

## Outline

---

Course Name	Introduction to Software Systems COMP 206 – Fall 2016	
Course Personnel	Instructor: David Meger Office: MC 112N Office Hours: Mon 1:30pm and Tues 12:30pm Email: <a href="mailto:david.meger@mcgill.ca">david.meger@mcgill.ca</a>	Teaching Assistants office hours are held inside or nearby TR 3120 or 3110  Olamilekan Fadahunsi: Thurs 4pm Pulkit Khandelwal: Tue 3pm Senjuti Kundu: Wed 10am Ming Ma: Mon 2pm Sandeep Manjanna: Wed 11am Inderjot Ratol: Tues 10am Xoey Zhang: Thurs 10am
Course Description	<p>This course introduces the core concepts that enable today's <b>software systems</b> and provides hands-on experience with developing such systems. Students will create low-level systems programs that interact with the operating system, the user, other programs on the same device, and attached networks while maintaining efficiency and security. A number of high-level tools commonly used in modern software industry to manage software complexity will be utilized, such as build system configuration and software version control.</p> <p>Our main focus is on the UNIX operating system programmed with C, Python and associated scripting and configuration tools. The following is a rough outline and ordering of topics:</p> <ol style="list-style-type: none"><li>1. Introduction to Software Systems</li><li>2. The UNIX OS architecture and associated tools</li><li>3. Systems programming in C</li><li>4. Systems programming in Python</li><li>5. C and Python integration (mixed-language systems)</li><li>6. Networked and web-based software systems</li></ol>	
Texts	<p>There is no required text for this course as the technologies we study are extensively documented on-line. Links to websites for additional reading will be provided regularly in lectures .</p> <p>Students are encouraged to obtain optional text books for further background reading. Again, there are dozens of useful books in these areas, but we recommend the following selection:</p>	

1. **Software Systems** by Vybihal and Azar. ISBN 978-0-7575-5727-9
2. **The C Programming Language, 2<sup>nd</sup> Edition** by Kernighan and Ritchie. ISBN: 0131103628
3. **C for Java Programmers** by Tomasz Muldner. ISBN:0201702797
4. **Learning Python**, 5<sup>th</sup> edition by Mark Lutz and David Ascher. ISBN: 1449355730
5. **Learning the Unix Operating System** by Jerry Peek, John Strang, Grace Todino

**Evaluation**

Assignments (4 – 5):	40%
Midterm Exam (in class, mid-October):	20%
Final Exam (3 hours, exam period):	40%

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 5% of its grade per day late. Assignments beyond 2 days late will not be accepted. You may not submit assignments via e-mail without the permission of the instructor.

The teaching assistants will provide targetted review sessions related to the assignment material when possible.

One in-class review session will be conducted by the instructors for each of the Midterm and Final exam.

**Examinations and Grading**

Students are responsible for all materials for the tests and exams, whether or not 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.**  
A supplemental exam is possible for 40% of the grade (to replace your final 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 return. Grade adjustments can only be carried out when there is a clear demonstration that an error or mis-understanding has happened during the marking process.

*Calculators*

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

*Crib Sheets*

A one page 2-sided crib sheet will be allowed for both the midterm and final.

*Dictionaries*

Dictionaries are not permitted, but translation dictionaries are.

*Handheld Devices*

Handheld devices other than simple calculators are not permitted.

Pre-requisites

The pre-requisite for this course is COMP-202 or equivalent programming experience. With this in mind, programming will be brisk using all the advanced features present in C and UNIX for developing software.

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.

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](http://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](http://www.mcgill.ca/integrity))

Final Exam 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/>

Students are required to present their I.D. Card (with photo) for entrance to their examination.

*Conflicts*

If you are unable to write your final examination due to scheduling conflicts, you must submit a Final Exam Conflict Form with supporting documentation at least **one month** before the start of the final examination period. Late submissions will not be accepted. For details, see:  
<http://www.mcgill.ca/student-records/exams/conflicts/Exam>

*Timetable*

Examination schedules are posted at the Centre and on the following page approximately 6-8 weeks before the examination period commences  
<http://www.mcgill.ca/student-records/exams/>  
The Centre cannot provide examination dates over the telephone.

# Lecture and Assignment Schedule

Week	Topic	Description	Exercises and Assessments
<b>Unix as an example software system</b>			
1	<b>Unix Introduction</b>	Introduction to software systems General concerns and context History and justification for Unix	
2	<b>Working with Unix</b>	Unix architecture: kernel, filesystem, users Basic Unix concepts and command-line tools	Assignment 1
3	<b>BASH scripting</b>	Bash syntax, functionality and examples	
<b>Systems programming with C</b>			
4	<b>Intro to C</b>	Syntax, type system, file IO	Assignment 2
5	<b>Low-level C</b>	System calls,	
6	<b>Managing memory in C</b>	Pointers, structs, bit operations	Assignment 3
7	<b>Scaling up C programs</b>	Makefiles, libraries, multi-file compilation	
8	<b>Advanced C concepts</b>	Security and Memory Exploits Multi-threaded and multi-process	Midterm Exam
<b>Systems programming with Python</b>			
9	<b>Intro to Python</b>	Basic structure Introduction to interpreter vs compiler	Assignment 4
10	<b>Essential Python constructs</b>	Container types Iteration Variable scope	
11	<b>Classes and mixed C/Python programs</b>	Revisiting libraries and executable structure ctypes Python as a collection of C libraries!	Assignment 5
<b>Networked and internet software systems</b>			
12	<b>Network and web software architectures</b>	Client/server communications and sockets The architecture of the web CGI programming	
13	<b>Large Linux/C/Python example systems</b>	Lots more fun examples	Practice Final