# Course Outline
# Introduction to Computer Science COMP 250

(Winter 2008 MWF 1:35-2:25 TROTTIER 100 )

| | |
|---|---|
| Instructor: | Professor Michael Langer |
| Office: | McConnell Engineering 329 |
| Tel: | 398-3740 |
| Email: | langer@cim.mcgill.ca |
| Course Web Page: | `http://www.cim.mcgill.ca/~langer/250.html` |
| Office Hours: | by appointment |

Teaching Assistants (T.A.): See course web page for this information.

## Introduction

This course is an introduction to several core topics in computer science. The main emphasis will be on data structures and algorithms, viewed from an object oriented perspective. We will see how many common data structures such as lists, stacks, heaps, trees and graphs and many common algorithms such as searching through and sorting a set of data can be defined and implemented in an object oriented language, namely Java. We will also learn about the complexity of algorithms, for example, why certain algorithms take less time to run or use less computer memory than other algorithms.

In subsequent data structures and algorithms courses, namely those that follow the COMP 251 stream, you most likely will not take an object oriented approach. Indeed you will tend to understand and "solve" problems with pencil and paper only. So why are we taking an object oriented approach here in COMP 250? There are several reasons. First, we would like to get you thinking about (and doing) object oriented programming (OOP) as early as possible in your education as a computer scientist. The object oriented paradigm has become mainstream in computer science and the sooner you get familiar and comfortable the better off you will be. This will not only be useful for you in subsequent software themed courses, namely those that follow the COMP 302 stream, but it will also be useful for those of you who will be involved in large software projects in industry or in research where you will most likely be programming in an OO language (either Java or C++). Indeed even if you only write code for yourself, the object oriented approach will help you to organize your code and share code with others.

The second reason for taking an object oriented approach to data structures and algorithms is that it gives you "hands on" experience which pencil and paper cannot give. Later, in the COMP 251 stream, when you are learning more advanced techniques of data structures and algorithms and do not wish to be slowed down by having to read or write real code, you will at least have *some* experience in programming the basics. This will help ground you and put you in a position where you *could* implement the techniques without too much effort to solve real problems.

## Prerequisites

The prerequisite for this course is `COMP 202 Introduction to Computing 1` or any one semester course on programming in a high level language such as C or Java. Typically the vast majority of students in COMP 250 have taken COMP 202, and so are familiar with the basics of Java and object oriented programming.

If you have not taken COMP 202, you should see the COMP 202 web page for the Fall 2008 semester to make sure you have the appropriate background for COMP 250. In particular, at an absolute minimum, you should have some experience with programming in a high level language and you should have written simple programs that use:

- variables, expressions
- conditional statements (if-then-else, switch)
- loops (while, for)
- arrays
- pointers (references)
- I/O (keyboard inputs or reading from a file, writing to console or to a file)

COMP 202 covers Java and hence students who have taken 202 will also be familiary with basic concepts in object oriented programming (OOP).

**IMPORTANT:** Those of you who have only programmed in C (but not C++) will not be familiar with the basic concepts of classes and objects are how they are defined in Java. *You will need to do some work in the first week or two of the semester to catch up*, in particular, you should do *both* of the following:

- Read Chapter 1 from the textbook for COMP 250, namely **Data Structures and Algorithms in Java** (Fourth Edition) by Michael T. Goodrich and Roberto Tamassia. This will give you a basic introduction to classes, objects, methods, etc.
- Read through the lecture slides from COMP 202 in the Fall 2008 semester. See `http://www.cs.mcgill.ca/~cs202/2008-09`.

## Topics

Here is a rough breakdown of the topics. For details on the lecture schedule, see `http://www.cim.mcgill.ca/~langer/250.html`.

- Object oriented concepts in Java (1 week)
  - inheritance, polymorphism, interfaces, types and casting
- Linear data structures (3 weeks)
  - arrays, lists, stacks, queues
- Non-linear data structures (4 weeks)
  - trees, heaps, binary search trees, graphs
- Algorithms (3 weeks)
  - searching and sorting
  - recursion
  - analysis (big O)

## Evaluation

Your final grade will be calculated using the following percentage breakdown.

- **Four Assignments (40 % total)**
  - A1 to be posted in mid-late January
  - A2 to be posted in early February
  - A3 to be posted in early March
  - A4 to be posted in late March

  You will be given at least 10 days to complete each assignment.

- **Three in-class Quizzes (30 % = 3 × 10 %)**
  - Quiz 1 in early February
  - Quiz 2 in early March
  - Quiz 3 in late March

- **Final Exam (30 %)** held during Final Examination Period.

If your percentage grade on the Final Exam is greater than your percentage grade on the Quizzes, then your 30 % Final Exam grade will replace your 30% Quiz grade and so your final exam will count for 60% of your final grade.

## Textbook and Lecture Notes

The required textbook for the course is **Data Structures and Algorithms in Java** (Fourth Edition) by Michael T. Goodrich and Roberto Tamassia. Published by John Wiley & Sons, Inc. ISBN: 0-471-73884-0. It is available for purchase in the McGill Bookstore, and also at the Schulich Library for 3 hour loan.

Lecture notes will be made available on the course web page, as will links to other material.

## McGill policy on academic integrity

*McGill University values academic integrity. Therefore, all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures. See `http://www.mcgill.ca/integrity` for more information, as well as the Student Guide to Avoid Plagiarism, `http://www.mcgill.ca/integrity/studentguide/`.*