CMPUT 201
Practical Programming Methodology

Michael Buro

mburo@cs.ualberta.ca
http://www.cs.ualberta.ca/~mburo

Lecture 1

- Course Overview
- Administrativa
- UNIX, C, C++ Intro
- Linux/shell/emacs/g++ Intro

My Background

- Ph.D. in CS (1994):
  Machine Learning applied to Heuristic Search
- 1995-2002: NEC Research Institute, Princeton
- Since July 2002: **A.I. Group** at U. of A.
- Research Interests:
  - Artificial Intelligence, Machine Learning
  - Heuristic Search
  - **A.I. for Real-time strategy games** - ORTS
- Wrote **Logistello** which defeated the reigning Othello World-champion **6-0** in 1997

Contact Information

- E-mail: mburo@cs.ualberta.ca
- WWW: www.cs.ualberta.ca/~mburo
- Phone: 492-1763
- Office: Athabasca Hall 3-37
- Office Hours: Tuesdays 3-3:30pm

Important!
If nobody shows up by 3:15pm,
I am free to go

Software Engineering Courses

- **201: Small scale programming**
- 301: Team work, object-oriented design
- 401: Large scale programming

CMPUT 201 Syllabus

- **C++ language**
  - types, variables, expressions, statements, loops, functions, recursion, exceptions
- **Programming Paradigms**
  - object oriented programming
    - classes, inheritance, interfaces
  - generic programming (templates)
- **C++ standard libraries**
  - libc++, STL (standard template library)

- **Low-level aspects:**
  - memory allocation, pointers
  - stack and class layout
  - bit operations
  - C I/O
- **UNIX commands and programming tools**
  - command line interface (shell)
  - editor (emacs)
  - compiler (g++)
  - debugger (gdb)
  - acting on file dependencies (make)
  - performance profiling (gprof)

Course Work

• **4 Assignments** (5% each)	20%
• **5 Homework Quizzes** (2% each)	10%
• **11 Labs** (1% each)	11%
• **Miderm Exam**	20%
• **Final Exam**	39%

Final grades: 4-point scale, distribution method

Course Information

- Course page: http://ugweb.cs.ualberta.ca/~c201
  - Lab material, assignments
  - Related resources
- Course news group: ualberta.courses.cmput.201
  - You can ask general 201 questions here
- Section home page:
  - **www.cs.ualberta.ca/~mburo/courses/201**
  - schedule, lecture notes, and additional material
- Apply for **Unix account** in CSC-143 this week!

Suggested Reading

- Horstmann & Budd: "Big C++"
- Jossutis: "The C++ Standard Library"
- Robbins: "UNIX in a Nutshell"

Further Reading

Meyers: **Effective C++, More Effective C++, Effective STL**
"eye-opening and funny: A must if you program in C++!"

Beneficial: Maintaining your own UNIX system.
E.g. Linux, OpenBSD, Solaris ...

Policy on Collaboration and Cheating

- All course work is to be done individually
- Severe misconduct will be reported to the Dean

We use various plagiarism detection tools to compare submitted assignment solutions with those of fellow students

UNIX and C

- D. Ritchie developed C for writing and maintaining the UNIX OS in the 1970s
- C is a high-level language with many low-level features:
  - C can **manipulate memory** directly
  - High-level features make C programs **easier** to write and maintain **than assembly language**
- C syntax is "succinct" making it sometimes hard to read (e.g.: `c &= ~8;`)
- C has no automatic checks: **"do-it-yourself"**
- C **lacks** object oriented and generic constructs

Why C?

- **Lean**, compiler easy to port to other architectures
- Close to assembly language, **fast** code easy to generate
- **Direct access** to memory, CPU registers, and I/O devices
  - well suited for implementing operating systems
  - Linux (free software UNIX clone) is mostly written in C
- Easy to **call C functions** from other languages
  - many libraries are written in C

C++

- Developed by B. Stroustrup at AT&T Bell Labs in the early 1980s
- Overcomes some of C's shortcomings while staying **compatible** with C, **fast code!**
- Additions:
  - **object oriented** features: classes & inheritance
  - **generic programming** (templates)
  - **exceptions** (advanced error handling)
- Standard libraries: libc++, STL
  - Large collection of useful classes
  - Containers: vector, set, list, map, ...