the Midterm Exam than on the Final Exam, then your Midterm grade will automatically be replaced with your Final Exam grade.

We will calculate your final course grade according to the above, and we will use a formula that rounds off to the nearest integer. If your grade is 84.4999 then it rounds to 84 and you get an A-, whereas if it is 84.5 then it rounds to 85 and you get an A. The same round off procedure holds for low grades. If your calculated final course grade is 49.49999 then it rounds to 49 which would be an F. To avoid a slippery slope, we will draw a hard line on this. So if you don't want to fail, then do the work to ensure you stay far above that line.

5.1 Quizzes

We will have nine Quizzes, four before the Fall study break and five afterward. We count the best eight of them, so each will be worth 1.25% on your final grade.

If you do not do a quiz, then you will receive a grade of 0 on it. The only exceptions are documented medical reasons or serious extenuating circumstances such as a death in the family.

The quizzes will be on mycourses. Each quiz will be posted by Thursday PM and the questions will include material from that week. The due date is the following Wednesday at midnight, but we strongly encourage you to do the quiz by end of the weekend so you are caught up for class that week.

5.2 Assignments

There will be four Python programming assignments. Each assignment will be worth 7% of your final grade. Here are the approximate dates on which the assignments will be posted:

- Assignment 1 to be posted in mid-September
- Assignment 2 to be posted in early October
- Assignment 3 to be posted in late October
- Assignment 4 to be posted in mid-November

You will be given two weeks[1] to complete each assignment.

Assignments are to be done individually and **must** represent your own work. See section 7.1 on Plagiarism Policy and Assignments below for further clarification.

5.2.1 Submissions and grading

Assignments will be submitted and graded using Ed Lessons. You are responsible for verifying that your submissions are successful.

Assignment grades will be visible on Ed Lessons once the assignments are graded, but these grades will only be visible on myCourses towards the end of the term. It is your responsibility to check that your grades are correct and to notify your instructor of any errors.

5.2.2 Policy on Missed or Late Assignments

If you do not do an assignment, then you will receive a grade of 0 on it. Same as the quizzes – the only exceptions are documented serious medical reasons or other unforeseen circumstances such as a death in the family.

If you need to submit the assignment late because of a minor illness or other personal reasons, then you will be given a late penalty. This policy is to be fair to students who do submit on time. Details on the policy will be stated on the Assignment PDFs.

We can waive the late penalty in cases of extended illness or other unforeseen personal circumstances. However, you must make a formal request either with a private post on Ed Discussion or email (see section 4.3).

Examples of *invalid* requests for an extension are:

- Your laptop broke. This is invalid because there are computers in the Trottier building that you can use.

[1]Some of the assignments may come in two parts, with the second part depending on material presented in the first half of the two week interval. This will still leave you one week for the second part.

- You lost your files e.g. because your laptop broke. This is invalid because you should use a cloud backup system, e.g. OneDrive, Dropbox, Google Drive, etc.
- You have other exams, a job interview, etc. This is invalid because these events are not unexpected and you have two weeks to complete your assignments. It is your responsibility to plan accordingly.

5.3 Exams

Both the midterm and final exam will be in-person. Each will consist of a set of short answer questions and a set of longer answer questions. The short answer questions typically ask you to evaluate a small set of Python instructions, or identify errors in code. The long answer questions typically ask you to write a Python code that solves a given problem in a given way.

These exams will be written with pencil and paper. Therefore you should practice doing exercises with pencil and paper through the semester to prepare for the exams. Some of the in-class activities will use pencil and paper to prepare you.

5.3.1 Midterm Exam

The midterm examination will take place on Oct. 21 from 6:30 PM - 8:30 PM. This is the first Tuesday after Reading Break. The room location for each student will be provided the week before the exam.

There will be *no make-up midterm*. If you miss the midterm exam for whatever reason, then your midterm grade will be replaced by your final exam grade.

5.3.2 Final Exam

The final exam will take place in-person during the final exam period.

5.3.3 Supplemental/Deferred exam

If you get a D or F grade in the course (but not if you get C or higher), then you may be eligible to write the Supplemental Exam. If you miss the Final Exam (J grade), then you may be eligible to write the Deferred Exam. The Supplemental/Deferred Exams will take place during Winter Term Reading Week. For official McGill policy info on Supplemental and Deferred exams, see here.

The Supplemental/Deferred exam will cover the same material as the Final Exam. For those writing the Supplemental Exam, a new grade will appear on your transcript which is calculated as follows: the Supplemental exam grade will replace the Final Exam grade, and your pre-existing midterm and assignment grades will be used as before (and hence double

counted in the GPA). For those writing the Deferred Exam, the exam grade will simply replace the component that would have been the Final Exam grade.

5.4 Regrade Requests

If you believe that you have been assigned an incorrect grade for one of your in-term assessments (quiz, assignment, midterm) then you can request a regrade. Only well-argumented requests will be considered. *You may request this regrade within 7 days from when your grade was published.* Please use Ed Discussion (private post) for a regrade request.

5.5 Additional work

Students who receive unsatisfactory final grades will **NOT** have the option to submit additional work to improve their grades.

6 Course Workload

McGill's policy on course workload is that there should be on average 3 hours of work per week per credit. COMP 208 is a 3 credit course and so we target 9 hours of work per week on average. Over a 13 week semester, this amounts to about 120 hours. This target workload is as follows. Note there are 13 weeks to the semester (26 class slots), not including Thanksgiving Reading Week.

- Video lectures ($\sim$2 hours per week = $\sim$25 hours total)

  The video's themselves on average will be about 1 hour per lecture (2 hours per week).

- In-class activities ($\sim$3 hours per week = $\sim$40 hours total)

  Each of the two sections 001 and 002 has two 80 minute "lecture" slots per week. Our plan is:

  - review what the videos covered for this lecture and address any common questions;
  - practice writing code with "pencil and paper" (or tablet) which will prepare you for the midterm and final exams;
  - practice programming exercises on your computer;
  - answer practice quiz questions

  The activities will in some cases give you the opportunity to meet and work with other students in the class.

  The activities will *not* be recorded, but the exercises covered there and any slides used in class will be made available on the mycourses after class.

- Study and extra home exercises (∼1 hour per week = ∼10 hours total)

  You will likely need to supplement the in-class activities with more work e.g. if we don't cover all the exercises planned.

- Quizzes (9 quizzes, each should take 30-60 min to do = ∼10 hours total

- Assignments (4 × ∼9 hours each = ∼35 hours)

  This is a rough estimate for an "average" McGill student with no background in programming and who will receives an average grade (about a B). If you have never programmed before and you would like a course grade higher than B, then you likely will need to spend more than 10 hour per assignment.

The workload does not include the time for writing the midterm. It also does not include time in December after classes are over when you will study for the final exam.

7 Policies on Academic Integrity

McGill University values academic integrity. All students must understand the meaning and consequences of cheating, plagiarism, and other academic offences under the Code of Student Conduct and Disciplinary Procedures. See www.mcgill.ca/integrity/ for more information.

7.1 Plagiarism on Assignments

We encourage you to discuss assignment questions with each other as long as each person does the assignment on their own. However, there is sometimes a gray zone here, so be careful! You should not work together (collaborate) on the assignments. You should also not show your solution code to each other, unless you are trying to get help when debugging. A good rule of thumb is that any discussion between students about an assignment should be verbal only and you should not leave the discussion with written notes. The programs you write should be your own work (not jointly owned/produced). And *sharing code is absolutely forbidden!*

It is also forbidden to use generative AI tools such as chatGPT or Gemini, etc, to generate assignment solution code. You may certainly use such tools outside the context of the assignments, e.g. similar to how you would use a search engine or calculator. But for writing and debugging the assignment code, you must do this on your own since that is how you will learn to program!

Include your name and McGill ID number at the top of each code file that you implement and submit. By doing so, you are certifying that the program is entirely your own, and represents only the result of your own efforts.

7.2 Plagiarism detection

We will use code similarity detection software to compare your assignment submissions to those of all other students registered in the course. However, we will not accuse anyone of copying or working in groups based solely on the output of these tools. Rather, we will use these tools to determine which submissions should be manually checked for similarity by the TA and instructor.

When an instructor suspects that plagiarism has occurred, the instructor will report the case to the Disciplinary Officer in the student's Faculty (Science, Arts, Engineering, etc). For more details on the process, see Section III Articles A.37 (p. 10) and A.48 (p. 13) of the Code of Student Conduct and Disciplinary Procedures which can be found here.

7.3 Posting assignment solutions on a website

You must not post your assignments on a public space such as GitHub, CourseHero, etc. This rule extends beyond the duration of the course. The reason for the rule is that instructors occasionally recycle assignments from previous years, and if the old versions are easily accessible then such posting can lead to plagiarism by others.

8 Heath and Wellness

8.1 Accommodations

If you require accommodations, the Accessibility and Achievement Office is there to help. This Office liaises with your instructor on your behalf to ensure that your accommodations are met and that you can succeed in your studies. Note that this office replaces the Office of Students with Disabilities.

Students who are pregnant and/or caring for a dependent may find it helpful to receive academic accommodations. McGill's guidelines for accommodations for students who are pregnant and/or caring for a dependent may be found here.

8.2 Student Wellness Hub

If you need to access services for health and wellness, or if you want to get more information, please consult our Student Wellness Hub at here, or drop by the Brown Student Services Building (downtown) or Centennial Centre (Macdonald Campus). Within your faculty, you can also connect with your Local Wellness Advisor (to make an appointment, visit here).

9 Other Policies

9.1 McGill Language Policy

In accordance with McGill University's Charter of Students' Rights, students in this course have the right to submit in English or in French any written work that is to be graded.

Conformément à la Charte des droits de l'étudiant de l'Université McGill, chaque étudiant a le droit de soumettre en français ou en anglais tout travail écrit devant être noté.

9.2 Extraordinary Circumstances beyond the University's Control

In accordance with section 3.2.3 of the *University Student Assessment Policy*, in the event of extraordinary circumstances beyond the University's control, the evaluation scheme in a Course is subject to change, provided that there be timely communications to the students regarding the change.

9.3 Copyright policy for course materials

You are not allowed to post any course materials online, including but not limited to GitHub, Course Hero, or any other websites. Course materials include video recordings, PDFs of lecture slides, tutorials, exercises, assignment questions and solutions, or anything else provided to you during the course.

Stated more formally: Instructor/TA-generated course materials are protected by law and may not be copied or distributed in any form or in any medium without explicit permission of the instructor/TA. Infringements of copyright can be subject to follow up by the University under the Code of Student Conduct and Disciplinary Procedures.

10 Land acknowledgement

McGill University is on land which has long served as a site of meeting and exchange amongst Indigenous peoples, including the Haudenosaunee and Anishinabeg nations. We acknowledge and thank the diverse Indigenous people whose footsteps have marked this territory on which people of the world now gather. Please see here for more details.