McGill School of Computer Science
www.cs.mcgill.ca

Course Outline

Course Name: Introduction to Software Systems
COMP-206 Winter 2021

	Section 1	Section 2
Instructors:	Joseph Vybihal "Joseph V"	Joseph D'Silva "Joseph D"
Office:	ENGMC 323	ENGMC 205N
Office hours:	Thu 10:00-11:00 EST or by appointment	Mon 11:00 – 12:00 EST or by appointment
Zoom link:	https://mcgill.zoom.us/j/88908819599	https://mcgill.zoom.us/j/93352787766
Home page:	http://www.cs.mcgill.ca/~jvybihal	https://www.cs.mcgill.ca/~jdsilv2/
Prof. Email:	jvybihal@cs.mcgill.ca	joseph.dsilva@cs.mcgill.ca

Course Discussions: Piazza: http://piazza.com/mcgill.ca/winter2021/comp206

Course Objectives: COMP-206 is a 3-credit full semester course in Software Development under the *nix(Unix/Linux) environment. It is offered in both the fall and winter semesters and is a required course for students in many of our degree programs. It provides a comprehensive introduction to and overview of the C programming language and how to use it with the *nix environment to build software.

This course focuses on System Application Development, which relates to the integration of differing software, programming languages and environments into a single application. In this light the course also teaches programming in Bash and interfacing with the operating system.

COMP 206 sets the stage for follow-on courses COMP-273, COMP-307 and COMP-310, plus others.

This course also gives the student basic Software Management skills in the form of the GNU tool set and Common Code Management techniques.

Course Description: Introduction to the *nix Environment, various commonly used *nix tools and utilities, shell scripting. Comprehensive overview of programming in C, use of system calls and libraries, debugging and testing of code; use of development tools like make and version control systems.

Texts:

Primary Text:

Software Systems ed3; Vybihal & Azar; Kendall/Hunt; ISBN 978-0-7575-9514-1.
You can purchase the textbook from here:
https://he.kendallhunt.com/product/software-systems

We will use the textbook in the course.

Supplementary Texts:

- GNU Software; Louksides & Oram; O'Reilly; ISBN 1565921127 (free on web)
- Just Enough Unix; P.K. Anderson; McGraw Hill; ISBN 0697131726
- C Programming Language; Kernighan & Ritchie; Prentice-Hall; ISBN 0131101633

Lectures:

Lectures will be delivered live through zoom (accessible through the mycourses portal of the course). Please note that you must sign-in to zoom using your mcgill (@mail.mcgill.ca) account to attend live lectures. The recordings will be available for later viewing under the lecture recordings menu in mycourses.

The expectation is that students either attend lectures or watch the recordings regularly (within 24 hours). "I need to catchup on lectures" is not an excuse for granting extensions for any assessments.

Q&A during live lectures will be handled strictly through Piazza's Live Q&A feature. This is to minimize disruptions during the live lectures as online lectures for large class sizes do not lend well for traditional Q&A format.

Evaluation:

Assignments	25%	5 Mini-Assignments (5% each)
2 online tests	30%	Online exam. (15% each)
Final Project	35%	Offered during the regular exam period
Lab Attendance	10%	Approx. 1 lab per week (1 hour).

Students who does not acquire at the least 50% marks on Final Project will receive an F irrespective of rest of their performance in the other assessments. The Final project covers all of the course topics.

Students with 80% or more lab attendance will be given full 10% grade. Attendances below that will be given points only proportional to their attendance (for example a student with 70% attendance will get only 7% grade). Only attendances of 45 minutes and above will be counted towards lab attendance. Anything below will be counted as 0. TAs will automatically record your attendance in Zoom.

The decision to make lab attendance mandatory was made after observing that many students are struggling with basic concepts while working on their assignments (spending longer hours on them) which could have been easily acquired by attending corresponding labs.

Students who get an A on BOTH class tests will have the lab attendance penalty waived. This is because, the purpose of the labs is to ensure that the students who are struggling with the material spends more time working on the practical problems that help them improve their standing.

Grading: All software solutions for assignments and the final project must compile with zero errors and must run to be graded. It does not need to run correctly (i.e., can have functional errors) for grading, but it must run. If your program compiles with errors or does not run at all, then you will receive zero points. The grader will not fix your code or look at the source code to give you partial grades.

If your assignment description requires you to turn in actual source code (not screenshots), no points will be given to screenshots of code. Learn how to move code from the SOCS servers to your laptop and vice versa as part of your initial labs / assignment.

Late work: You will be notified in advance of assignment due dates. All assignments are due on My Courses at the indicated time and date. Late assignments will lose 15% of its grade per day late (rounded up). Assignments beyond 2 days late will not be accepted. You may not submit assignments via e-mail without the permission from the instructor.

Other midterms, assignments, etc., do not constitute valid excuses for extensions as this is something every student has to work with. If you are not able to manage the workload you should consider withdrawing from one of the courses or consult your academic advisor.

Planned events (weddings, job interviews, etc.) also do not constitute reasons for extensions. It is the responsibility of the student to gather enough documentation of proof for any valid emergencies. Decisions for such cases will be made under the sole discretion of the instructor.

In some valid cases, instructor may choose at their discretion to shift the weight of your missed deliverable to something comparable that does not compromise the evaluation objectives of the course. This may also take the form of an oral examination.

Students with disabilities: Please ensure that you are registered with OSD. Any special accommodation requests should come to us only from OSD and not directly from the student. The OSD office is responsible to ensure that the student's needs are valid in accordance with the Univ. guidelines and conveying the allowed accommodations to the instructors. Such accommodations are usually limited to exams/tests. Anything outside of that would require a "reasonable accommodation" request process from the OSD.

Additional Work: Students with grades of D, F or J will not be given the opportunity to complete additional work to upgrade their grade.

Supplemental Exam: There is no supplemental exam for this course.

Re-grading: Mistakes can occur when grading. Not surprisingly, requests for re-grading always involve those mistakes in which the student received fewer points than they deserved, rather than more points than they deserved. With that in mind: if you wish me to re-grade a question on an exam or assignment, I will do so. I reserve the right to re-grade other questions as well.

Cheating/Collaboration: Collaboration is encouraged but your discussions should be public in the sense that anyone including the professor should be allowed to listen in. Assignments are original works created by the student alone. You are permitted and encouraged to have conversations with other students concerning the contents of the assignments and how to do them, but your work must be original. If two or more assignments are found to be identical (or portions of assignments) then all parties will lose points. This includes the student who permitted their assignment to be copied. This includes written solutions and software source code. The students will be reported to the university plagiarism department and their course letter grades maybe withheld till a decision is made.

Please keep in mind that your interaction with mycourses is recorded and can provide us with various digital information that helps us zero-down on students who might be potentially collaborating. In no instance a case of cheating will be put forward without manual verification of the submissions and facts by the instructor.

Reusing the entire code from the public domain is not acceptable. Whenever parts of code are borrowed from the public domain, include the source (e.g. URL) in the comments of the code (citation).

You must be able to explain what your code is to the instructor / TA if we ask you to. We reserve the right to do so if we suspect plagiarism. Not able to explain your code to us is a valid reason to lose points for the work.

Software mechanisms maybe used to detect plagiarism cases.
You are not allowed to share your assignment code with other students or put them in public domain.

You may reuse the code provided to you by the instructor and TAs in lecture slides, assignment solutions (including your own), labs, textbook, etc.

Labs:

Students must sign up for one of the several available lab slots depending on their schedule by going to mycourses -> groups and picking a lab slot that suits them (will be announced when the signup process opens up). Please keep in mind that the number of available spots per lab slot is limited to minimize over-crowding. Therefore, you are requested to sign up at the earliest. Labs are tentatively scheduled to start on Jan 21st. Depending on the course progress, there will be 9 – 11 labs scheduled in the semester.

As mentioned before, labs are mandatory and covers 10% of your course grade. Its purpose is to give you more hands-on practice of lecture topics and "warm-up" your skills to reduce your effort in doing assignments. It is highly recommended to attempt the labs on your own before going to the lab. Even if you are able to comfortably solve the lab problems on your own, we

recommend that you attend labs regularly as you might learn a different (or perhaps an efficient) approach to solving the problem from there.

You are expected to have already attended the lecture/watched the recording before going to the lab. TAs WILL NOT cover lecture topics again for the lab due to time constraints. Lab slots will not be used to for any assignment related questions either.

Lab Slot	Schedule	TA Name	TA email
1	TBD	Anirudha Jitani	anirudha.jitani@mail.mcgill.ca
2	TBD	Anirudha Jitani	anirudha.jitani@mail.mcgill.ca
3	TBD	Jenny Wan Yi Jiao	jenny.jiao@mail.mcgill.ca
4	TBD	Marek Borik	marek.borik@mail.mcgill.ca
5	TBD	Abraham Yesgat	abe.yesgat@mail.mcgill.ca
6	TBD	Safa Alver	safa.alver@mail.mcgill.ca
7	TBD	Ayrin Ahia-Tabibi	airin.ahia-tabibi@mail.mcgill.ca

Communication

My Courses**:** All official communication, including announcements, lecture material, assignments, grades will be found on My Courses. Students are responsible to keep themselves up-to-date with the announcements and emails send by the instructors and TAs.

Course Discussions: The online free tool, piazza.com, is used as our course discussion board. Please make sure to enroll in the Winter 2021 COMP 206 course on Piazza. Use this as your primary communication medium, since your questions are public and can help other students. If your question involves sharing a significant amount of your assignment source code, either go to a TA office hour or use a private post to instructors in Piazza. Keep in mind that last minute questions may not get a response in time to help you. Therefore, always start the work well ahead of your assignment due date. TAs are also NOT responsible to make your code work. They will offer you suggestions on how to debug, etc. It is YOUR assignment and it should demonstrate that you are able to master the content and build a software artifact with minimal help from others. For final project, any help will be limited to clarifications on the project description. There will be no debugging help provided by the TAs or the instructor.

Private Email: The professor and TA have private email accounts that you may also use, however these communication channels are for personal queries. For example: if you have a problem with your grade then email the TA who graded you directly, do not email the prof.

Appointments: Please email me directly to book at appointment outside office hours. Please keep in mind that such appointments are made for extenuating reasons, as I have a tight schedule throughout the week. You can also email me about your issues anytime if this is something that can be quickly addressed over an email. All assignment / lecture topic questions, however, must be addressed through piazza or office hours.

Office Hours: I have posted office hours. Come to those times without appointment. For general course topic related questions, you can go to either instructor's office hours. For procedural matters, please visit the instructor responsible for your section.

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.

```
CommunicationAlgorithm() :
        if (public) piazza(); // all will benefit
        else if (about marks) emailTAPrivately();
        else if (medical or special) emailProfPrivately();
```

Your Grading TA

Each student is assigned a single TA who will be "their" grading TA for the entire course. This TA will be responsible for grading your assignments. This TA is not necessarily the same as your Lab TA (but could be).

You can attend any TA office hour for help.

The table below identifies which TA you have been assigned to for grading purposes.

Tentative TA Group by Student Last Name

#	Last, First name Student	TA Name	TA Email Address
1	A - Bergeron, J	Anirudha Jitani	anirudha.jitani@mail.mcgill.ca
2	Bernard Litta, A - Chong, Chris	Anirudha Jitani	anirudha.jitani@mail.mcgill.ca
3	Chow, Alexa - Fafard, Finn	Jenny Wan Yi Jiao	jenny.jiao@mail.mcgill.ca
4	Fan, Yuhe - Hart, Jake	Marek Borik	marek.borik@mail.mcgill.ca
5	He, Chang - Kapur, Nimish	Abraham Yesgat	abe.yesgat@mail.mcgill.ca
6	Kern, Amanda - Liu, Siyu	Ayrin Ahia-Tabibi	airin.ahia-tabibi@mail.mcgill.ca
7	Liu, Hansen - Mouazer, Janis	Ayrin Ahia-Tabibi	airin.ahia-tabibi@mail.mcgill.ca
8	Moumneh, Mou. - Picard, Francis	Xinyi Xie	xinyi.xie@mail.mcgill.ca
9	Piccone, Cameron - Schull, Gab.	Olamilekan Fadahunsi	olamilekan.fadahunsi@mail.mcgill.ca
10	Scott, Mat. - Tian, Shu	Olamilekan Fadahunsi	olamilekan.fadahunsi@mail.mcgill.ca
11	Timofeeva, Anna - Wazin, Sajad	Shaurya Mehta	shaurya.mehta@mail.mcgill.ca
12	Wei, Andrew - Yu, Jiangshan	Jia Lin Tian	jia.tian@mail.mcgill.ca
13	Yuan, Lin - Z	Alessia Woolfe	alessia.woolfe@mail.mcgill.ca

Tentative Course Schedule

3 Lectures per Week

LECTURE DESCRIPTION	CLASS WORK
Unit 1 – Course Introduction	
INTRODUCTION TO SOFTWARE SYSTEMS Introduction to the course. What is this course about? Importance of Systems and command-line development.	Course outline Textbook: Chapter 1
Unit 2 – The Unix Environment	
THE UNIX/LINUX OPERATING SYSTEM The story of Unix. The architecture of Unix OS. Getting access to the SOCS Linux servers. SFTP and SSH clients, logging in, file transfer, working from home and school	Textbook: Section 2.0
THE SHELL The OS shell environment. The command-line prompt. Home vs root. Basic commands: ls, cd, mkdir, rmdir, cp, mv, cat, more, man, logout, paths.	Textbook: Sections 2.1-2.2
REGULAR EXPRESSIONS AND WILD CARDS Command-line commands that use wild cards and regular expressions. Using redirection. Using grep.	Textbook: Sections 2.2-2.3
VIM & DEVELOPER TECHNIQUES Non-GUI editors and their importance. Common developer techniques: directory structures, procedures, commands: chmod, tar, zip, backups.	Textbook: Sections 2.2-2.3
Unit 3 – Bash Programming	
INTRODUCTION TO BASH SCRIPTING What is BASH programming? When to use BASH programing? Examples of simple Bash programs.	Textbook: Section 2.4 Example 1
BASH EXPRESSIONS Variables (bash, shell, session), math expressions, and I/O.	Textbook: Section 2.4 Example 2
BASH CONTROL STRUCTURES Conditions, iteration, and functions.	Textbook: Section 2.4 Examples 3 – 4
BASH DEVELOPER TECHNIQUES Bash as an aid to developers, development environment initialization, standardizing operations (archiving and backups, etc.) Debugging shell scripts. **NEW**	
ADVANCED UTILITIES **NEW** find, sed, awk. sort	
SCHEDULING at, crontab **NEW**	
SESSIONS AND BASH SCIPTS The Session. Session memory. Customization. System vs session scripts.	
TEST #1	TEST #1 – Units 1 to 3
Unit 4 – C Programming	
INTRODUCTION TO THE C LANGUAGE The story of C. Why C? Children of C. Hello World example with puts() and getc(), GCC basics, compiling, running, errors. Bash compiling scripts.	Textbook: Section 3.0 Example 1
DATA AND CONTROL STRUCTURES IN C Types, variables, expressions, conditions and iteration.	Textbook: Section 3.0 Example 2 & 4
STDIO.H and STDLIB.H getchar, putchar, puts, printf, scanf, sprintf, sscanf. I/O issues and data validation. STDIN, STDOUT and STDERR.	Textbook: Section 3.0 Example 3, 3.1
ARRAYS AND STRINGS Array, strings, static & invariant data, writable data, array addressing.	Textbook: Section 3.0 Example 5
POINTERS, STRINGS, AND STRING.H Pointer referencing and de-referencing. Example: make string.h functions using pointer referencing.	Textbook: Section 3.0 Example 6
FUNCTIONS AND SCOPE Function syntax, scope rules. Call-by-value and call-by-reference.	
STRUCT AND UNION	Textbook: Section 3.0 Example 7

Struct and Union syntax. Array of struct.	
DYNAMIC MEMORY Dynamic arrays, dynamic structs, linked lists.	Textbook: Section 3.0 Example 8
SEQUENCIAL TEXT FILES The file concept. Streams. Text and CSV files.	
TEST #2	TEST #2 – Unit 4
Unit 5 – Basic Software Development Techniques	
MODULAR PROGRAMMING C object files, compiler performance, team programming basics, about large projects. The extern expression. The Pre-processor.	Textbook: Section 3.0 Example 9, 4.0
GNU TOOLS The makefile. The profiler. The GDB.	Textbook: Section 4.1
REPOSITORIES What is a repository? Ways to use repositories. Using git. More team programming basics. Branching.	Textbook: Section 4.1
Unit 6 – Systems Programming	
INTRODUCTION TO SYSTEMS time.h, Bit-wise operations. Bash to C parameter passing. Void * referencing. About machines.	Textbook: Section 3.0 Example 6
CONCURRENT PROGRAMS Shell memory-based communication, ampersand operator, ps, kill, and pwd.	Textbook: Section 4.2
INTER PROCESS COMMUNICATION What is a process. C process creation: system and fork. Producer Consumer problem.	Textbook: Section 4.2
BLOCK FILES AND RANDOM FILES Sequential Block, Random and Binary files (fread, fwrite, fseek).	
Time Permitting	
NETWORKS AND WEBSITES Basic network architecture & the SOCS web server.	Textbook: Sections 5.0 – 5.1
INTER PROCESS COMMUNICATION WITH CGI Using CGI to communicate with C using the Internet.	Textbook: Sections 5.2-5.4
MORE CGI and C	Textbook: Sections 5.5-5.7
Unit 7 – If Time Permits…	
VOID * AND FUNCTION *	
SIGNAL HANDELING	
SOCKET COMMUNICATION	
ASSEMBLER WITH C	

General Course Information

Course Requirements: The pre-requisite for this course is COMP-202 or COMP-250.

Right to submit in English or French written work that is to be graded

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.

Classroom Rules: All electronic devices (cell phones and beepers) must be turned off or left on silent mode during class time.

Assignments Pickup: All assignments are submitted to and picked-up from My Courses.

Computing Resources: Trottier 3rd floor. The technicality of remotely accessing them from your home will be discussed and demonstrated in the class and labs.

SOCS Unix Account: IMPORTANT !! All assignments, labs and final project work have to be performed by the students in the SOCS computing infrastructure and NOT on your personal computers. As such, it is important to apply for an SOCS Unix Account at the earliest. It is highly discouraged to use your personal computer for any programming work as it will put you on a significant disadvantage as the course progresses as well as when you have to take advanced system courses that requires you to use a specialized software infrastructure (same with several software engg. Professions).

Apply for the account here: https://newuser.cs.mcgill.ca
You must be on McGill Wifi to access the above website or on a McGill VPN
https://mcgill.service-now.com/itportal?id=kb_article&sysparm_article=KB0010687
Contact McGill IT if you have issues accessing VPN (not the CS help desk)
Forgot your Username and/or Password?
Reset it at https://newpassword.cs.mcgill.ca/

SOCS helpdesk - help@cs.mcgill.ca (For any issues regarding your Unix account).

Examinations and Grading:

Students are responsible for all materials for the tests and exams, whether it is covered in class. Exams will be a combination of all types of questions based on all sources, and students may be required to integrate theoretical concepts from the text to substantiate their arguments.

No make-up tests or make-up assignments are allowed in this course. There is no supplemental exam.

If you are not satisfied with the grading of an assignment or mid-term test, you may request a review within 7 days of your grade being published. Indicate in writing or during a meeting with the TA/instructor where and why you feel the marks are unjustified and give it back to your TA/instructor for re-grading. Note that at the discretion of the TA/instructor, the entire assignment or mid-term test may be re-graded, and your grade can go up or down (or stay the same) accordingly.

Calculators
Only non-programmable, no-tape, noiseless calculators are permitted. Calculators capable of storing text are not permitted in tests and examinations.

Dictionaries
Dictionaries are not permitted, but translation dictionaries are.

Handheld Devices
Handheld devices capable of storing text and having calculator functionality (e.g. Palm, etc.) are not permitted.

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 lecture attendance should be considered essential. It is also extremely important to practice the topics discussed in lectures by yourself. Systems courses are like sports. You do not become a good athlete by watching it, but by continuously practicing it.

The material covered in the classroom will be used to supplement textbook readings.

Copyright:

All the materials provided to you for lectures, assignments, labs, project, etc. are copyright of the respective person who publishes it.
Students are not allowed to pass these materials to anyone else, including posting to third party websites.
Students are allowed to retain such materials for their personal use, including in their personal cloud storage, private repositories, etc.
Violators can be subject to legal and University disciplinary procedures.

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 renseignements, veuillez consulter le site *www.mcgill.ca/integrity*).

Final Exam/Project Policy: *Regulations*
Students should not make other commitments during the final exam period. Vacation plans do not constitute valid grounds for the deferral or the rescheduling of examinations. See the Centre Calendar for the regulations governing Examinations:
http://www.mcgill.ca/student-records/exams/regulations/

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.

Students Rights and Responsibilities:

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

Students Services and Resources:

Various services and resources, such as email access, walksafe, library access, etc., are available to students:
http://www.mcgill.ca/student-records

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

Note: In the event of extraordinary circumstances beyond the Instructor/University's control, the content and/or evaluation scheme in this course is subject to change.