CSC 191: Computer Algorithms, Data Structures and Languages; Winter 2002

Homepage: `http://www.dgp.utoronto.ca/~mjmcguff/courses/csc191/02s/`
Students are required to check the homepage regularly for announcements.

Instructor: Michael McGuffin (mjmcguff@cs.utoronto.ca)
Lectures: Mondays at 4 pm, Tuesdays at 4 pm, and Thursdays at 1 pm, in GB220. Final lecture April 11.
Instructor's office hour: Thursdays 2-3 pm in SF2110.

Lab time: Fridays 3-6 pm in SF1106 / SF1012. Students are **strongly encouraged** to attend supervised lab time so they can obtain help from a teaching assistant if required.

Newsgroup: `ut.ecf.csc191`

Textbooks:
Required: Shaffer. *A Practical Introduction to Data Structures and Algorithm Analysis*, 2nd ed., 2001.
Recommended: Lippman and Lajoie. *C++ Primer*, 3rd ed., 1998.
Optional: Cormen, Leiserson, Rivest. *Introduction to Algorithms*, 2nd ed.

Marks breakdown: Assignments - 20%, Midterm - 30%, Final - 50%.
There will be 5 to 6 assignments given over the course of the term. All assignments will involve an implementation portion (in C++) to be submitted electronically; some assignments may also involve a written portion (involving mathematical exercises) to be submitted at the beginning of class. The implementation portion of Assignment 1 will be due Friday, January 25, just before midnight (23:59:59). Late assignments will not be accepted. Due dates for subsequent assignments will be announced during the term.

Good programming style is expected and required, and will be marked as part of each assignment.

Discussion among students of problem solving strategies is encouraged, however, any work submitted for credit must be the student's own work. A good rule of thumb is: nobody may leave a discussion with written notes of what was said, and furthermore, each person should watch 1 hour of television after a discussion before working individually on assigned work. Source code will be checked for plagiarism by an automated system capable of detecting similar programs even if considerable effort has been taken to conceal the plagiarism. **Cases of plagiarism will be treated as offenses** under the Code of Academic Discipline of the University, and will be dealt with accordingly. Penalties range from grade penalties to suspension. It is an offence not only to copy work, but also to knowingly allow your work to be copied.

Topics to be covered include, but are not limited to:

- A review of C++. Classes, private and public membership, inheritance, virtual functions, overloading, overriding, polymorphism, templates. OOP philosophy. Habits and techniques for writing better code.
- Recursion and problem solving using divide-and-conquer.
- Analysis of algorithms and asymptotic ("big-oh") notation to express time/space complexity.

- Linked lists, arrays, multi-dimensional arrays, stacks, queues, deques, sets.
- Sorting algorithms. Insertion sort, shellsort, quick sort, merge sort, et al.
- Binary trees and non-binary trees. Implementation, traversal, searching, balancing, and applications of trees. AVL trees.
- Heaps and heapsort.
- Hashing and hash tables.
- Game trees. The minmax algorithm and alpha-beta pruning.
- Graphs. Minimal spanning trees. Shortest paths. Topological sort.
- Grammars. Left-most and right-most derivation of sentences. Syntax trees. Recursive-descent parsing.
- (Time permitting) Introduction to complexity theory. Deterministic and non-deterministic Turing machines. The halting problem. NP and NP-completeness.