**CSCB70S - Fundamental Data Structures and Techniques** **Spring 1998**

**Instructor**: Daniel Panario, panario@scar.utoronto.ca

**Tutor**: Nick Cheng, nick@scar.utoronto.ca

**Lectures**: Mon, Wed 2:10–3:00 in S309

**Tutorials**: there are 4 tutorial sections

| | | |
|---|---|---|
| Wed | 11:10–12:00 | (R4031) |
| Wed | 12:10–13:00 | (R4224) |
| Fri | 10:10–11:00 | (S358) |
| Fri | 11:10–12:00 | (H402) |

**Office hours**:

Lecturer: Mon, Wed 3:10–4:00 in S-626A.
Tutor: Mon, Thu 10:00–11:00 in S627B.

**Textbook**:

No textbook. We will use the CSC270 Course 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*", 2nd ed., Addison-Wesley, 1991.

**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. Throughout the course, programming assignments stress both the proper use of abstract data types (lists, stacks, trees, heaps) and approaches to writing larger, more complex programs. Prerequisite: CSCA58F/S. Pre or co-requisite: MATA26Y & one A-level science course.

**Course Outline (tentative)**:

1. Numerical Methods & C Programing Language [5 weeks]. C control constructs, data types, I/O, and functions. Floating point numbers and their perils. 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. Topological sort. Maybe shortest path and transitive closure.
3. Simulation & C++ [3 weeks]. Data structures for simulation. Event–driven models. Queueing.
4. Optimization [2 weeks]. Greedy algorithms. Dynamic programming.

**Home page for the course:**

`http://www.cs.toronto.edu/~daniel/teaching/B70/index.html`

**Marking Scheme:**

| | |
|---|---|
| 4 Assignments at 10%, 12%, 18%, 10% | 50% |
| 1 Term Test at 15% | 15% |
| Final Exam at 35% | 35% |
| | 100% |

To pass the course, you must receive at least 35/100 on either the final exam or the term test. See the webpage of the course for the policies on plagiarism and lateness.

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

**Schedule:**

**First lecture** - Jan 5

| Assignment | Hand-out Date | Due Date | Contents | Worth |
|---|---|---|---|---|
| 1 | Jan 12 | Jan 30 | Numerical analysis | 10% |
| 2 | Feb 9 | Feb 27 | Graphs | 12% |
| 3 | Mar 2 | Mar 20 | Simulation | 18% |
| 4 | Mar 23 | Apr 3 | Dynamic programming | 10% |

**Midterm test** - Feb 11 (2:10 in lecture)

**Reading week** - Feb 16 - 20

**Last date to drop** - Mar 6

**Last lecture** - Apr 8

**Final exam** - Apr 20 - May 6, 1998.