

# Concurrent Programming

## COMP 409

McGill University, Winter 2018

### Course Details

**Time:** Tuesday, Thursday 10:05–11:25

**Place:** MAASS 217

**Instructor:** Professor Clark Verbrugge

**Office:** McConnell Eng., room 230

**Office hours:** Tuesday 14:30-16:00, Friday 10:00–11:30, or by appointment.

**Phone:** 514 398-2411

**Email:** [clump@cs.mcgill.ca](mailto:clump@cs.mcgill.ca)

**Teaching Assistants:** TBA

### Email, Website

McGill's MyCourses will be used for course announcements, to manage assignments and for online discussions. Students are expected to monitor their McGill email account for course-related news and information.

The external course website is: <http://www.sable.mcgill.ca/~clump/comp409>

### Pre-requisites

- COMP 251 (Data Structures and Algorithms).
- COMP 302 (Programming Languages and Paradigms).
- COMP 310 (Computer Systems and Organization) *or* ECSE 427 (Operating Systems).
- There is a non-trivial programming requirement; ability to program in C and Java will be required.

Students registering without the pre-requisite may find the course removed from their transcript by their Faculty. Please consult the instructor if you do not have all the pre-requisites.

### Textbooks

The following texts, along with your own course notes constitute reference sources for this course.

- Maurice Herlihy and Nir Shavit. *The Art of Multiprocessor Programming* (revised 1st edition).  
This is the primary text for the course. It does not cover all topics, but it is a good and relatively modern reference. It is available at the bookstore as a physical book, and in the library in e-book form.
- Gadi Taubenfeld. *Synchronization algorithms and concurrent programming*.  
A previously used text; it does not cover the same material, but it provides a different perspective.
- Brian Goetz et al.. *Java concurrency in practice*.  
An old, but still relevant practice-oriented text for Java programming
- Gregory Andrews. *Foundations of Multithreaded, Parallel, and Distributed Programming*.  
An ancient text, but a classic one; covers some of the basics.

## Evaluation

4 Assignments:	40%
Midterm:	10%
Exam:	50%

The exam will be open-book. A supplemental exam (50%) will be held if required.

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.

**Assignment and Exam Policy:** Assignments must be submitted on time. Late assignments will only be accepted in highly-exceptional circumstances, typically requiring a medical note as well explicit permission from the instructor. Note that I do not consider your workload in other courses exceptional, no matter what courses you take!

No assignment submissions will be accepted after marked assignments have been returned, or after solutions have been discussed. Students are expected to monitor their returned work. Corrections to assignment grades will not be made later than 2 weeks after an assignment grade is given out.

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 <http://www.mcgill.ca/integrity/> for more information).

In all cases, to be accepted **work submitted for this course must fully and completely represent your own efforts.** Copying assignments or tests, or allowing others to copy your work, whether whole or partial will not be tolerated.

## Course Content

Note that lecture dates are approximate: topics may shift and/or span lectures.  
Chapter(.section) readings from Herlihy and Shavit are shown next to topics

Lec#	DoW	Date	Topic	Readings
1	Tuesday	January 9	Introduction	1
2	Thursday	January 11	<b>No class!</b>	
3	Tuesday	January 16	Hardware, atomicity	B.2–B.5, B.7
4	Thursday	January 18	Mutual exclusion, Java/PThreads	2.1–2.3, A.2.1, A.4
5	Tuesday	January 23	Simple locks	2.4–2.6, 7.1–7.4
6	Thursday	January 25	Complex locks	7.5, 8.1–8.5
7	Tuesday	January 30	Termination, barriers, priority, TSD	17.1–17.6, A.2.4
8	Thursday	February 1	Deadlock, race conditions	
9	Tuesday	February 6	Expressiveness	4.1, 4.2, 5.1, 5.2
10	Thursday	February 8	Linearizability, scheduling	3.1–3.6
11	Tuesday	February 13	Hardware memory consistency	3.8
12	Thursday	February 15	Memory models: Java	
13	Tuesday	February 20	Memory models: C++	
14	Thursday	February 22	Concurrent data structures	6.1–6.4, 10.1–10.6, B.8
15	Tuesday	February 27	<b>Midterm (TBC)</b>	
16	Thursday	March 1	Concurrent data structures	11.1–11.4, 9.8
17	Tuesday	March 13	OpenMP	
18	Thursday	March 15	SPMD and PGAS	
19	Tuesday	March 20	Work stealing	16.1–16.5
20	Thursday	March 22	Transactional memory	18.1–18.4
21	Tuesday	March 27	Message-passing	
22	Thursday	March 29	Process algebra	
23	Tuesday	April 3	Process algebra	
24	Thursday	April 5	Dataflow	
25	Tuesday	April 10	Dataflow	
26	Thursday	April 12	Catch-up and/or review	

## Assignments

Expected assignment distribution dates and due dates are listed below. Note that this is mainly to help you in general planning; topic descriptions are vague and non-exhaustive, and both the topic and the associated dates may change. Be sure to consult MyCourses for final, official due dates.

Assig.	Main Topic	Available	Due
1	Basic concurrency and locking	Wednesday, January 24	Wednesday, February 7
2	Thread interaction and coordination	Wednesday, February 7	Wednesday, February 21
3	Concurrent data structures	Wednesday, March 7	Wednesday, March 21
4	High-level concurrency	Wednesday, March 21	Wednesday, April 4