CSC270F - Fundamental Data Structures and Techniques (Day) **Fall 1999**

Instructor: Daniel Panario, daniel@cdf.toronto.edu

Home page for the course:

`http://www.cs.toronto.edu/~daniel/teaching/270day/index.html`

Lectures: Mondays and Wednesdays from 11:10 to 12:00 in LM 162.

Tutorials: Fridays from 11:10 to 12:00; tutors and places are:

Tutor name	Room	Student name
Alfredo Gabaldon	(SS 2135)	A-C
Babak Farzad	(SS 2106)	D-I
Carl Kumaradas	(SS 1074)	J-L
Thomas El-Maraghi	(SS 1072)	M-S
Luke Stras	(SS 2111)	T-Z

Office hours: Mondays from 5:10 to 6:00 PM in SF3207, and Wednesdays from 12:10 to 1:00 PM in SF2302D. The tutors marking an assignment will have office hours on Fridays before the assignment is due (Fridays Oct. 8, Oct. 22, Nov 19, Dec. 3), from 4:10 to 6:00 PM in SF3207.

Textbook:

No textbook. We will use the "CSC270S Fall 1999 Readings" notes in the bookstore.

C programming language book:

K.N. King, "*C programming: a modern approach*", Addison-Wesley, 1996.

Other texts for consult:

B.W. Kernighan and D.M. Ritchie, "*The C programming language*", 2nd ed., Prentice-Hall, 1988.

B. Stroustrup, "*The C++ programming language*", 3rd ed., Addison-Wesley, 1998.

Course goals (from the academic calendar): Standard programming methods, with an introduction to C and C++. Use of classes to represent abstract data types, graph representation and graph algorithms. Simulation: Data structures and program organization for event-driven models. Representation of floating-point numbers; introduction to numerical methods. Optimization using dynamic programming. Programming assignments stress both the proper use of abstract data types (lists, stacks, trees, heaps) and approaches to writing larger, more complex programs. The primary purpose of the course is to teach data structures and programming techniques, not to introduce the languages C and C++.

Prerequisite: CSC148H. Co-requisite: MAT 132Y/133Y/135Y/137Y/138Y/157Y/A26Y.

Course Outline (tentative):

1. Numerical Methods & C Programing Language [4 weeks]. C control constructs, data types, I/O, and functions. Finding roots and maxima/minima. Integration methods.
2. Graphs & C++ Programing Language [3 weeks]. C pointers. Object oriented programming. C++ classes & templates. C++ I/O. Graph representations. Breadth-first search. Depth-first search. Shortest path and transitive closure. Maybe topological sort.
3. Simulation & C++ [3 weeks]. Data structures for simulation. Event–driven models. Queueing.
4. Optimization [2 weeks]. Greedy algorithms. Dynamic programming.
5. Number Representation [1 week]. Floating point numbers and their perils.

Marking Scheme:

4 Assignments at 10%, 10%, 12%, 8%	40%
1 Term Test at 15%	15%
Final Exam at 45%	45%
	———
	100%

To pass the course, you must receive at least 35% on the final exam.

Schedule:

First lecture - September 13

Assignment	**Hand-out Date**	**Due Date**
1	September 27	October 12
2	October 12	October 25
3	November 1	November 22
4	November 22	December 6

Midterm test - Friday, October 29 (in tutorial)

Last date to drop - Friday, November 5

Last lecture - Wednesday, December 8

Final exam - December 13 - 21, 1999.

Additional material will be taught in the tutorials. You are expected to know this material. Graded assignments and midterm test will be handed back in tutorial.

See the webpage of the course for the policies on plagiarism and lateness.