Welcome to CS 250B Winter 2004

Introduction to Computer Science

Instructor : Claude Crépeau

(text used by permission of Prof. Prakash Panangaden)

First of all let me welcome you to the computer science program at McGill. We all hope that you will find the courses educational, challenging and entertaining. This, the first course, will introduce you to (some of) the basic concepts of computer science. The most fundamental idea is that of an *algorithm*. This is an ancient concept, at least 2500 years old, and was articulated by Arab, Aztec, Chinese, Greek and Indian mathematicians a long time before physical computers were imagined. It is remarkable that this central concept has remained unchanged over the years.

In this class we will learn to design algorithms that are *correct* and *efficient*. We will learn a scientific approach to discussing the efficiency of algorithms, and rigourous techniques for reasoning about algorithms. We will also learn how to implement algorithms (this is what is normally called "programming").

You will have noticed that programming is only one of the topics that I have mentioned. Programming is a vital skill, and becoming good at it is a major component of the course. However, the design and study of algorithms is the core of computer science. Please do not confuse knowledge of one or another system or machine as being equivalent to knowing "computer science". It is a sad fact that many colleges and educational institutions offer courses on "using computers" and call it computer science.

Systems and machines change but algorithms are of general and permanent utility. This course is **not** about how to use Windows98 or a UNIX system, nor is it even about learning a specific language. It is about algorithms. At the end of this course you will be able to design and implement a variety of algorithms. This knowledge will be *portable* across languages and machines.

Having said all this, learning about algorithms in a purely abstract way is not very useful either. Thus there will indeed be programming assignments. I assume that you know the elements of programming in some language such as Pascal or Basic. We have a laboratory of about 20 Pentiums which will run Java. Several other labs on campus are also available. We will use the language called "Java" in this class. This is a relatively new but very popular object-oriented programming language.

Programming Assignments

There will be 5 programming assignments. You will be expected to develop and **test** and **exhibit** working programs. We expect that you will comment all programs thoroughly. We want you to hand in a diskette containing the program, a printout of the program listing and a brief discussion of the test cases that you checked. Handing in a program listing with no indication that you actually ran it is useless to us. Most of you paid no attention to that sentence, let me reiterate it. **If you just hand in the program listing you will get ZERO!!**. We, after all, cannot tell whether a program runs by staring at it. If we could, we probably would not need computers!!

References

A reasonable reference for this course is:

> Data Structures and Algorithms in Java (Third Edition) by Michael T. Goodrich, Roberto Tamassia, publisher: John Wiley and Sons; ISBN 0-471-46983-1

I will not follow this book but it gives some useful information on most topics in the course. To do well in the course you must know the material that I covered in my lectures. However, you **must** get "A Little Book on Java", by F. Ahmad and P. Panangaden, available in PDF from the course web page. In addition a book called "The Java Programming Language, Second Edition" by Arnold and Gosling is a recommended reference. This is the classic reference on Java by the people who invented the language. Another book that you might consider is "The Java Tutorial" by Campione and Walrath. This is the "official" tutorial and contains lots of details. Another reasonable book is "Java, How to Program" by Deitel and Deitel. This book is thick and contains lots of examples. **Please keep in mind that the emphasis on the course is the design of algorithms and data structures rather than the Java language**. I have ordered only the first of these books through the bookstore but the others are widely available. I am only recommending these books tepidly, you need not get any of them. **Last year several students said that they never looked at anything except my lecture notes.**

Grading

The homework accounts for 30% of the total course marks. I will not give extensions unless you have a valid medical reason. Being busy with other classes is not a valid medical reason. The midterm examination accounts for 10% and the final examination will be 60% of the total. The midterm examination will be in class, the final examination will be in the final examination period. All my examinations are open-book and open-notes, but don't peek at your neighbour's paper! I will give grades based on the distribution of marks and not on any standard formula for converting raw marks to letter grades. However I will stay close to the guidelines and will not give you letter grades lower than what you would expect according to the guidelines.

Staying in touch

I will hold office hours once a week, Thursday from 13:30 to 16:00. Please do come and see me during those hours. Do not feel that you are infringing on my time - my office hours represent time that I have reserved specifically to see you.

Outside my office hours finding me is a hit or miss proposition. I recommend using email to ask me questions. In addition we will have an electronic bulletin board devoted to this class so I can make general announcements, clarifications of homework and engage in discussions. I will have one TA run this over WebCT. This may take a few weeks to set up as I have no experience with this tool. **There is a class web page**. Please ensure that you have access to it. The URL is `http://crypto.CS.McGill.CA/~crepeau/CS250/`

NOTE: 308-203 and 308-250 are considered to be equivalent from a prerequisite point of view, and may not both be taken for credit. Computer Science Major and Honours students are strongly advised to take 308-250 instead of 308-203. They are also advised to take 189-240 simultaneously with 308-250 (or with 308-202 or -203, and before 308-251).

Course Schedule for CS 250B Winter 2004

Week 1	Lecture 1	Jan 5	Introduction, Algorithms	
	Lecture 2	Jan 7	Algorithms	
	Lecture 3	Jan 9	Basics of Java	
Week 2	Lecture 4	Jan 12	Arrays and Iteration in Java	HW1 out
	Lecture 5	Jan 14	Procedural Abstraction	
	Lecture 6	Jan 16	Classes and Methods	
Week 3	Lecture 7	Jan 19	Thinking Recursively	
	Lecture 8	Jan 21	Recursive Methods in Java	
	Lecture 9	Jan 23	Induction	
Week 4	Lecture 10	Jan 26	Recursion and Induction	HW1 due, HW2 out
	Lecture 11	Jan 28	Running Time and Big O	
	Lecture 12	Jan 30	Big O, Omega, Theta	
Week 5	Lecture 13	Feb 2	Running Time	
	Lecture 14	Feb 4	Running Time and Recursion	
	Lecture 15	Feb 6	Running Time and Recursion	
Week 6	Lecture 16	Feb 9	Lists	HW2 due
	Lecture 17	Feb 11	Stacks	HW3 out
	Lecture 18	Feb 13	Polynomials	
Week 7	Lecture 19	Feb 16	Sorting Lists	
	Lecture 20	Feb 18	Classes and Objects	
	Lecture 21	Feb 20	**MIDTERM**	
Week 8	Lecture 22	Mar 1	Classes and Objects	HW3 due, HW4 out
	Lecture 23	Mar 3	Lists in Java	
	Lecture 24	Mar 5	Queues	
Week 9	Lecture 25	Mar 8	Graphs and Trees	
	Lecture 26	Mar 10	Trees	
	Lecture 27	Mar 12	Binary Sarch trees	
Week 10	Lecture 28	Mar 15	DFS and BFS	
	Lecture 29	Mar 17	Heaps and Heapsort	
	Lecture 30	Mar 19	Computational Geometry	
Week 11	Lecture 31	Mar 22	Simple closed path	HW4 due, HW5 out
	Lecture 32	Mar 24	Convex Hull	
	Lecture 33	Mar 26	Pattern Matching	
Week 12	Lecture 34	Mar 29	Regular expressions and FSA	
	Lecture 35	Mar 31	FSA simulation	
	Lecture 36	Apr 2	Guest lecture	
Week 13	Lecture 37	Apr 5	Computability Theory	
	Lecture 38	Apr 7	Complexity Theory	
Week 14	Lecture 39	Apr 13	Review of the course	HW5 due