Concurrent Programming
COMP 409B

McGill University, Winter 2008

Course Details

Time: Monday, Wednesday, Friday 10:35am–11:25am
Place: MAASS 328

Instructor: Professor Clark Verbrugge
Office: McConnell, room 230
Office hours: Tuesday 13:00–14:30, Friday 9:00–10:30, or by appointment.
Phone: 398-2411
Email: `clump@cs.mcgill.ca`

Teaching Assistant: Gregory Prokopski
Office: McConnell, room 234
Office hours: Tuesday and Thursday 10:30-12:00, or by appointment.
Email: `gproko@sable.mcgill.ca`

Email, Website

Students are expected to monitor their McGill email account for course-related news and information. The 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 or Java will be required.

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

Textbook

There is one required text for this course, and one recommended. There are also several further books on reserve in the Schülich library that are not essential but may be useful if you need additional references.

Required: *Synchronization algorithms and concurrent programming.* by *Gadi Taubenfeld*
Recommended: *Java concurrency in practice.* by *Brian Goetz et al.*

Others: *Foundations of Multithreaded, Parallel, and Distributed Programming.* by *Gregory Andrews.* This serves as a basic introduction to some of the main concurrent programming problems.
Multithreaded Programming with Java Technology by *Lewis* and *Berg* **or** *Multithreaded Programming with PThreads* by *Lewis* and *Berg*. These texts cover additional information on multithreaded programming; they are very similar, emphasizing either Java or POSIX (PThreads). The Java text is now dated and

superseded by the recommended text on Java programming. A further, more detailed book on PThread programming is *Threadtime* by *Norton* and *Dipasquale*.

Class examples will be primarily in Java, but unless otherwise stated students may complete assignments using either the C/C++ and PThreads environment/language, or Java. The last assignment will require use of Java.
The required and recommended texts are available in the bookstore. All texts are on reserve in the Schülich Library.

Description

Students will learn the fundamentals of programming for concurrency. This includes both theoretical foundations, and practical experience with multithreaded programs. Both Java and POSIX (PThreads) environments will be discussed in detail, and significant programming will be required.
Upon completion of the course, students should have a good understanding of concurrent programming constructs and how to use them effectively and correctly. Additionally, students will gain experience with several major theoretical concurrency modelling systems/paradigms.

Evaluation

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

Exams and assignments may be written in English or French. Both the exam and midterm will be open-book. A supplemental exam (100%) will be held if required.

Assignment and Exam Policy: Assignments must be submitted on time. Late assignments will only be accepted in highly-exceptional circumstances and only with **written** permission of the instructor. No assignment submissions will be accepted after marked assignments have been returned, or after solutions have been discussed in class.
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).
More specifically, **work submitted for this course must represent your own efforts.** Copying assignments or tests, or allowing others to copy your work, will not be tolerated. Note that introducing syntactic changes into a copied program or assignment is still considered plagiarism.

Course Content

Note: lecture dates are approximate and may shift. Readings from Taubenfeld are required and will be further supplemented with handouts and electronic resources. Goetz readings are recommended; other readings are optional.

Week 1–2	**Readings**
• Introduction.	**T1.1–1.2**[1], **G1,11.1–11.3**[2], A1[3]
• Hardware	LB$_J$15,16[4], LB$_P$16,17[5]
• Atomicity & Independence.	**T1.4**, **G2**, A2.1–2.5
• Intro to multithreading; Java & POSIX.	**T1.5**, **G3,4**, LB$_J$2–4, LB$_P$2–4, ND 1–4[6]
• Mutual Exclusion	**T2.1–2.2,2.4,4.1–4.3**, A2.8
Week 2–4	
• Mutual Exclusion (continued)	
• Locks & Barriers	**T2.3,3.2,5.1–5.2,5.8,15.2**, A3.1–3.4, LB$_J$6, LB$_P$6, ND5
Week 4–5	
• Semaphores (all the flavours)	**T4.6,8.1.3–8.1.4**, A4.1–4.3
• Condition Variables & Monitors	**T4.7,8.1.5–8.1.6**, A5.1, LB$_J$7, LB$_P$7
Week 5–6	
• Condition Variables & Monitors (continued)	
• Readers & Writers	**T8.2**, A4.4,5.4.3–5.4.5
Week 6–7	
• Concurrency problems	**T8.4–8.5**, A5.2
• Deadlock	**T7**, **G10**
• Scheduling & Priorities	LB$_J$5, LB$_P$5, ND6
Week 7–8	
• Termination & Suspension	**G7**, LB$_J$9, LB$_P$9, ND8
• **Midterm: March 3 (in class)**	
• TSD, Java & POSIX miscellany	LB$_J$8, LB$_P$8,10, ND9
Week 8–10	
• Memory Consistency	**G16**
• The Java concurrency API	**G6,8,13**
• Concurrent data structures	**T4.5**, **G5,14**
Week 10–11	
• Message passing	A7.1–7.5
• Formal models: CSP	A7.6
Week 12–13	
• Alternative language designs: Dataflow	
• Transactional concurrent programming	

[1] ie Taubenfeld book sections 1.1 through 1.2.
[2] ie Goetz book chapters 1, and 11.1 through 11.3.
[3] ie Andrews book chapter 1.
[4] ie Lewis & Berg Java book, chapters 15 and 16.
[5] ie Lewis & Berg PThreads book, chapters 16 and 17.
[6] ie Norton & Dipasquale, chapters 1 through 4.