COMP 250 - Introduction to Computer Science

Course Outline

McGill University, Winter 2020

Instructors:

Giulia Alberini

Office:	McConnell Engineering 233
Email:	cs250@cs.mcgill.ca (if course related – see our email policy below) giulia.alberini@mcgill.ca (if not course related)
Office Hours:	Thursdays 16:00 – 18:00 in McConnell 103

Roman Sarrazin-Gendron

Office:	McConnell 324
Email:	cs250@cs.mcgill.ca (if course related – see our email policy below) roman.sarrazingendron@mail.mcgill.ca (if not course related)
Office Hours:	Mondays 13:15 – 15:15

Class Info:

Section 1		**Section 2**	
Lecture room:	ADAMS AUD	Lecture room:	ADAMS AUD
Class times:	MWF 9:35-10:25	Class times:	MWF 11:35-12:25

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/2019-2020/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 Winter 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, doing one of them now will help when you take COMP 251. You *must* take one of these two MATH courses either before or while taking COMP 251; we strongly recommend that you do it *before*.
- ... take Calculus 2, if you haven’t done so already. See Math prerequisites above.
- ... do not attempt to take COMP 250, 206, 273 all in one semester, unless you have a lot of programming experience already. Instead just taking 250 and 206 and leave 273 for next semester.

3 Course Materials

3.1 Lecture Recordings

We will record the lectures from section 2, and make the recordings available on myCourses. The two sections will be merged on myCourses, and will be treated as one single course - same lectures, same assignments, same exams. The instructors will alternate teaching the two sections based on the topics. This means that both sections will be taught by both instructors.

3.2 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 2020 lecture schedule and slide content, and post these materials in myCourses.

3.3 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 Assistants (T.A.s) will be available for office hours, on the third floor of the Trottier building room 3110, 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

This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TAs, 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: `https://piazza.com/mcgill.ca/winter2020/comp250/home`

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. *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.

4.3 Getting help from an instructor or T.A.

If you have a technical question about the course material or an assignment, *do not email the instructors*. Instead, in no particular order:

- See one of the T.A.'s/instructors during their office hours.
- Post your question on *Piazza*. (See discussion board policies above.)
- For help with course material (but not assignments), ask at the CSUS Help Desk in Trottier 3090.

4.4 Instructor email policy

Email the instructors *only if*:

- You have an urgent and important personal matter. In that case, assuming the matter is related to COMP 250, please email both instructors using cs250@cs.mcgill.ca and one of instructors will get back to you as soon as possible. Be sure to send your email from your mcgill address and include your student ID.
- You notice a mistake has occurred such as a problem with a question on a quiz, missing slides or PDF on myCourses, or lecture recordings haven't been posted within 48 hours of the lecture, etc.

When emailing instructors or T.A.'s, please follow the guidelines on etiquette described in the video here and on this website.

McGill official email policy is here. Note that you should include your name and student ID on all emails sent to instructors.

4.5 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 Evaluation

Your final grade in the course is calculated by taking the *maximum* of the following two options:

- **Assignments:** 40%
- **Quizzes:** 10%
- **Final Examination:** 50%

OR

- **Assignments:** 40%
- **Final Examination:** 60%

This means that students whose final exam percentage grade is greater than their quiz percentage grade will have their course grade automatically computed using the second scheme (i.e. 40% assignments, and 60% final). However, the assignments are a key part of learning the material, and as such there is no 100 % final option.

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.

There are many factors that determine your course grade including how hard you work, how talented you are in this subject, how much time you have available because of other commitments, what your academic background is, what your health situation or family situation is, etc. *However, we do not consider these factors when we calculate your final course grade. Rather, we calculate your final grade automatically, according to the grading scheme specified above.* No exceptions.

At the end of the course, students often contact instructors with requests for a grade change. When the situation described in such requests are handled by the policies above, we will refer you back to these grading policies.

5.1 Assignments

There will be **four** assignments throughout the semester consisting of writing Java programs. It is *very important* that you complete all assignments, as this is the best way to learn the material. Each assignment is worth 10% of your final course grade. Here are the dates on which the assignments are expected be posted:

- A1 to be posted around January 24
- A2 to be posted around February 10
- A3 to be posted around March 9
- A4 to be posted around March 23

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

To receive full grades, 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 myCourses. 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.

You will be able to see your assignment marks on myCourses. It is your responsibility to check that the marks are correct and to notify your instructor of any errors or missing marks.

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, 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 we give you two weeks. You need to plan accordingly.

Late submissions beyond the two day deadline are not allowed under any circumstances. If you are so far behind on your work that you cannot meet the late deadline, then this is a problem that extends beyond COMP 250 and you should seen an academic advisor and possibly consider a late withdrawl from one of your courses.

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 myCourses 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 five quizzes throughout the semester, each worth 2 % of your final course grade. The quizzes will be combination of multiple choice, true/false, etc.

These quizzes will be done online with myCourses. We strongly suggest that you do them in a location where the internet connection is reliable.

The quizzes are designed to take less than 40 minutes. We give you a full hour, and this is to accommodate students registered with OSD who need more time for exams. You can do the quiz anytime during the day from 8 AM to 8 PM, and from wherever you wish.

No crib sheets, or web access, etc is allowed, and so the conditions of the quiz are the same as the final exam.

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

It is possible that you will not be available to write all of the quizzes. Therefore, we are making the quizzes optional in the following sense. You will receive a grade of 0/2 if you cannot write a quiz. However, when we calculate your final course grade, if your final exam percentage grade is greater than your quiz percentage grade including quizzes that you missed, then we will *automatically* make your final exam worth 60% of your final grade instead of 50%, and your quizzes will be worth 0%.

The planned dates and topics covered by the quizzes are listed below.

1. Friday Jan 31 (Java syntax, primitive data types, arrays, and OOD basics)
2. Friday Feb. 14 (OOD (polymorphism and type conversion), Array lists Linked lists, and quadratic sorting)
3. Friday Feb. 28 (stack/queue, OOD (interfaces), and induction)
4. Friday Mar. 20 (recursion, binary search, merger sort, quick sort, and trees)
5. Friday Apr. 3 (heaps, hashing, and graphs)

5.3 Final Exam

The Final Exam will be held during Final Examination Period. It will be *closed book*. No crib sheet or electronic devices are permitted. No calculators. No cell phones. The exact format of the final exam will be announced in class and on myCourses.

Failing grade policy: *You must get a score of at least 50% on the final exam to pass the course.*[1] If your score is below this threshold, you will automatically receive an F grade for the course, regardless of your scores on the assignments and quizzes.

5.4 Supplemental/Deferred Exam

The Supplemental/Deferred exam will cover the same material as the Final Exam and it will replace the Final Exam grade, with the same grading policy as described above. For information on Supplemental Exams, see `https://www.mcgill.ca/science/student/general/exams/supplemental`.

[1] This threshold of 50% will be applied only if the class median grade on the final exam is 70% or higher. If the class median grade turns out to be lower, then the threshold will be lowered also.

5.5 Additional Work

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

5.6 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.

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 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 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 Piazza 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 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`

6.3 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.

6.4 Cheating on quizzes

The quizzes will be online and we will let you do them with a 12 hour time interval, and wherever you like (from home, library, Trottier labs, etc). 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 12 hour time limit for finishing the quiz is complete is cheating and is absolutely forbidden.

7 McGill language policy

In accord 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.

`https://www.mcgill.ca/study/2017-2018/university_regulations_and_resources/undergraduate/gi_lang_policy`

8 Required Software

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/), a more powerful IDE which can assist you in writing your code. All instructors and teaching assistants will provide support for Eclipse.

The JDK is installed on the computers in the SOCS laboratory, as is Eclipse. You are encouraged to install the JDK and Eclipse (or the IDE of your choice) 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 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`

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. As well, some special tutorials will be provided:

Tutorial	Title	Contents
T0	Basics of Course Software Tools	The JDK, and Eclipse: creating, load- ing, editing, saving, compiling, and running programs. How to use the debugger in Eclipse.

Tutorial	Title	Contents
TF	Final Exam Review	Review of all material for the final.

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.

9.2 Approximate Schedule of Topics

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

	Week	Topics	Events
Preliminaries	1–Jan 6	What is an algorithms? Java Syntax Primitive data types, char, and Unicode Variable declarations and scope Primitive type casting	
	2–Jan 13	Arrays Reference types Errors and Exceptions	
OOD in Java 1	3–Jan 20	Modifiers, attributes, and constructors get/set methods, Mutable vs Immutable Packages, and UML diagrams	
	4–Jan 27	Inheritance, and modifiers Object class and type conversion Polymorphism, and Abstract classes	Quiz 1
Linear Data Structures	5–Feb 3	Array lists Singly linked lists Doubly linked lists	A1 due
	6–Feb 10	Quadratic sorting algorithms Stacks Queues	Quiz 2
OOD 2	7–Feb 17	Interfaces Iterable, Iterator, and Comparable Induction	

	Week	Topics	Events
Recursion	8–Feb 24	Recursion Binary search Merge sort and quick sort	A2 due Quiz 3

	Week	Topics	Events
Non-Linear Data Structures	9–Mar 9	Trees Tree traversals Binary trees	
	10–Mer 16	Binary search trees Heaps	Quiz 4
	11–Mar 23	Maps and hash codes Hash maps Intro to graphs	A3 due
Analysis of Algorithms	12–Mar 30	Graph traversals Asymptotic notations Big O, Big Omega, Big Theta	Quiz 5
	13– Apr 6	Limits Recurrences	A4 due
	14–Apr 13	Review	