McGill

Faculty of Science

COMP 208

Computer Programming for Physical Sciences and Engineering

Course Outline
Fall 2023

Instructors:
Jonathan Campbell, jonathan.campbell@mcgill.ca
Sec. 001, WF 2:35-3:55, ENGTR 0100

Michael Langer, michael.langer@mcgill.ca
Sec. 002, MW 10:05-11:25, MDHAR G-10

Contents

1 Overview

Welcome to COMP 208! Please read this document carefully and keep it for reference throughout the term.

This course introduces you to computer programming and is designed for those with little or no background in the subject. No knowledge of computer science in general is necessary or expected, although of course basic computer skills such as searching the Web, sending email and other such basic tasks will be assumed.

The course uses the Python programming language, which is one of the most popular languages (along with Java, C++, and many others). A large part of this course will focus on the basic building blocks of programming, which provide the foundations to learning other languages such as Java or C++.

Learning how to program is not easy; it is not a set of facts that one can simply memorize. In principle, a computer program is simply a set of instructions that tells a computer to perform a task. However, finding the right set of instructions can be quite challenging. For that, one has to learn how to structure a larger problem into small subsets, and then find the solution to each particular subset. This course aims to teach you a way of thinking that will enable you to build non-trivial programs.

1.1 Primary Learning Objectives

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

- Design and describe instructions that can be used by a computer to solve a problem or perform a task.
- Translate these instructions into a programming language that a computer can understand (Python).
- Write Python programs that solve complex problems by decomposing them into simpler sub-problems.
- Apply programming style and structure conventions to make programs easy to understand, debug and modify.
- Learn independently about new programming language features and libraries by reading documentation and by experimenting.

- Solve basic problems in scientific computing, including numerical methods such as root finding, numerical integration, and systems of linear equations.

1.2 What This Course is Not About

This course is not about how to use a computer. It will not teach you how to send email, browse the Web, create word processing documents or spreadsheets, setup and configure a computer, use specific software applications (except those needed to complete coursework), design Web pages, or deal with operating system or hardware problems.

1.3 Course prerequisites

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

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

These co-requisites will come up for a few of the topics that focus on Physical Sciences and Engineering, including derivation, integration, and basic linear algebra.

2 Course materials

2.1 Recordings and slides

Lecture recordings and slides for both sections 001 and 002 will be made available to all students on myCourses. The slides may differ a bit between the two sections, but the material covered in each lecture will be the same.

2.2 Recommended exercises

Sets of exercises will be posted during most weeks to allow you to practice what you learn in the lectures and to go beyond the examples given in the lectures.

2.3 Textbook and other references

There is no required textbook for this course. However, you may choose to consult the following:

- *Think Python 2e*, by Allen B. Downey. Available here at no cost under the terms of the Creative Commons Attribution-NonCommercial 3.0 Unported License .
- *Numerical methods in engineering with Python 3*, by Jaan Kiusalaas. PDF copy available to McGill students here.

Other resources will be made available or suggested during the term.

2.4 Copyright policy

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 but are not limited to video recordings, PDFs of lecture slides, tutorials, recommended exercises, assignment questions, 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.

3 Communication policies

3.1 Office hours

Teaching Assistants (T.A.s), TEAM Mentors and the instructors 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 in the first weeks of the term. Feel free to attend as many of these office hours as you like, whichever best fit with your schedule.

3.2 Ed Discussion board

We will be using Ed Discussions for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TAs/Mentors and instructors. *Please post all your questions related to the course content and the assignments on Ed, rather than emailing questions to the teaching staff.* By using the Discussion Board, you will receive an answer faster, and everyone in the class will be able to benefit from it.

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 the assignment. It also allows you to post anonymously if you do not wish to be identified for some reason.

You may freely answer other students' questions as well, with one important exception: you may not provide solution code (although you are permitted to provide one or two lines of code to illustrate a point).

You can access Ed Discussions through a link on our class' myCourses navigation bar.

3.2.1 Discussion board guidelines

We encourage you to share your knowledge and help each other out by answering your peers' questions on the discussion board. The instructors and TAs/Mentors will try to moderate the discussion board, but it works best when students help each other out.

When posting to the discussion board, please obey the following guidelines. *Postings that do not conform may be deleted.*

- Choose the appropriate topic for your question.
- Use the search feature to see if your question has been asked before. Do **not** make duplicate posts.
- Choose a suitable subject line, so that others know what the post is about.
- If you have multiple questions that are unrelated to each other, then use multiple postings so people can follow the thread.
- Proofread before posting. Take an extra minute to ensure that what you write makes sense.
- If you would like your post to be deleted, just add a request within the thread.
- Be polite and respectful.

Formally and officially: The University is committed to maintaining teaching and learning spaces that are respectful and inclusive for all. To this end, offensive, violent, or harmful language arising in contexts such as the following may be cause for disciplinary action:

- Zoom sessions, including username (use only your legal or preferred name), virtual backgrounds, 'chat' boxes, whiteboard annotations, breakout rooms.
- Ed Discussion board.
- Other in-course venues of discussion.

3.3 Contacting instructors and teaching assistants

For private matters, you can send email to a TA or to your instructor(s) directly with 'COMP 208' in the subject header. For instructors, please use cs208-prof@cs.mcgill.ca. Be sure to send your email from your `@mail.mcgill.ca` address, and include your student ID. Email sent from non-McGill accounts may be spam filtered or simply ignored.

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

Here are some examples of private matters that may necessitate you to email your instructor:

- requesting an assignment extension in limited circumstances (see section 4.1.2)
- if you are anxious about your performance in the class and would like to discuss it

3.4 Course announcements

Important course information will be announced in class and/or on myCourses and Ed Discussion. Please subscribe now to myCourses Announcements if you have not already done so. *You are expected to monitor your McGill email, myCourses, and Ed discussions for course-related news and information.*

4 Methods of Evaluation & Grades

We offer you the choice between two alternative grading schemes. You must choose either Option A or Option B by September 13. You will indicate your choice through a form that will be provided on myCourses. Once you have made your choice, it **will not** be possible to switch.

Option A

- **Assignments:** 35%
- **Midterm:** 15%
- **Final Exam:** 50%

Option B

- **Assignments:** 35%
- **Midterm:** 15%
- **Final Exam:** 35%
- **Group work:** 15%

We will calculate your final course grade according to the above percentages, and we will use a formula that rounds off to the nearest integer. If your grade is 84.4 then it rounds to 84 and you get an A-, whereas if it is 84.6 then it rounds to 85 and you get an A. If your grade is 84.5, our formula will round it up to 85. The same round off procedure holds for low grades. If your calculated final course grade is 49.4 then it rounds to 49 which is an F. We draw a very hard line on this, so if you don't want to fail then you should stay far away from that line.

4.1 Assignments

There will be four assignments consisting of writing Python programs. It is *very important* that you complete all assignments, as doing so is the best way to learn the material. By working hard on the assignments, you will gain essential experience needed to solve programming problems. The first assignment will be worth 8% of your final grade, while the others will be worth 9% each. Here are the approximate dates on which the assignments will be posted:

- Assignment 1 to be posted on or around September 20
- Assignment 2 to be posted on or around October 4
- Assignment 3 to be posted on or around November 6
- Assignment 4 to be posted on or around November 20

You will be given approximately two weeks to complete each assignment. The deadline will be specified on the assignment PDF.

Assignments (as well as all other course work) **MUST** represent your own personal efforts (see section 5.1 on Plagiarism Policy and Assignments below).

If you do not do an assignment, then you will receive a grade of 0 for it. No exceptions.

4.1.1 Assignment submissions

Assignment submission will take place through codePost. You will be added on codePost before the first assignment is released, and you will be instructed on how to use codePost. You are responsible for verifying that your submissions are successful. If you believe the content of your submission is different from what you have submitted, you must email your instructor preferably within a few days of the assignment deadline and provide evidence of your correct submission.

Assignment grades will be visible on codePost once the assignments are graded, and then 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.

4.1.2 Late Policy

Unforeseen events may arise that prevent you from submitting an assignment on time. For example, you might be sick for several days in the week before the assignment is due. Our standard late policy is that you may submit up to two days after the deadline, but with a small penalty: we will deducted 10% each day for which they are late, including weekends and holidays; that is, assignments that are between 0 and 24 hours late will be deducted 10 points, and assignments that are between 24 and 48 hours late will be deducted 20 points. We are willing to waive this penalty in cases of *extended* illness or other unforeseen personal circumstances. However, you must make a formal request (see section 3.3 for email policy).

Examples of invalid requests are:

- Your laptop broke or was stolen. This is not a valid excuse. You should be using a cloud backup system, e.g., Dropbox, Google Drive, etc., to be able to access a version of your file.
- You have other exams, a job interview, etc. These are invalid because they are not unexpected and you have two weeks to complete your assignments. It is your responsibility to plan accordingly.

Assignments submitted more than 2 days after the deadline will not be accepted, nor graded, and will therefore receive a grade of 0.

The instructors reserve the right to modify the lateness policy for a particular assignment; any such modifications will be clearly indicated at the beginning of the relevant assignment specifications. *Plan appropriately and do* ***not*** *submit to codePost only minutes before the assignment deadline. Requests for waiving the late penalty because the system was busy or your machine/internet was too slow will not be accepted.* Take care: programming assignments are notoriously time-consuming and individual exceptions to the lateness policy will not be granted without appropriate justification submitted in writing and supported by documentary evidence.

4.1.3 Assignment Review Meetings (Peer Code Review)

In the week following the due date for each assignment, you will be asked to meet briefly with a TA and a small number of other students. In the meeting, the TA will present selections from your assignment submissions, and discuss these solutions with you as a group. You may be asked to discuss portions of your code or another student's code during these meeting. These discussions will *not* be used to determine your grade for the assignment – the code itself determines the grade. However, if you are not able to explain your own code then the TA may consider this as evidence of plagiarism and investigate further. (See Sec. 5.1.)

Details about how to schedule a meeting with the TA will be shared with you after the due date of an assignment. *If you do not schedule a meeting or you do not attend a meeting that was scheduled, then your assignment will not be graded and you will receive a zero.*

We take a hard line on these code reviews because we believe they are valuable. Code reviews prepare you to deal with criticism and they can teach you to provide constructive criticism to others [1]. Even if your professional career does not involve programming, these 'soft' or 'people' skills are likely to be important.

Peer code reviews also promote opportunities for deeper learning and critical thinking. They provide you with learning opportunities at a level that traditional teaching approaches do not often address. [2] That is, although we discuss the basics of programming during lecture, we often do not have time to analyze a particular piece of code and explain what is desirable about it and/or how to improve it. Peer review can help to fill in this missing part of the foundations of programming and has been shown to be successful in introductory computer science classes [3].

4.2 Midterm Exam

The midterm examination will take place in-person on the evening of Monday, October 30. Room location(s) for each student will be provided the week before the exam.

4.3 Final Exam

The final exam will take place in-person during the final exam period. It will be worth 50% for students who select Option A and 35% for students who select Option B. Note that it is **not** possible to change to a different Option after having selected one.

4.4 Supplemental/deferred exam

Under certain conditions, you may be eligible to write the Supplemental/Deferred exam which takes place during Winter Term Reading Week. This exam will cover the same material as the Final Exam. For those writing the Deferred Exam, the exam grade will simply replace the Final Exam grade. 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 (and hence double counted in the GPA). For more information on Supplemental and Deferred exams, see here.

[1] Anewalt, K. Using peer review as a vehicle for communication skill development and active learning. J. Computing Sciences in Colleges, 21, 2, 2005, 148-155.

[2] Gehringer, E.F., Chinn, D., Mardis, M. and Perez-Quinones, M. Panel: Using peer review in teaching computing. In Proc. 36th SIGCSE Technical Symposium, ACM Press, New York, 2005, 321-322.

[3] Petersen, A. and Zingaro, D. Code reviews in large, first-year courses. In Proc. 23rd ITiCSE Technical Symposium, ACM Press, New York, 2018, 354-355.

4.5 Group work (grading Option B only)

To encourage community building and enhance your learning experience, at the beginning of the semester, Option B students will be randomly divided into groups of 16-25 with whom they will regularly interact throughout the term. Each group will be further sub-divided into four smaller groups, each containing 4-6 students. These smaller groups are the groups within which you will be working more closely.

Each group of 16-25 students will be assigned a number. For instance, group 7 might contain 18 students, divided into four subgroups.

7A	7B	7C	7D
Shoaib Fei Louise Miguel	Janet Breanna Xin Samuel	Emma Teri Miles Yuxuan Mahir	Cassia Maxime Yue Clinton Hugo

Each group (e.g., group 7) will meet on a designated day/time for one hour every week starting September 15. There will be approximately 10 meetings in total.

We will share with you the schedule of sessions and possible times through myCourses in the first weeks of class. Option B students will choose one of the days/times that is convenient, and that will be their slot for the rest of the term.

For each of the eight presentation weeks, a list of presentation exercises will be posted on myCourses. One of the subgroups of each group (e.g., 7B of group 7) will present 4-5 exercises to the other subgroups (e.g., 7A, 7C and 7D). Students from the same subgroup should prepare and work together. In some cases, each member will have to present one exercise, while in other cases, some students can present together, depending on the exercise. The subgroups that are not presenting on a given week will attend and give feedback, as described below.

Each subgroup will present in two of the eight weeks. The particular dates for presentations for each group will be shared with students on myCourses. *Each of the two presentations is worth 3% of the course grade.* Note that students will be graded individually here.

For the six (out of eight) weeks on which a subgroup is not presenting, the students will instead listen to the presentations of the other groups and give feedback with questions or comments. The best 5 of 6 participation grades will be selected and each will count for 0.6% of your final grade, for a total participation grade of 3 %. Note that we take the best 5 of 6 so that if you miss a session for whatever reason, you will not be penalized.

Guidelines for presentations and feedback will be shared with you as a PDF on myCourses. The guidelines will contain both the rubric used to evaluate your presentations as well information on how your feedback/participation will be evaluated.

In addition to the presentations and feedback above, students will also collaborate with their subgroup on *two group exercises, each worth 3% of their final grade.* These exercises will be similar to assignment questions, but less time consuming. Students will work together during the same weekly time slot as above, supervised by the team leader. (They may possible need to meet outside of class time to continue to work on the exercise.) Non-contributing group members will receive a mark of 0 for that exercise.

We believe these group activities overall will contribute to helping Option B students in several ways: learning how to work in groups, improving presentation skills, better understand the content presented in lectures. This is in line with research which shows that, through peer learning activities, students:

- build greater confidence and independence in learning, and obtain a deeper understanding of the course content;[4]
- critically reflect on their own work by engaging with the assessment criteria as they provide feedback on peers' work and by considering their peers' alternate approaches to the problem; [5]
- develop assessment skills for future academic and post-university work; [6] [7]
- receive more useful feedback which helps to identify shortcomings in their understanding of the course content [8]

The group activities will also give students an opportunity to meet and interact with other students in the course, and will hopefully be fun!

4.6 Regrade Requests

If you believe that you have been assigned an incorrect grade for one of your assessments (assignment, midterm, final, group exercise or presentation), you can request a regrade. Only well-argumented requests will be considered. *You may request a regrade within only 7 days from when your grade was published.* Requests after the deadline will not be considered.

Regrade requests for assignments must be made through the codePost website, though followup may be by email. Requests made initially by email will not be considered.

[4] Keenan, C. (2014). Mapping student-led peer learning in the UK. Higher Education Academy. Retrieved from: www.advance-he.ac.uk/knowledge-hub/mapping-student-led-peer-learning-uk (accessed 19 July 2019).

[5] Mulder, R., Baik, C., Naylor, R., & Pearce, J. (2014). How does student peer review influence perceptions, engagement and academic outcomes? A case study. Assessment & Evaluation in Higher Education, 39(6), 657–677. https://doi.org/10.1080/02602938.2013.860421

[6] Rubin, R. S. (2006). The academic journal review process as a framework for student developmental peer feedback. Journal of Management Education, 30(2), 378–398. https://doi.org/10.1177/1052562905277185

[7] Wanchid, R. (2015). Different sequences of feedback types: Effectiveness, attitudes, and preferences. PASAA: Journal of Language Teaching and Learning in Thailand, 50, 31–64.

[8] Grunwald, D., Boese, E., Hoenigman, R., Sayler, A., & Stafford, J. (2015). Personalized attention @ scale. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (pp. 610-615).

4.7 Additional work

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

4.8 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.

4.9 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.

5 Policies on Academic Integrity

Official policy: 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).

5.1 Plagiarism Policy and Assignments

You must 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 or module is entirely your own, and represents only the result of your own efforts.

Work submitted for this course must represent your own efforts. Assignments *must* be done *individually*; you *must not* work in groups. Do not ask friends or tutors to do your work for you. You *must not* copy any other person's work in any manner (electronically or otherwise), even if this work is in the public domain or you have permission from its author to use it and/or modify it in your own work (obviously, this prohibition does not apply to source code supplied by the instructors explicitly for this purpose). To be clear, you must not copy or view any code from any source, online or otherwise, except the code written by the instructors during lectures. Furthermore, you **must not** give a copy of your work to any other person.

The plagiarism policy is not meant to discourage interaction or discussion among students. You are encouraged to discuss assignment questions with the instructors, TAs, Mentors, and your fellow students. There is no better way to learn than through discussion with your peers. You are also encouraged to help each other out with debugging problems, especially with the basic mechanics of debugging, such as how to make the best use of an IDE. Finally, you are highly encouraged to post questions on Ed and to answer each other's questions there as well. However, there is a difference between discussing ideas and working in groups or copying someone else's solution. Your discussion should never go so far that you are revealing the solutions to each other. *Sharing code is absolutely forbidden.* The solution code that you submit must be your work. A good rule of thumb is that when you discuss assignments with your fellow students, you should not leave the discussion with written notes. Also, when you write your solution to an assignment, you *must* do it on your own.

5.2 Getting Help and Partial Credit

Students who require assistance with their assignments should see a TA/Mentor or instructor during office hours or make use of the Ed discussion board. If you have only partially finished an assignment, *comment out the parts that do not work*, and submit what you managed to complete for partial credit.

Code files that do not run at all (syntax or runtime errors) will be heavily penalized and almost certainly result in a grade of zero, so make sure to submit only working code, even if it does not fully satisfy the assignment requirements.

5.3 Plagiarism detection

We will be using automated code similarity detection tools to compare your assignment submissions to that of all other students registered in the course. These tools are cutting-edge and are very effective at their job. 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.

5.4 Posting assignment solutions on a website

You must not share your assignment solutions by posting them 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.

6 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.

7 Accommodations

7.1 Student Accessibility and Achievement Office

If you require accommodations, the Accessibility and Achievement Office is here 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.

7.2 Pregnancy and Caregiving

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.

Campus Computer Laboratories

All students registered in COMP 208 may use the computer laboratory facilities provided by the School of Computer Science to do their work regardless of the program in which they are registered. These facilities are located on the third floor of the Trottier building. To learn more, please see here.

You may also use other computer laboratory facilities on campus to do your work. Most facilities are available to all McGill students, but there are facilities which grant usage privileges only to students registered in a course or program offered by the faculty or department which manages the facility. Students should contact the work area of their choice to inquire about access requirements, opening hours, or any further information such as software availability.

8 Health and Wellness Resources

The academic environment at McGill is challenging. Our terms are intensive, and classes are not the only demanding part of your life. Student well-being is a priority for McGill. 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 Campus Computer Laboratories

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

You may also use other computer laboratory facilities on campus to do your work. Most facilities are available to all McGill students, but there are facilities which grant usage privileges only to students registered in a course or program offered by the faculty or department which manages the facility. Students should contact the work area of their choice to inquire about access requirements, opening hours, or any further information such as software availability.

10 Required software

Typically, when programmers write code, they use what is called an integrated development environment (IDE) to write programs. IDEs provide an editor that allows you to type your program, commands to compile and run it, and many other useful tools, all in one application.

We will be using 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 editors out there, so if you prefer to use another IDE (such as Spyder or IDLE) or text editor (such as Sublime Text) you are welcome to do so. Note that if you use a different editor, the teaching staff may not be familiar with it and therefore could not be of help with any issues stemming from that editor.

11 Tutorials

Depending on the TA workload and availability, we may be able to offer tutorials. These will optional, and will be designed to help you with the material and assignments and prepare for midterm and final exams, and to give you a chance to ask questions in a smaller environment than lectures.

12 Course Topics and Schedule

The following schedule may change depending on how the semester unfolds.

Week	Topics	Notes
0	Course introduction Binary numbers How does a computer work? What is programming/computer science? Thonny Basic Python programs Values and types	
1	Variables, assignment statements input prompt, mathematical, relational and logical operators expressions conditional statements	
2	functions: void vs. fruitful docstrings scope `math` module	
3	Turtle plagiarism while loops Newton's sqrt algorithm for loops	A1 out
4	`random` module numerical methods: integration, root finding modules and import string slicing and methods	
5	string traversal bug types (syntax, runtime, logic) debugging	A1 due A2 out
6	Thanksgiving + Fall Break	

Week	Topics	Notes
7	lists list algorithms: search and sort lists and strings Object identity, mutable vs immutable	A2 due
8	Nested lists Shallow vs deep copies doctest A2 due	
9	file I/0	midterm exam
10	Dictionaries Tuples & `enumerate` `matplotlib`	A3 out
11	OOP: objects and classes Instance methods `__str__`	
12	`numpy` image analysis	A3 due, A4 out
13	linear algebra - Gauss Jordan linear regression random walks Monte Carlo integration	
14	Special topics Exam Review (time permitting)	A4 due