

CMPUT-201 Course Outline (Winter 2005)

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: Tuesdays 15:00–15:30 (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 teaches C and C++, although these will not be treated as distinct languages. Rather, we will be using C++ from the beginning and view C as a subset. In terms of problem domains, CMPUT201 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, exception handling etc. The CMPUT201 homepage at <http://ugweb.cs.ualberta.ca/~c201> is the official course repository. Please visit this page regularly to obtain course information of general interest such as news and links to tutorials, manuals, and reference cards. Information on this section (schedule, lecture notes) can be found at <http://www.cs.ualberta.ca/~mburo/courses/201>.

Textbooks

There are many good (and many more bad) books on C, C++, and Unix. I am recommending – **but not requiring** – the following books for this course:

- Horstmann & Budd: Big C++ (good and comprehensive introduction)
- Josuttis: The C++ Standard Library: A Tutorial and Reference.
- Meyers: Effective C++, More Effective C++, Effective STL (advanced, very good!)
- Robbins: Unix in a Nutshell (Unix command reference)

Course Work and Evaluation

This course has weekly labs, 4 programming assignments, 5 homework quizzes, a midterm exam, and a final exam. The weights are as follows:

Component	Weight
4 programming assignments (5% each)	20%
5 homework quizzes (2% each)	10%
11 labs (1% each)	11%
midterm exam	20%
final exam	39%

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

Assignments are due by 7:59pm on the particular due date. A 15% late penalty is assessed for each day (or a fraction of a day) that the assignment is late, including weekends and holidays. There are weekly labs where

students do exercises that need to be handed in by the end of the lab. You can hand in lab solutions up to 48 hours late, but only at most three times. Additional lates get 0 marks. Homework quizzes are released every two weeks Thursdays in class. Solutions are collected in class two weeks later. If you can't make it to class, slide your solution under my door **before** the due date. Late quiz solutions get 0 marks. The in-class midterm exam is only given on the preset date. If a student cannot attend this exam due to illness or severe domestic affliction the student 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, 2005** at 15:00–18:00 in room CSC 2-49.

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.

Plagiarism and Cheating

Make sure you have read and 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.

All work is to be done individually. Suspected cases of plagiarism or cheating will be reported to the Dean. The Dean will decide on further action which can result in 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. We even take submissions from previous terms into account.

Tentative Schedule

Visit the section web page for up-to-date information.

1	Jan.10	C++ Intro, Linux/shell/emacs/g++/make	
2	Jan.17	Simple types, declarations, operators, bit operations	L1
3	Jan.24	Operators cont., control flow, functions, standard i/o, variable scope	L2
4	Jan.31	Scope continued, arrays, function parameters	L3
5	Feb.7	Programming with arrays, C-structs, pointers	L4
6	Feb.14	C-strings, pointer arrays, memory allocation, makefiles, typedef, preprocessor	L5
7	Feb.21	Reading Week	no labs
8	Feb.28	Multi-dimensional arrays with pointers, function pointers, bitfields, unions, classes	L6
9	Mar.7	Assignment operator, inheritance, polymorphism	L7
10	Mar.14	Virtual functions, operators, static members	L8
11	Mar.21	Template functions, class templates, specialization	Q/A lab
12	Mar.28	Type traits, STL overview, vector, list, set	L9
13	Apr.4	Map, iterators, STL algorithms, sorting	L10
14	Apr.11	Exceptions, auto pointers, C/C++ file I/O, Review	L11

- Jan.14: Assignment 1 out
- Feb.4: Assignment 1 due, Assignment 2 out
- Mar.1: Midterm exam (closed book)
- Mar.4: Assignment 2 due, Assignment 3 out
- Mar.25: Assignment 3 due, Assignment 4 out
- Apr.5: Course evaluation
- Apr.12: Assignment 4 due
- Apr.14: Last 201 class