McGill

Course Outline

Course Name: Programming Challenges
COMP 321 Winter 2021

Instructors: Section 1 - David Becerra

Office Hour: W 16:00
Email: david.becerra@mcgill.ca

Welcome to COMP-321! Please read this document carefully and keep it for reference throughout the term.

Course Format:
This course consists of nine 1-hour lectures, and three 3-hour programming contests. The lectures and contests will be held online. In addition to the 18 hours of in-class work, the students will be given weekly training problems to exercise their skills at home, which brings the total amount of work for the course to approximately 45 hours.

- Lectures (9): Mondays 8:35 - 9:25 (via zoom)
- Laboratories (3): Mondays (Feb 15, Mar 15, Apr 12)
    - 24h window - 3.5 hours individual contest (via mycourses + kattis)

Course Goals:

- Give students the opportunity to test their algorithm design and programming skills on tricky problems and puzzles.
    - Given a problem, we want to:
        - Solve it efficiently by using algorithms and data structures.
        - Convert our solution into a program.
        - Do it as quickly as possible (under pressure)
        - And do it correctly (without bugs)
- Encourage students to join the McGill team for the programming contest.
- To have **FUN!**

Comp321 is right for you if:

- You are looking for a fun course that puts a fresh face on standard topics in programming and algorithms. OR
- You are planning to apply soon to a job interview in a big company. OR
- You are looking for a 1-credit course, but you are willing to work a bit more than the normal 1-credit course load. OR
- You are interested in joining the McGill team to participate in diverse programming contest competitions. OR

- You are simply motivated by the thrill of competition and learning.

Pre-requisites:

- COMP-250, or COMP-206
- MATH-240 and MATH-223
- COMP-251 will be an asset
- Or eagerness to learn (please consult the instructor if you want to discuss pre-requisites)

Required Software:

- You can solve the proposed exercises in any of the following programming languages: Java, Python and C/C++.

Textbook:

There is no required material; however, I recommend the following material:

BOOKS

- Skienna, S; Revilla, M., Programming Challenges, Springer Verlag, 2003. ISBN: 0-387-00163-8.n
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms (Third Edition), MIT Press, Cambridge, MA, 2009
- Principles of Algorithmic Problem Solving. Johan Sannemo, 2018.
- Competitive Programming by Steven Halim (https://sites.google.com/site/stevenhalim/)
- Any other good textbook on algorithms and C++, Java and/or Python reference

ONLINE CONTESTS:

- TopCoder
- Google Code Jam
- CodeChef

TRAINING PROGRAMS:

- USACO training program.

ONLINE JUDGES:

- https://open.kattis.com/
- http://www.spoj.com/
- https://uva.onlinejudge.org/

COLLEGIATE CONTESTS:

- https://icpc.baylor.edu/
- http://www.ioinformatics.org/index.shtml

Course Outline:

Lecture	Date	Topic
L1	Jan-11	Introduction
L2	Jan-18	Data Structures
L3	Jan-25	Problem Solving Paradigms I (Brute Force)
L4	Feb-01	Problem Solving Paradigms II (Dynamic Programming)
L5	Feb-08	Problem Solving Paradigms III (Greedy)
L6	Feb-15	Programming Contest (L1 - L5)
L7	Feb-22	Graph Theory
L8	Mar-08	Strings
L9	Mar-15	Programming Contest (L1 - L8) [emphasis L7-L8]
L10	Mar-22	Computational Geometry
L11	Mar-29	Algebra, number theory and combinatorics
L12	Apr-12	Programming Contest (L1 - L 12) [emphasis L10-L11]

Course Grading:

- **55%** for 11 assignments [5% each]. (9 home training problem sets + 2 postmortem contest).
- **45%** for 3 programming contests. (17% best, 15% medium, 13% worst).

The grading will be given by the online judges. Judging is relentlessly strict and the grading will be based on the online judge acceptance criteria. There is no re-grading for this course.

General Information

Communication:

- **My Courses**: All official communication, including announcements, lecture material, assignments, grades will be found on My Courses.
- **Course Discussions**: TBD.

- **Private Email**: The professor and TA have private email accounts that you may also use, however these communication channels are for personal queries.
- **Office Hours**: Please take a look at all posted office hours. Come (i.e., connect via zoom) to those times without appointment.
- **After lecture**: Some optional time will be available just after class to ask questions. I do not guarantee the length of this time since other constraints may interfere.
- **Email Policy**: E-mail is one of the official means of communication between McGill University and its students. As with all official University communications, it is the student's responsibility to ensure that time-critical e-mail is accessed, read, and acted upon in a timely fashion. If a student chooses to forward University e-mail to another e-mail mailbox, it is that student's responsibility to ensure that the alternate account is viable. Please note that to protect the privacy of the students, the University will only reply to the students on their McGill e-mail account.

Assignments & Programming Contests:

- **Assignments**:
  - All assignments are submitted to and picked-up from myCourses.
  - After the lecture, you get a set of problems about that topic.
    - Each set has approximately 6 problems.
    - Each problem has a varying difficulty.
    - Each problem is assigned some amount of points based on its difficulty.
    - To get a perfect score you need to get at least a certain amount of points. Usually representing the solution of 2 problems.
    - The grade follows linearly from the number of points you get.
    - The deadline of a problem set is the following class (Monday next week).
    - Late submissions will not be accepted.
    - Individual solutions will not be partially graded (i.e., they must be accepted by the judge).
- **Late Policy**: Due date/time, location/mode for returning your solutions, and accepted formats are announced for each assignment. Failure to return your assignment in time will result in penalties or even absence of grading. Late submission of 24h or less will receive a penalty of 25%. In all other cases, your assignment will be refused and not graded.
- **Programming Contests:**
  - The programming contests sessions allow the students to apply their problem solving skills on a set of problems and puzzles using either C, C++, Java or Python.
    - The lab sessions will consist of approximately 11 problems of varying difficulty. The idea of the contest is to solve as many problems as possible during 3 hours.

    - Each student will get a randomly selected set of problems. It means that it is improbable that you will get the same 11 problems to solve than a classmate.
    - To get full marks you do not have to solve ALL the questions. This number will be defined later in the course when the instructor has more information regarding the current programming level of the registered students. Sometimes, this number is defined once the contest has passed. The grade DOES NOT follow linearly from the number of exercises you solve.
- **Cheating/Collaboration**: I greatly encourage you to discuss the assignment with each other student in the class (except during the programming contests). However, these discussions should not so far be that you are sharing code or giving away the answer. A rule of thumb is that your discussions should be considered public in the sense that anything you share with a friend should be sharable with any student in the class. It is highly possible that you will find solutions to the contests on-line. Please do not copy the answers. You are registered in this course to train your problem solving skills and copying code from others does not help you in this goal. You can copy/paste general algorithm code, but you should understand the code that you are copying.

Additional Information:

The course slides are not meant as a complete set of notes or a substitute for a textbook, but simply constitute the focus of the lecture. Important gaps are left in the slides that are filled in during class, thus **l**ecture attendance should be considered essential.

Academic Integrity: *Code of Student Conduct*

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

L'université McGill attache une haute importance à l'honnêteté académique. Il incombe par conséquent à tous les étudiants de comprendre ce que l'on entend par tricherie, plagiat et autres infractions académiques, ainsi que les conséquences que peuvent avoir de telles actions, selon le Code de conduite de l'étudiant et des procédures disciplinaires (pour de plus amples reseignements, veuillez consulter le site www.mcgill.ca/integrity).

Student Rights and Responsibilities:

Regulations and policies governing students at McGill University can be downloaded from the website: https://www.mcgill.ca/students/srr/

Students Services and Resources:

Various services and resources, such as email access, walksafe, library access, etc., are available to McGill students: https://www.mcgill.ca/studentservices/

Various services and resources are offered to computer science students: https://mcgill-csus.ca/

Minerva for Students: http://www.mcgill.ca/minerva-students/

Important Note:

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.

Land acknowledgement:

McGill University is on land which has long served as a site of meeting and exchange amongst Indigenous peoples, including the Haudenosaunee and Anishinabeg nations. We acknowledge and thank the diverse Indigenous people whose footsteps have marked this territory on which people of the world now gather. Please see here for more details: https://www.mcgill.ca/edu4all/other-equity-resources/traditional-territories .