CMPUT-201 Course Outline (Winter 2006)

Instructor:	Dr. Michael Buro
Office:	Ath 3-37
Phone:	492-1763
Email:	`mburo@cs.ualberta.ca`
Section Web Page:	`http://www.cs.ualberta.ca/~mburo/courses/201`
Course News Group:	`ualberta.courses.cmput.201`
Office Hours:	Wednesdays 14:00-14:30 (or by appointment)

Course Objectives

The main theme of CMPUT-201 is small–scale programming to well defined interfaces. In this course we use components as black boxes that have an interface, we build components that are supposed to implement a given interface, and use tools like compilers, debuggers, and profilers to do this effectively. In terms of programming languages, this course starts with teaching C and then will move to C++ for more advanced programming concepts. In terms of problem domains, CMPUT-201 focuses on the manipulation and transformation of basic data types such as numbers, arrays, and strings. In terms of techniques, the course focuses on low–level language issues such as data representation, memory management, pointers, call stack, and exception handling.

The CMPUT-201 homepage at `http://www.cs.ualberta.ca/~mburo/courses/201` is the official course repository. Please visit this page regularly to obtain course information of general interest such as news, links to tutorials, manuals, reference cards, and lecture notes. General course-related questions should be posted to the newsgroup which will be monitored by the TAs and me.

Textbooks

There are many good (and many more bad) books on C, C++, and Unix. For this course we have chosen the following books:

- Horstmann & Budd: Big C++ (good and comprehensive C++ introduction, required)
- Sobell: A Practical Guide to Linux Commands, Editors, and Shell Programming (good Linux introduction, required)
- King: C Programming: A Modern Approach, First Edition, (good C source, recommended)
- Josuttis: The C++ Standard Library: A Tutorial and Reference (good C++ library description, recommended)
- Meyers: Effective C++, More Effective C++, Effective STL (advanced, very good!)

Course Work and Evaluation

This course has weekly labs and assignments, one midterm exam, and a final exam. The weights are as follows:

Component	Weight
11 assignments (3% each)	33%
midterm exam	21%
lab exam	10%
final exam	36%

N.B.: Apply for a Unix id in CSC 1-43 **before** the first labs start in the second week!

Assignments are handed out Tuesdays in class. Solutions are due at the begin of the following Tuesday lecture at 12:30pm in class (or submitted electronically using *astep*). A 15% late penalty is assessed for each day (or a fraction of a day) the assignment solution is late, including weekends and holidays. If you can't make it to class, slide your solution under my door **before** it is due. Please notify the TAs by email on the due day in case you will be submitting late. Make sure that your submitted software runs on the ugrad lab machines (`ug01`...`ug34`) in CSC 159.

The in–class midterm exam is only given at the preset date. If you cannot attend the exam due to illness or severe domestic affliction you may request an excused absence by contacting me by email within 48 hours of the missed midterm exam. If an excused absence is granted, the weight of the exam will be shifted to the final exam. The final exam cannot be rewritten (even if it is worth more than 40% in the case some of your term work was shifted to the final exam). However, a student missing the final exam can apply for a deferred exam through the Faculty Office in which the student is registered. If approved, the student can attend the deferred final exam scheduled on **June 1, 2006** at 15:00–18:00 in room CSC 249. The alternate date for the lab exam in case of an excused absence is **April 18, 2006** at 14:00–16:50 in CSC 159.

Final grades will be based on the 4–point grading system and assigned in accordance with the University of Alberta grade distribution guidelines for second year courses as specified in the University of Alberta Marking and Grading Guidelines. I have the discretion in setting the borderline between passing and failing and, in doing so, may consider a student's entire performance across the term as well as the overall percentage.

Collaboration, Plagiarism, and Cheating

Make sure you are familiar with the Code of Student Behaviour in the University of Alberta Calendar (online at `http://www.ualberta.ca/~unisecr/policy/sec30.html`). Plagiarism and other forms of cheating are considered to be serious academic offences.

Discussing assignments among students is allowed. You must, however, submit individual solutions. All sources including books, webpages, and fellow students who took part in assignment discussions, need to be stated. Failure to do so constitutes plagiarism. You also must be able to explain your solutions in detail individually. We reserve the right to conduct interviews in suspicious cases.

Plagiarism or cheating will be reported to the Dean. The Dean will then decide on further action which can result in you being assigned a grade of F for the course and an extended suspension from university. Note that we will use software tools to detect similarities among submitted solutions.

Tentative Schedule

Duration (weeks)	Topic
1.5	Unix Operating System Introduction
5.5	Procedural Programming (C,C++)
3	Object Oriented Programming (C++)
2.5	Generic Programming (C++)

Important dates:

- Feb.16: Midterm exam (closed book, 50 minutes)
- Mar.20: Lab exam
- Apr.4: Course evaluation
- Apr.11: Last class

Please visit the section webpage for up–to–date information.