Department of Computer Science
University of Toronto

CSC207---Software Design, Summer 2007

Course Information

Lecturer

Wael Aboelsaadat
Office hours: Thursday 5:00-7:00pm
Office hour location: Bahen 4261
Email: wael@cs.toronto.edu

Teaching Assistants

Nilesh Bansal, Jin Chen and Andres Cavilla
Office hours: TBA
Office hour location: Bahen 2272

Information Sources

- Course web site: http://ccnet.utoronto.ca/20075/csc207h1y/

**ALL ANNOUNCEMENTS WILL BE MADE THROUGH THE COURSE WEB SITE AND IT IS
YOUR RESPONSIBILITY TO VISIT IT FREQUENTLY.**

Lectures, Tutorials and Office hours

- Lecture: Tuesday 2:10-4:00pm @ Bahen 1210
- Tutorial: Tuesday 5:10-6:00pm @ Bahen 1210
- Tutorials begin on the first week of the term.

Schedule

Date	Lecture	Tutorial	Assignment
Week of May 14	intro, cvs, maps	unix, cvs	E1 out May 17th
Week of May 21	shell, testing	shell, discuss E1	
Week of May 28	python	eclipse, testing	E1 due May 31st 8:00pm E2 out May 31st
Week of Jun 4	python, reg exp	python, discuss E2	
Week of Jun 11	python tools, make	reg exp examples and sample midterm questions	E2 due Jun 14th 8:00pm
Week of Jun 18	debug, graphs	**midterm 1**	E3 out Jun 21st
Week of Jun 25	graphs, xml, dom	discuss E3, python tools	
Week of Jul 2	reflection, filters	make, xml	E3 due Jul 5th 8:00pm E4 out Jul 5th
Week of Jul 9	config, patterns	discuss E4, reflection, filters	
Week of Jul 16	patterns	**midterm 2**	E4 due Jul 19th 8:00pm E5 out Jul 19th
Week of Jul 23	patterns	patterns	
Week of Jul 30	refactoring, obj models	discuss E5, refactoring, obj models	
Week of Aug 6	review	review	E5 due Aug 10th 8:00pm

Prerequisites

- If you lack a course-prerequisite/CGPA-requirement, the CS undergraduate office will eventually remove you from the course. Only in special cases will I give my permission for a student to take CSC207 without the course prerequisites. See me as soon as possible to discuss this.

Course Grading Scheme

Item	Weight
Exercise 1	10%
Exercise 2	10%
Exercise 3	10%
Exercise 4	10%
Exercise 5	10%
Midterm 1	7% (50 minutes, Jun 19)
Midterm 2	8% (50 minutes, Jul 17)
Exam	35%

- All assignments are to be done individually.
- Students receiving $\leq 60\%$ on an exercise **must make an appointment** to see me.
- You must receive at least 40% on the final exam in order to pass this course.

Late Policy and Remarks

- Exercises may be handed in up to 4 hours late: a 25% deduction is applied. Anything submitted more than 4 hours late will not be marked. Exceptions will be made only for natural disasters, illness (accompanied by a doctor's note), and serious personal emergencies (accompanied by documentation).
- There will be a minimum of 10% deduction for having to fix incorrect submissions such as incorrect filenames — capitalization counts. **Java programs that do not compile and Python programs with syntax errors will receive a grade of 0**. They will be remarked only if you provide a remarking request that clearly states how to fix the compilation or syntax errors.
- All decisions related to late assignments/remarking must come through instructor.
- Remark request form is on the course website.

Policy on Collaboration

Plagiarism -- or simply, cheating -- is taken to be the handing in of work not substantially the student's own. It is usually done without reference, but is unacceptable even in the guise of acknowledged copying. It is reprehensible, and the penalty ranges from a zero on the assignment to suspension from the university.

It is not cheating, however, to discuss ideas and approaches to a problem, nor is it cheating to seek or accept help with a program or with writing a paper. Indeed, a moderate form of collaboration is encouraged as a useful part of any educational process. Nevertheless, good judgment must be used, and students are expected to present the results of their own thinking and writing. Never copy another student's work -- it is plagiarism to do so, even if the other student "explains it to you first." Never give your written work to others. Sharing work with others for the purposes of plagiarism is also a violation. Do not work together to form a collective solution, from which the members of the group copy out the final solution. Rather, walk away and recreate your own solution later. The basic premise is that you should do your own thinking, your own design, and your own coding.