McGill

Faculty of Science

COMP 250 - Introduction to Computer Science

Course Outline

McGill University, Winter 2023

Instructor:

Giulia Alberini
Office: McConnell Engineering (MC) 233
Contact Info: giulia.alberini@mcgill.ca
Office Hours: Fridays, 10:00 - 12:00 in MC 103

1 Overview

This course introduces you to two core topics in computer science: data structures and algorithms. You will learn basic data structures for lists (arrays, linked lists, stacks, queues), trees (search trees, heaps), and graphs. You will also learn basic algorithms – both recursive and non-recursive – that use these data structures. You will also learn how to analyze such algorithms in terms of the amount of computation they use. These data structures, algorithms, and analysis tools all will be used heavily in subsequent CS courses.

The assignments in the course will use Java programming language. You will learn how to implement basic data structures and algorithms using Java. Java is a object oriented language, and so you will also learn some of the basic ideas of object oriented design such as how classes can be organized into hierarchies and how variables and methods defined in the classes of the hierarchy are related to each other.

2 Prerequisites

According to `https://www.mcgill.ca/study/2022-2023/courses/comp-250` the official prerequisite is *"Familiarity with a high level programming language and CEGEP level Math."* Here are more details about the programming and math prerequisites.

2.1 Programming prerequisites

Starting in Fall 2019, both COMP 202 and COMP 208 switched from Java to Python. For this reason, we will be assuming that students coming into 250 know basic Python at the level of COMP 202. All programming

assignments in COMP 250 will be in Java, and most of the examples seen in class will also use Java. We will be using the first two weeks of the semester learning enough Java to allow you all to enjoy and successfully complete COMP 250.

Here are some frequently asked questions (FAQ) about the programming prerequisite:

Q: I took COMP 208 (before Fall 2019) and so I know some C but not Java (nor Python). What am I missing?

A: If you know some C, then you are in a strong position to learn Java because you are very familiar with *types* already. However, you don't yet know about objects and classes. We will learn about them, and how to use them in Java, during the first couple of weeks.

Q: I took a programming course prior to coming to McGill in which we used one of Python / Matlab / Javascript / R. What am I missing?

A: As described above, we will be spending some time together learning Java. The assumption we will be making is that you have a certain level of experience in Python. Be sure that you are comfortable with the basics elements of programming from COMP 202, in particular, variables, expressions, and assignments, conditional statements (if-then-else), loops (while, for), methods/functions, data structures such as list/arrays and strings, input/output from a keyboard and to a console and from/to a file. *If you are not comfortable with these basics, then you should take COMP 202 or 204 or 208 this semester, instead of COMP 250.*

Q: I have not taken a programming course. Instead I learned programming on my own, e.g. online course. How much programming experience do I need?

A: You should have at least 50 hours experience programming in whatever language you do know. That is roughly the minimum amount of experience that a student who has taken COMP 202/208/etc already has. If you don't have that level of experience, then you should not take COMP 250 this semester, and instead you should take an introduction to programming course, namely COMP 202 or 204 or 208.

2.2 Math Prerequisites

The official prerequisite is "CEGEP level math". This means specifically Calculus 1 and 2. Although COMP 250 will not use derivatives and integrals, some of the ideas from Calculus will be used. For example, we will use the limits of a sequence when we discuss the runtime of different algorithms and we compare one algorithm to another. We also will use sequences and series, and so you are expected to know the difference between an arithmetic versus geometric series. Another important concept from Calculus is logarithms. You need to know how logarithms are defined - namely a logarithm is the inverse of an exponential. You also need to know and understand the basic rules of logarithms. Finally, the course will require that you are able to think logically. Although most of you will not (yet) have studied formal logic, you will nonetheless be expected to understand at least intuitively what statements like "for all" and "for each" mean, and to understand intuitively how to negate such statements. Such mathematical thinking will come more naturally to those of you who are stronger in mathematics, but everyone is capable of such thinking to some extent – and it improves with practice. So if you haven't yet taken Calculus e.g. if you are doing a B.A., then you should at least do Cal 1 before taking this course.

2.3 Recommended Co–requisites

If you are registered for COMP 250 in Fall 2020 and you are thinking of pursuing a program in Computer Science, then we strongly recommend that you take:

- MATH 240 (for CS only programs) or MATH 235 (if you do Math & CS program). These courses will help you with the mathematical parts of COMP 250. Moreover, you *must* take one of these two MATH courses before taking COMP 251. [Note: MATH 235 is not offered in Winter 2023]

- Calculus 2, assuming you have taken Calculus 1. Some of the upper level COMP courses will require Calculus 3, so the sooner you take Calculus 2 the better.
- COMP 206. This is not necessary though, and if you are just doing a minor in CS then maybe not do 250 and 206 at the same time. COMP 206 will use the C programming language. There are many similarities (and differences!) between C and Java, so there are advantages but also disadvantages in doing both at the same time.

  If you are considering taking COMP 250, 206, and 273 all in one semester, then I would suggest not doing so – unless you have a lot of programming experience already. Instead just take 250 and 206 and leave 273 for next semester.

3 Course Materials

3.1 Required Software

The programming language for this course is Java. Thus, you will use the Java compiler and the Java Virtual Machine (JVM) to compile and run the programs you are required to write for the assignments. The Java compiler and the JVM are included in a larger software package called the Java Development Kit (JDK).

You can use any **plain-text editor** of your choice to write your programs, and then use the tools included with the JDK to compile and run them. For this course, we recommend Eclipse (`http://www.eclipse.org/`) or IntelliJ, a more powerful IDE which can assist you in writing your code. The entire teaching staff will provide support for both.

You are encouraged to install the JDK and IDE on your own machine so you do not have to depend on the SOCS computer laboratory facilities to do your work. Installing any of these is fairly straightforward. If you need help, you will find additional instructions on myCourses or you can consult a TA during office hours.

- **Required:** The JDK.
  - Windows users: You may download the JDK installation program from the following Web site: `http://www.oracle.com/technetwork/java/javase/downloads` (choose Java - Download or JDK (click on the Download JDK button), with no additional software such as Java EE or NetBeans). The JDK is available at no cost, and there is no time limit on its use. **You should install the JDK before any IDE.**
  - Mac users: JDK 6.0, 7.0, or 8.0 is installed by default on most Mac computers. It is available as a Mac OS software update.
  - GNU/Linux users: A JDK is available in the software repositories of most of the major GNU/Linux distributions like Ubuntu or Fedora; you can install it through your package manager.
- **Recommended:** An IDE that you can use to write and run your programs. More powerful IDEs such as Eclipse or IntelliJ offer fantastic benefits such as automatically checking your code for errors and built-in debugger. This can be a great help when writing more complex programs.
  - Eclipse: `http://www.eclipse.org/downloads/` (choose Eclipse IDE for Java Developers)
  - IntelliJ IDEA: `https://www.jetbrains.com/idea` (choose Community)

3.2 Lecture Recordings

We will record the lectures, and make the recordings available on myCourses.

3.3 Lecture Slides, Lecture Notes, Exercises

There is no course textbook. Instead, we will have a set of slides, which will be made available on myCourses.

There is also a complete set of Lecture Notes and Exercises from Fall 2017 created by Prof. Michael Langer available `http://www.cim.mcgill.ca/~langer/250-2017.html`. Time permitting, we will update these materials to match the Winter 2023 lecture schedule and slide content, and post these materials in myCourses.

3.4 Copyright policy

You are not allowed to post any course materials on github, coursehero, any other websites. This includes PDFs of lecture slides, lecture notes, exercises, quizzes, assignment questions or anything else that we provide for you.

Stated more formally: "Instructor-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(s). 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 and Course Assistants (TAs and CAs), TEAM Mentors and the instructor will be available for office hours to help you with your assignments and answer questions about the course material.

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 Ed Discussion for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TAs/CAs/Mentors, and the instructor. 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 Ed**. 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).

Find our class page through the a link shared on myCourses.

Discussion Board Guidelines

Please help out by answering each other's questions. The instructor and TAs/CAs 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 250" 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.

If you have a technical question about the course material or an assignment, **do not email us**. Instead:

- If it is an assignment question, then see someone from the teaching staff during their office hours, or post your question on the discussion board.
- If it is a question about a midterm or lecture material, then post your question on the Ed Discussion Board. (See Discussion Board policies above.) That way, other students can benefit from the answer. We will do our best to answer your question within 24 hours.

  For other help with the course lecture material (but not assignments), you can ask at the CSUS Help Desk in Trottier 3rd floor.

4.4 Course Announcements

Important information about the course will be announced in class and/or on myCourses and Ed. 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 Ed 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:** 27%
- **Midterms:** 40%
- **Final Project:** 25%
- **Final Quiz/Presentation:** 8%

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 Java 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 assignments are each worth 9% of your final course grade. Here are the dates on which the assignments are expected be posted:

- Assignment 1 to be posted around January 30
- Assignment 2 to be posted around February 24
- Assignment 3 to be posted around March 24

You will be given two weeks to complete each assignment. The deadline will be specified on the assignment PDF. This is much longer than you need. We give two weeks to allow you flexibility. Get started early!

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.

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 more serious 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.
- You were sick for a few days. This is not a valid excuse since the assignment is posted for two weeks.

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

The instructor 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 your assignments 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 Midterms

We will have 2 in-person midterms (space permitting), each worth 20% of your final grade. They will be *closed book* examinations. No electronic devices are permitted. No calculators. No cell phones. A one

2-sided crib sheet will be allowed.

The midterms consist of open questions and will be graded out of 100 points each. The tentative dates for the midterms are:

- February 13 (6pm to 8pm)
- March 27 (6pm to 8pm)

5.3 Final Project

You will be asked to complete a final project due on April 27th 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 Final Quiz/Presentation

You will have the choice to end the semester with either a final quiz or a final oral presentation worth 8% of your final grade. You will be asked to make this choice by January 20th (details on how to do that will be announced on myCourses/Ed). You will not be able to change your mind after that. Please note that if you do not indicate a choice, then you will be automatically signed up to take the Final Quiz.

5.4.1 Final Quiz

The final quiz will be held on April 12th. If the date changes, we will announce it ahead of time on myCourses/Ed. The quiz will cover all the material seen throughout the semester and it will be a combination of multiple choice questions, true/false, fill in the blank, etc. It will be administered through myCourses. We strongly suggest that you do it in a location where the internet connection is as reliable as possible.

The quiz will open at 8am and it will remain available until 6pm. The quiz will be timed. It is designed to take less than 60 minutes, but we will give you 90 minutes from the time you start the quiz to complete it. If you do not write the quiz, you will receive 0 for it.

If you are registered with OSD and you require accommodations different than the one already provided please make sure to have your OSD advisor contact me.

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

5.4.2 Final Presentation

You can choose to replace the final quiz with an oral presentation (Note: you must do so by January 20th). With the oral presentation we would like to give you the opportunity to experience a mini mock coding interview. You will first present the solution to problems assigned to you, and then there will be time to answer various questions on the topics covered. Overall, the assessment will last no more than 30 minutes. All presentations will be scheduled in the last week of the semester depending on students and teaching staff's availability. More details on this will be given in class and on myCourses/Ed.

5.5 Regrade Requests

Please note that if you think that you have been assigned an incorrect grade for one of your assessments (assignment, midterm, quiz, or presentation), you can request a regrade. **Note that these requests will be considered only if made within 4 days from when your grade was published.** Requests after the deadline will not be accepted.

5.6 Supplemental/Deferred Exam

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

5.7 Additional Work

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

5.8 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. Therefore 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 rely on 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 instructor 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 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 you to pose questions on the discussion board and to answer each other's questions there too.

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 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. 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 and text matching software

The solutions that you submit must be your own work. We will run software for detecting similarities between submissions, and we will conduct a manual code review in cases where similarity between two solution is suspiciously high.

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. If you do and if another student copies your solution from there, then there will be no way to discriminate who did the work, and you may be accused of plagiarism along with the other student(s).

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 quiz will be administered through myCourses and you will have a chance to do it within a 12 hours time window, and wherever you like. We will use the honour system here, namely you must do the quiz entirely on your own just as if you were writing an exam in class. Any communication between two students about the 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 quiz 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.

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 Course Content

Note that minor changes in content, and times for tutorials and assignments may occur. It is your responsibility to attend class and be aware of what content is being covered.

9.1 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 be an occasion to implement what you have learned in class. For example, a tutorial in the sixth week might cover how to implement singly linked lists.

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

9.2 Approximate Schedule of Topics

The schedule on the next page is only approximate and may/will change depending on how the semester unfolds.

	Week	Topics	Events
Preliminaries	1–Jan 2	What is an algorithms? Java Syntax, Variable declarations and scope Methods, conditional statements, and loops	
	2–Jan 9	Primitive data types Arrays and Reference types Errors and Exceptions, try-catch	
OOD in Java I	3–Jan 16	Modifiers, fields, and constructors get/set methods, Mutable vs Immutable Packages, and UML diagrams	Add/drop deadline (Jan 17)

	Week	Topics	Events
	4–Jan Jan 23	Inheritance, and modifiers Object class and type conversion Polymorphism, and abstract classes	
Linear Data Structures	5–Jan 30	Array lists Singly linked lists Doubly linked lists	A1 posted
	6–Feb 6	Quadratic sorting algorithms Case study on Linked lists Asymptotic notations Big O, Big Omega, Big Theta	
	7–Feb 13	Stacks Queues	Midterm 1
OOD 2	8–Feb 20	Interfaces Iterable and Iterator Comparable	A2 posted
	9–Feb 27	Reading Break	
Recursion	10–Mar 6	Induction Recursion Binary search	
	11–Mar 13	Merge sort and quick sort Recurrences	
Non-Linear Data Structures	12–Mar 20	Trees Tree traversals Binary trees Binary search trees	A3 posted
	13– Mar 27	Heaps Maps and hash codes Hash maps	Midterm 2
	14– Apr 3	Intro to graphs Graph traversals	Presentation
	15– Apr 10	Extra topics	Final Quiz / Presentation Final Project posted