

# Welcome to COMP 206

This is the course outline for `#COMP206` in Winter 2018 term. Welcome to the course!

## Defining the Course

COMP 206 - Software Systems is a course that teaches important aspects of low-level programming that are crucial to creating pieces of software that can interoperate across different computers, different types of devices, and online across the world. We study these topics conceptually as well as through hands-on creation of software in the C programming language. By the end of the course, you should have a strong ability to program in C generally. More importantly, you should be able to “Think in C”, which is often a helpful way to get insights on the way that operating systems and more recent programming languages work.

## Objectives

- Become familiar with common challenges in creating software systems and best-practice solutions
- Understand how software runs on the Linux operating system, including the structure of a running process, scheduling, user and file permission and networking
- Experience implementing software system elements using C programming language
- Experience setting up and maintaining a software system with Linux tools, Github and CMake

## Topic List

- Reasons for being of software systems. Examples and history.
- Study of a modern OS - Linux operating system, POSIX, standard Linux tools
- Aspects of system software in C:
  - Input and output to files and terminal
  - Libraries and linking
  - Memory allocation
  - Pointers
  - System calls
  - Socket communication
  - Multi-process software systems - synchronization

- Web-based software
- Case studies and example problems

## Term Schedule

Week starts	Topic	Assessments (estimated)
08-Jan	Modern Software Systems	A0
15-Jan	Basics of C	A1
22-Jan	Strings and Data	
29-Jan	IO and Memory	A2
05-Feb	Libraries and Multi-file Programs	
12-Feb	System calls	Midterm 1
19-Feb	Linux, GNU, POSIX	A3
26-Feb	Automation by Scripting	
05-Mar	Reading week	A4
12-Mar	Multi-process programs	
19-Mar	Security	Midterm 2
26-Mar	Networking	A5
02-Apr	Web and Mobile	
09-Apr	Wrap-up and final review	

## Instructor and TA Information

### Instructor Info

David Meger: Office McConnell Engineering 112 N (north wing)

Office hours: Tuesday: 4-5

Thursdays 1-2:20

Email: [david.meger@mcgill.ca](mailto:david.meger@mcgill.ca) (please try My Courses Discussion first for Assignment queries)

### Teaching Assistants

Name	Day of Week	Time of Day	Location	Email
Dave	Tuesday	16:00-17:00	MC112N	<a href="mailto:david.meger@mcgill.ca">david.meger@mcgill.ca</a>
Dave	Thursday	13:00-14:20	MC112N	<a href="mailto:david.meger@mcgill.ca">david.meger@mcgill.ca</a>
Joe	Wednesday	13:00-14:00	TR3090	<a href="mailto:zhaohui.su@mail.mcgill.ca">zhaohui.su@mail.mcgill.ca</a>
Shabir	Friday	15.30-16.30	MC224	<a href="mailto:shabir.abdulsamadh@mail.mcgill.ca">shabir.abdulsamadh@mail.mcgill.ca</a>
Jonathan	Wednesday	14:30-15:30	TR3090	<a href="mailto:jonathan.campbell@mcgill.ca">jonathan.campbell@mcgill.ca</a>
Prabhjot	Tuesday	9:00-11:00	MC226	<a href="mailto:prabhjot.sandhu@mcgill.ca">prabhjot.sandhu@mcgill.ca</a>
Xoey	Wednesday	10:00-11:00	MC202	<a href="mailto:xoey.zhang@mcgill.ca">xoey.zhang@mcgill.ca</a>
Xiru	Friday	15:30-16:30	MC224	<a href="mailto:xiru.zhu@mcgill.ca">xiru.zhu@mcgill.ca</a>
Vishal	Thursday	10:15-11:15	TR3090	<a href="mailto:vishal.jain@mail.mcgill.ca">vishal.jain@mail.mcgill.ca</a>
Pulkit	Wednesday	16:30-17:30	TR3090	<a href="mailto:pulkit.khandelwal@mcgill.ca">pulkit.khandelwal@mcgill.ca</a>

## Assessment

Assignments (5): 35%

Midterm Exams (2, dates to be confirmed): 25%

Final Exam (during official exam period): 40%

Assignments will have 2 due dates, the full-marks and the extended deadlines.

Submissions prior to the full-marks deadline will have no penalty. Submissions prior to the extended deadline will have a 10% deduction. No assignments will be accepted after the extended deadline. In cases of medical difficulty, exemptions can be granted. For example, with an exemption on A1, the marks of A2-A5 only will count for the total 35% (8.75% each).

## Texts and Resources

There is no required book for this course. Provided course notes and slides will be the core method for learning and studying the material. We will often point to useful resources online and sometimes even give assigned readings. In case of wanting additional background, each of textbooks can be helpful, in order of decreasing relevance:

1. The C Programming Language, 2nd Edition by Kernighan and Ritchie. ISBN: 0131103628
2. Software Systems by Vybihal and Azar. ISBN: 978-0-7575-5727-9

3. Learning the Unix Operating System by Jerry Peek, John Strang, and Grace Todino
4. C for Java Programmers by Tomas's Muldner. ISBN: 0201702797

## Diversity and Inclusion

COMP 206 is meant to be an introductory course to one of the most important areas of CS, and indeed one of the most important skill-sets that people will use to influence the world in our lifetimes. Our goal is to make this content equally accessible to students of all backgrounds and we work to pro-actively acknowledge and address any bias that may occur during the term. Equal treatment of students from every gender, race and orientation is a top priority. We openly welcome suggestions on how to improve inclusion, by contacting the TAs or instructor either with your name or anonymously.

## Examinations and Grading

All exams will be closed book, with a one-page crib sheet (double sided) allowed only and no calculators. The exams will be a mix of multiple-choice and written answers testing both the basic concepts of the course and the ability to create programs that exercise this knowledge.

No make-up exams are possible. In the case that you cannot write a midterm, with supporting documentation, the midterm weight can be instead moved to the Final Exam, making it worth up to 65%.

All of the TAs and instructor are humans (for now) and mistakes in grading can happen. Re-grading requests should be made first to the TA who marked the work. The request should include an email explaining where you think the confusion or error was made and some evidence that the marks are

## Policy on Plagiarism and Independent Work

**#COMP206** is one of the most programming intensive courses in our undergraduate program. While we will have written tests that ensure you know the material, the real intention of the course is to ensure you have the skills to write effective Software Systems yourself. These skills are needed in the workplace and in research to help you change the world and earn a good salary.

In order to get the experience and develop your own Software Systems skills, you must turn in programs that you have completed yourself. You can get tips and ideas from myself, TAs and other students, but you must do the software development yourself, which means typing in all of the code in a way you understand, running and debugging it and ensuring it passes the specifications given in the assignment document.

THINGS THAT ARE NOT OK: Copy/pasting directly from the code of other students. Sending the assignment to any paid or free service, either local or online for someone else to solve. Giving your code to others.

THINGS THAT ARE OK: You can copy/paste from the web if you find resources there, since that's exactly how many working programmers operate. You can take notes or phone images of code that the instructor or TA give in office hours. We plan to be very helpful there and will often give really good hints on how to set up the solutions, so we do recommend coming. We will also be fair and ensure you're left with enough parts to solve yourself afterwards. You can share notes taken at a TA session with other students, as long as you do not include your own code that you have written afterwards.

I hope everyone will follow these rules and we have a great term. We will be using automated tools and our own manual checks to ensure this is the case. In case of any detected plagiarism, we will proactively report through McGill's academic disciplinary procedure.

## Language Policy

As is McGill's policy, all work can be submitted in either English or French.