McGill

Faculty of Science

COMP 202 - Foundations of Programming

Course Outline

McGill University, Winter 2021

Instructors:

Giulia Alberini

Office:	McConnell Engineering (MC) 233
Contact Info:	giulia.alberini@mcgill.ca
Office Hours:	**TBD**

Jonathan Campbell

Office:	McConnell Engineering (MC) 228
Contact Info:	jonathan.campbell@mcgill.ca
Office Hours:	**TBD**

1 Course Description

Welcome to COMP 202! Please read this document carefully and keep it for reference throughout the term.

This course introduces students to computer programming and is intended for those with little or no background in the subject. No knowledge of computer science in general is necessary or expected. On the other hand, basic computer skills such as browsing the Web, sending e-mail, creating documents with a word processor and other such fundamental tasks will be necessary in this course.

The course uses the Python programming language. Python is an example of a programming language (as are 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 students *a way of thinking* that will enable them to build non-trivial programs.

1.1 Primary Learning Objectives

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

- Design and describe precise, unambiguous instructions to solve a problem or perform a task;
- Translate these instructions into a language that a computer can understand (Python);
- Write programs that solve complex problems by decomposing them into simpler subproblems;
- Apply programming style conventions to make your programs easy to understand, debug and modify;
- Learn independently about new programming language features and libraries, as you encounter them, by reading documentation and by experimenting;
- Give and receive constructive feedback on your/others work.

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 e-mail, browse the Web, create word processing documents or spreadsheets, set-up 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. However, the course offers introductory tutorials that provide instruction in aspects of computer usage necessary to complete coursework.

1.3 Course Prerequisites

A CEGEP-level mathematics course or equivalent. For students who did not attend CEGEP, any upper-level mathematics course is sufficient. However, attention to detail, rigour, and the ability to think in an abstract manner is much more important than knowledge of calculus, algebra, or trigonometry.

2 Course Design

From last term, the course was redesign with the intention of keeping student engagement at the very center. We therefore decided to use a flipped classroom approach, where we will provide prerecorded concept videos that you can watch at your pace and time each week, in addition to weekly live sessions dedicated to active learning activities. The activities on which we will be focusing this semester are:

- Short peer-to-peer presentations
- Live polling
- Case studies
- Think-Pair-Share

All such activities, as well as the tools we will be using throughout the term, will be presented and explained to you during the first two weeks of class. We believe that these activities will contribute to helping students learn how to work in groups, improve their presentation skills, and, most importantly, better understand the content presented by the instructors through the concept videos. Research shows that, through peer learning activities, students:

- build greater confidence and independence in learning, and obtain a deeper understanding of the course content;[1]

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

- 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; [2]
- learn to provide feedback in a constructive manner;
- develop assessment skills for future academic and post-university work; [3] [4]
- receive more feedback than when instructors are the only ones providing it;
- and, in computer science, peer instruction has been found to improve final exam performance [5], reduce failure rates [6], and contribute to improved retention[7].

To encourage community building and enhance your learning experience, at the beginning of the semester you will be randomly divided into groups of 16-20 students that you will regularly interact with throughout the term. Each group will be assigned a TA and a TEAM mentor with whom the group will be able to have weekly exchanges. Each group of 16-20 students will also be further sub-divided into four smaller groups, each containing 4-5 students. These smaller groups are the groups within which you will be working more closely. Overall, there will be 20-25 groups of 16-20 students, each subsequently divided into 4. To keep things as simple as possible, we will be numbering the main groups using numbers from 1 to 30. For instance, group 23 might contain the following 18 students:

Shoaib
Miguel
Xin
Teri
Mahir
Yue
Fei
Janet
Samuel
Miles
Cassia
Clinton
Louise
Breanna
Emma
Yuxuan
Maxime
Hugo

We will then refer to the four smaller subgroups of group 23 by using the group number together with a letter. For instance, these could be the four subgroups of group 23:

23A	23B	23C	23D
Shoaib	Janet	Emma	Cassia
Fei	Breanna	Teri	Maxime
Louise	Xin	Miles	Yue
Miguel	Samuel	Yuxuan	Clinton
		Mahir	Hugo

Depending on the particular type of in-class activity, you might find yourself working with the larger group or the smaller group. At times, you might also find yourself paired with a random student from the class, but this will happen rarely. We want to keep the same groups throughout the term to give you an opportunity to get to know each other, build trust, and learn how to work as a group.

In order for you to be able to work with the same group throughout the semester, we need to first divide the

[2] 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

[3] 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

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

[5] Simon, B., Parris, J., & Spacco, J. (2013, March). How we teach impacts student learning: peer instruction vs. lecture in CS0. In Proceeding of the 44th ACM technical symposium on Computer science education (pp. 41-46).

[6] Porter, L., Bailey Lee, C., & Simon, B. (2013, March). Halving fail rates using peer instruction: a study of four computer science courses. In Proceeding of the 44th ACM technical symposium on Computer science education (pp. 177-182).

[7] Porter, L., & Simon, B. (2013, March). Retaining nearly one-third more majors with a trio of instructional best practices in CS1. In Proceeding of the 44th ACM technical symposium on Computer science education (pp. 165-170).

entire class into 4 groups of 100-120 students each. These four groups will be meeting weekly for a one hour live class together with one of the instructors and the TAs assigned to those students. Two of these groups will be meeting on Fridays from 12:30pm to 1:30pm. The other two will be meeting on Fridays from 2pm to 3pm. **IMPORTANT: You will have a chance to self-enroll in the group of your choice until until Jan. 24th at 11:59pm. To do so, from the myCourses page, click on "Groups" on the top navigation bar. Detailed information on how to self-enroll in a group can be found here. If you have not chosen a group by Jan. 24th at 11:59pm, then you will be randomly assigned to one.**

3 Course Materials

3.1 Prerecorded concept videos

Recordings of videos introducing the main programming concepts will be made available to all students on myCourses.

3.2 Lecture Slides, Tutorial Slides

The slides used in the recordings by the instructors and those used by TAs in the tutorials will be made available to all students on myCourses.

3.3 Recommended Exercises

Sets of recommended exercises will be posted every week. The solutions to some of these exercises will be presented by students during our weekly live lectures.

3.4 Recommended Textbook

Think Python 2e, by Allen B. Downey.

Available at no cost under the terms of the Creative Commons Attribution-NonCommercial 3.0 Unported License at `https://greenteapress.com/wp/think-python-2e/`.

3.5 Other References

- *How to Think Like a Computer Scientist: Interactive Edition*, by Brad Miller and David Ranum. This is an interactive textbook based on the work by Jeffrey Elkner, Allen B. Downey, and Chris Meyers. Available at no cost at:

  `https://runestone.academy/runestone/books/published/thinkcspy/index.html`

- *The Python Tutorial.* You can browse or download this from Python Software Foundation's Web site at:
  - `https://docs.python.org/3/tutorial/`
- *The Python 3 Documentation.* You can also browse or download this from Python Software Foundation's Web site at:
  - `https://docs.python.org/3/`

3.6 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, tutorial slides, recommended exercises, assignment questions, or anything else provided to you during the course.

Stated more formally: "Instructors/TAs-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 instructors/TAs. Note that infringements of copyright can be subject to follow up by the University under the Code of Student Conduct and Disciplinary Procedures."

4 Communication Policies

4.1 Office Hours

Teaching Assistants (T.A.s), TEAM Mentors and instructors will be available for office hours through Zoom to help you with your assignments and answer questions about the course material.

Note that this term we will be offering three different types of office hours:

- Small group office hours offered by the TA/TEAM mentor assigned to your group. These will be weekly office hours scheduled within your groups based on yours and the TA/TEAM Mentor personal availability.
- Individual (first come first serve, through the use of a waiting room) office hours. The schedule of these office hours will be set at the beginning of the semester and shared with all of you. You should feel free to attend as many as these office hours as you want, and whichever one best fits with your schedule.
- Open office hours: weekly office hours led by 1 TA which will focus on reviewing questions from the quizzes or FAQ regarding assignments.

A link to a Google calendar with everyone's office hours will be shared with you on myCourses.

4.2 Discussion board

We will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TAs/Mentors, and the instructors. Rather than emailing questions to the teaching staff, **we encourage you to post all your questions related to the course content and the assignments on Piazza**. By doing so, you will be sure to receive an answer faster, and everyone in the class will be able to benefit from it. 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). If you have any problems or feedback for the developers, email `team@piazza.com`.

Find our class page at: `http://piazza.com/mcgill.ca/winter2021/comp202`

Discussion Board Guidelines

Please help out by answering each other's questions. The instructor and TAs will try to moderate the Discussion Board, but the Discussion Board works best when students help each other out. When posting to the Discussion Board, please obey the following guidelines. *Posting that do not conform may be deleted.*

- Choose the appropriate folder (matching the topic).
- **Use the search feature to see if your question has been asked before.**
- Choose a suitable subject line, so that readers know what the posting is about.
- If you have multiple questions that are unrelated, then use multiple postings so people can more easily follow the thread.
- Proofread before posting. Take an extra minute to ensure that what you write makes sense.
- If you would like your posting to be deleted, just add a request within the thread.
- Be polite and respectful.

4.3 Contacting Instructor and Teaching Assistants

For private matters only, you can send e-mail to a teaching assistant or instructor directly with "COMP 202" in the subject header. Be sure to send your email from your `@mail.mcgill.ca` address and include your student ID. When emailing instructors or TAs, please follow the guidelines on etiquette described in the video here and on this website.

4.4 Course Announcements

Important information about the course will be announced in class and/or on myCourses and Piazza. Please subscribe now to myCourses Announcements, if you haven't done so already.

Students are expected to monitor both their McGill e-mail account, myCourses, and Piazza for course-related news and information.

5 Grading Scheme and Deadline Policy

Your final grade in the course is calculated using the following scheme:

- **Assignments:** 30%
- **Quizzes:** 30%
- **Final Project:** 25%
- **Presentations:** 8%
- **Feedback:** 5%
- **Participation in surveys:** 2%

When we calculate your final course grade, 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 a hard line on this, so if you don't want to fail then you should stay far away from that line.

Official language policy for graded work: 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. See here for more details: `https://www.mcgill.ca/study/2019-2020/university_regulations_and_resources/undergraduate/gi_lang_policy`.

5.1 Assignments

There will be **three** 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. Each assignment is worth 10% of your final course grade. Here are the dates on which the assignments are expected be posted:

- Assignment 1 to be posted around January 25
- Assignment 2 to be posted around February 16
- Assignment 3 to be posted around March 20

You will be given *at least* 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 the section 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.

Assignment Submissions

Assignment submission will always take place through codePost. **You will be added as COMP 202 students on codePost by January 24th.** The instructors and TAs will discuss how to use codePost during the lectures and tutorials, but every student is responsible for verifying that their submissions are successful. If you believe the content of your submission is different from what you have submitted, you must e-mail your instructor within **4 days** of the assignment deadline in question to provide evidence of your correct submission.

Assignment marks will be visible both on codePost and on myCourses. It is your responsibility to check that your marks are correct and to notify your instructors of any errors or missing marks. If you believe that your assignment was graded incorrectly, you should submit a question on your graded submission through codePost and request a regrade. Only well argumented requests will be considered. **Note that you can request a regrade only within 7 days from when your grade was published.** Requests after the deadline will not be accepted.

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; late assignments will be deducted 10% each day or fraction thereof for which they are late, including weekend days 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 illness or other unforeseen personal circumstances. However, you must make a formal request (see email policy).

Examples of invalid requests are:

- Your laptop broke or was stolen. This is not a valid excuse. You should be using an automatic backup system, e.g. Dropbox, Google Drive, etc.
- You have midterm exams, a job interview, a family wedding, etc. These are invalid because are not unexpected and you have two weeks to complete your assignments. You need 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 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.

5.2 Online Quizzes

We will have 6 biweekly quizzes throughout the semester, containing a combination of multiple choice questions, true/false, fill in the blank, etc. These quizzes will be administered through myCourses. We strongly suggest that you do them in a location where the internet connection is as reliable as possible.

Quizzes will open on Thursday at 12pm and remain available until 11:59pm on Sunday night. They will test you on the topics covered by the concept videos in the previous two weeks. Quizzes will be timed. The quizzes are designed to take less than 60 minutes, so in line with the principles of Universal Design for Learning (see our section on Accommodations), you will have 150 minutes from the time you start the quiz to complete it. If you do not write a quiz, you will receive 0 for it.

It is possible that you will not be available to write all of the quizzes. Therefore, we will be selecting only your 5 best quizzes out of 6 to count toward your final course grade, with each of them worth 6% of your final course grade.

The quizzes must be done entirely on your own. See our policy below on "Cheating on quizzes".

5.3 Final Project

You will be asked to complete a final project due on April 30th which consists of larger scale assessment involving many concepts presented in the course. The project counts for 25% of your final course grade.

5.4 Presentations

One of the main activities we will be using this semester involves in-class presentations. Each week we will be uploading a list of recommended exercises to practice what you have learned in the pre-recorded concept videos. We encourage all the students to try out these exercises consistently. Each small group of 4-5 people will rotate presenting 4-5 exercises (one per member) in class to the other groups. For instance, in week 3 the small groups named 'xA' (where x is a number) will be presenting to 'xB', 'xC', and 'xD'. Students from the same small group can prepare and work together, but each member will have to present one exercise and they will be evaluated on their presentation. This allows us to have "experts" on the recommended exercises every week. All the students from the larger group (comprising xA-xD) will then be able to refer to at least one of their peers for help and clarification. The small groups that are not presenting will instead give feedback to each presenter on their work. This will allow each student to receive feedback on how to improve their presentation skills.

Your work during the presentation activities will comprise 13% of your final grade. You will be presenting your work on a question on two live sessions during the semester (the particular dates will be determined by your group number), with each presentation worth 4% each. The other 5% will come from the peer feedback that your group submits on every other group's presentations. That is, during every live session that you aren't presenting (a total of six), you will listen to the presentations of other groups, and write up, with your team, feedback on those presentations. A guide on how to successfully participate in these activities will be shared with you. The guide will contain both the rubric used to evaluate your presentations as well as the rubric used to evaluate your comments. The best 5 out of 6 feedback will be selected and each will count for 1% of your final course grade.

Please note that to get credit for your feedback you MUST be present during the entire duration of the group presentation. **If one submits a feedback file without being present in the live session**, not only the student will be receiving a 0 for their work, but since this is considered to be an academic offence, the case will be reported to the Disciplinary Officer of the student's Faculty.

NOTE: If you cannot regularly take part to the live sessions activity due to ongoing and long-term technical reasons (cannot connect to Zoom, bad internet connection), then please **inform us before the first presentation** by emailing your documentary evidence to this effect. We will then replace the activity with an **oral exam** that you'll be taking with the instructor toward the end of the course (date to be determined) which will count for 13% of your final grade.

5.5 Participation in surveys

Our course has been invited to participate in a new study on students' experiences in introductory STEM courses, led by Professor Kristy Robinson from the Department of Educational and Counselling Psychology. To help us understand students' experiences in the course, you will be asked to complete four surveys (10-15 minutes each) about this course during the term.

The first survey is due on Jan. 18th. The links needed to fill out the surveys will be shared with you on Piazza/myCourses.

For completing the surveys you will receive 2% of your final course grade. Note that you have to complete *all* the surveys to receive this 2%. Note that if you enrolled to the course *after* Jan. 18th, you will still have the opportunity to fill out the first survey.

You will have the opportunity to release your survey responses and your course grades for use in a research study on students' experiences in introductory STEM courses. **Your responses and your choice to participate in the study will be confidential, will not affect your grades in any way, and will only be seen by the researchers and not your instructors**. If you choose to allow your surveys to be used in the research, your identity will remain confidential in all reports stemming from the study. Thank you for your help with improving COMP202 and science education!

5.6 Regrade Requests

Please note that if you think that you have been assigned an incorrect grade for one of your assessments (assignment, quiz, feedback, or presentation), you can request a regrade. **Note that these requests will be considered only if made within 7 days from when your grade was published.** Requests after the deadline will not be accepted.

5.7 Supplemental/Deferred Exam

There will be no supplemental or deferred exam for this course as there is no final exam.

5.8 Additional Work

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

5.9 Extraordinary Circumstances beyond the University's Control

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. See section 3.2.3 of the *University Student Assessment Policy.*

6 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 offenses under the Code of Student Conduct and Disciplinary Procedures (see www. mcgill. ca/ integrity/ for more information).*

6.1 Plagiarism Policy and Assignments

You must include your name and McGill ID number at the top of each source 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 instructors explicitly for this purpose). 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 instructors, TAs/Mentors, and your fellow students. There is no better way to learn than through discussion with your peers. You are highly encouraged to post questions on Piazza and to answer each other's questions there too. However, there is a difference between discussing ideas and working in groups/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 should do it on your own.

6.2 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 discussion board on Piazza. 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.

6.3 Plagiarism Detection

We will be using automated code similarity detection tools to compare your assignment and quiz submissions to that of all other students registered in the course, and these tools are very effective at what they have been designed for. However, note that the main use of these tools is to determine which submissions should be manually checked for similarity by an instructor or TA; we will not accuse anyone of copying or working in groups based solely on the output of these tools.

You may also be asked to present and explain your assignment submissions to an instructor at any time.

When the 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:

`https://www.mcgill.ca/secretariat/files/secretariat/code_of_student_conduct_and_disciplinary_procedures.pdf`

6.4 Posting assignment solutions on a website

We encourage you to use tools like GitHub for version control systems. However, you must not share your assignment solutions by posting them on a public space such as your GitHub account.

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 (GitHub has a search feature) then this leads to plagiarism by others.

6.5 Cheating on quizzes

The quizzes will be administered through myCourses and you will have a chance to do them within a 84 hours time window, and wherever you like. We will use the honour system here, namely you must do the quizzes entirely on your own just as if you were writing an exam in class. Any communication between two students about a quiz before the time window is closed and the quiz is complete is cheating and is absolutely forbidden. Taking screen shots of the questions and/or posting any material from the quizzes online during or after the quiz is taking place is considered to be a violation of Student Code of Conduct and will be reported to the disciplinary officer of your faculty.

7 Land Acknowledgment

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 peoples of the world now gather. Please see here for more details: `https://www.mcgill.ca/edu4all/other-equity-resources/traditional-territories`.

8 Accommodations

For this course, we are adopting flexible assessment strategies that create greater access for all students by incorporating principles of Universal Design for Learning. As such, we have taken into consideration the variety of learner needs and barriers that students may face in this course and have designed the assessments with these considerations in mind. Additionally, we recognize that any student may experience unexpected interruptions in a remote learning environment. Therefore, additional time has been built into the assessment strategy to address these potential barriers. Because of these modifications, authorized time-based accommodations for students registered with OSD will not apply.

8.1 Office for Students with Disabilities

There may be exceptional circumstances in which other disability-related accommodations may still be needed. If you feel this is the case for you, please reach out to OSD via email at `exams.osd@mcgill.ca`. They will assess the situation and coordinate with the instructors when necessary.

8.2 Pregnancy and Caregiving

Students who are pregnant and/or caring for a dependent also often 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 at `https://www.mcgill.ca/study/2018-2019/university_regulations_and_resources/graduate/gi_accommodation_pregnancy_caring_dependants`

9 Required Software

Typically when programmers write code they used 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 recommend using a simple Python IDE called Thonny. Thonny comes with Python 3.7 built in. This means you only have one thing to install and you are ready to learn how to program! :)

You can install Thonny through going to `https://thonny.org/` and following the instructions to download and install the software. Thonny supports Windows, Mac and Linux.

(Note: If you use macOS Catalina or Big Sur (10.15 or 10.16), please download Thonny at this link.)

There are many other editors out there, so if you prefer another IDE (such as Spyder or PyCharm) or using a text editor (such as Atom or Gedit) you are welcome to do so. Note that if you use a different editor, the teaching staff may not be familiar with your choice of editor.

10 Course Content

Note that minor changes in content, reading material, and times for tutorials and assignments may occur. It is your responsibility to keep up with the course and be aware of what content is being covered.

10.1 Tutorials

Throughout the term, there will be several (optional) tutorials. These will be designed to help you with the material and assignments, and to give you a chance to ask questions in a smaller environment than lectures. It is not necessary to register for tutorials.

The tutorials will review and practice material presented in class. For example, a tutorial in the sixth week might cover for loops and strings to ensure that everyone is caught up.

The schedule of the tutorials will be shared with you on myCourses and Piazza. The Google calendar containing all the information on the course office hours will also contain all the tutorials.

10.2 Approximate Schedule of Topics

The references to chapters in the table below are from the recommended textbook (`https://greenteapress.com/wp/think-python-2e`). Although our lectures will not follow the textbook exactly, reading the textbook is highly recommended. **The following schedule is only approximate and may/will change depending on how the semester unfolds.**

	Week	Topics	Reference
Intro	1–Jan 10	What is programming? What is computer science? How does a computer work? Binary numbers	Chapter 1
Fundamentals	2–Jan 17	Basic Python programs Values and types Variables, assignment statements, and inputs	Chapter 2
	3–Jan 24	Mathematical, relational and logical operators The `math` module, and expressions Conditional statements	Chapter 5
	4–Jan 31	Functions and docstrings Scope of variables While loops	Chapters 3, 6 and 7
	5–Feb 7	Modules, and `random` Turtle	Chapter 4
	6–Feb 14	For loops and loop commands Strings and loops Debugging	Chapters 8-9 Appendix A
	7–Feb 21	Raising Exceptions Nested loops	
	8–Feb 28/Mar 7	Intro to lists Objects identity Mutable vs Immutable	Chapter 10
	9–Mar 14	How to use doctest Nested lists Shallow vs deep copies	
	10–Mar 21	Dictionaries Tuples enumerate Function objects	Chapters 11-12
	11–Mar 28	File IO Exception handling Matplotlib	Chapters 13-14
OOP	12–Apr 4	Intro to objects and classes Instance methods `__str__`	Chapters 15-16
	13– Apr 11	class and static methods Objects and equality	Chapter 17