CIS 399 Unix & C++

- Instructor: John Conery
  309 Deschutes
  conery@cs.uoregon.edu
- Web Site: http://cs.uoregon.edu/classes/cis399
  - outline
  - schedule
  - textbook info
  - links

Course Goals

- Practical programming experience
- Software skills for CIS classes (and beyond)
- Introduction to C++ for Java programmers
- Productivity in a Unix environment
  - shells, editors, project management, scripts, …
- Have Fun!

Prerequisites

- CIS 313 & 323
- I will assume you have experience with object-oriented programming in Java
- The C++ topics will be presented by comparison with Java
- This is not a course on C++ programming…
- I will also assume you are familiar with basic data structures (lists, trees, hash tables, …)

Course Philosophy

- Learn by doing
- Implementing an idea in software is a great way to learn
- Projects:
  - several short weekly or bi-weekly assignments
  - seven projects in all
- Individual projects -- no group projects
- A word about plagiarism….

Course Structure

- Mondays
  - Intro to topic, background information
- Wednesdays
  - Project specifications, detailed information
- Fridays
  - Discussions

Requirements

- (Bi-)Weekly assignments
  - design, code, test
  - submit code and short writeup
  - grade based on correctness, style, writeup
- Project
  - same criteria as smaller projects
- Electronic submission
  - create "tar file" containing code and writeup
  - button on project web page runs a program that submits the file
- No Exams
Grading
- Weekly projects
- Extra credit
- Participation in class discussions

Programming Environment
- All lectures, examples, projects will assume you are using the CIS Unix environment
- You may use other environments (e.g. Linux)
  - you must install software we use (tcsh, Perl, C++, …)
  - you take responsibility for downloading projects
- I strongly recommend you upload and test before submitting
- Count on using e-mail extensively
  - e.g. comments on projects will be returned by mail

Which Software to Use
- This course will cover a lot of software "tools"
  - shells, scripting languages, editors, and more
- In some cases, the tool I describe in lecture will be required
  - e.g. Perl vs. Python
- Other tools are strongly recommended
  - e.g. xemacs vs. vi, tcsh vs. ksh, …
- In other cases you can use any software
  - mail, net browser, …

Textbooks
- Kernighan and Pike, The Practice of Programming
  - Lots of practical advice on designing, coding, and testing programs
  - Many examples compare Java, C, C++
  - Several of our Friday discussions will be based on chapters in this book
- Koenig and Moo, Accelerated C++
  - Not oriented toward Java programmers, but appears to be a good, complete, concise intro to the language
- No text for Unix -- use man pages and web resources

Some Other Resources
- Budd, C++ for Java Programmers
  - It’s got the right title…
  - Expert level comparison after you’ve learned C++
- Eckel, Thinking in C++
  - Excellent, highly recommended
  - On-line!
  - See also Thinking in Java
- Myers, Effective C++
  - Available as CD (very nice)
  - Chapters on specific topics
  - Put it on your list of graduation presents…

Next…
- Getting Started With Unix