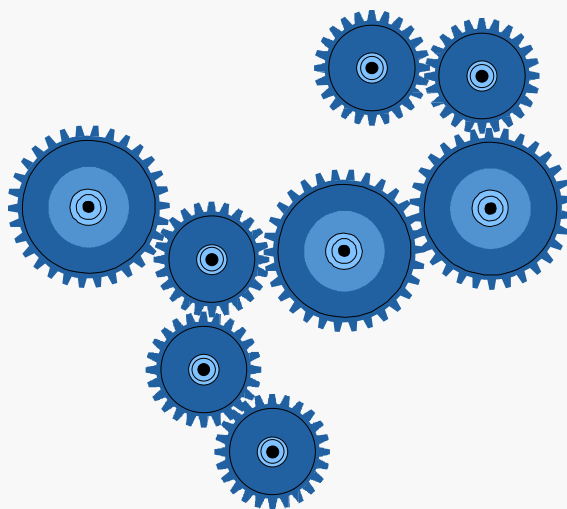


*Welcome to
Computer Science 1044
Introduction to Programming
in C/C++*



Instructor:

Instructor: William D McQuain
Email: mcquain@cs.vt.edu
Office: 631 McBryde Hall
Office Hours: 2:00—4:00 MW
and by appointment

Course Description

Credits: (2H, 2L) 3 credits

Prerequisites: None. However, computer/web literacy is assumed.
For CS majors, we also assume a minimum of one year of programming background in a suitable language. If you lack that, you should consider enrolling in CS 1114 instead.

Objectives:

The purpose of this course is to teach the fundamentals of structured programming and problem solving in the C/C++ programming language.

Required:



Programming and Problem Solving with C++, 3rd Ed., by Nell Dale, Chip Weems and Mark Headington, Prentice-Hall, ©2002

Recommended:



CS 1044 Course Notes, Fall 2002 Edition, by McQuain and Barnette, ©2002
(available for purchase at A-1 Copies in University Mall)

Other Useful Sources of Information:



Visual C++ Online Help

CS 1044 website: <http://courses.cs.vt.edu/~cs1044/fall02/mcquain/>

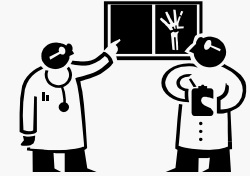
C/C++ Usenet group: alt.comp.lang.learn.c-c++

Evaluation and Grading:

Point Distribution

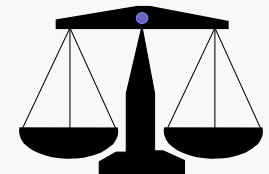
Final grades will be based on the average achieved over the following :

Item	Weight	Tentative Dates
In-lab Assignments and Homework	20%	Varied
Out-of-lab Projects and Software Engineering	40%	Varied
Tests (2)	8%, 12%	TBA
Final Exam	20%	TBA



Grade Scale:

The usual 10-point scale will apply (subject to any curve). A final average of 90% will guarantee an A-, 80% will guarantee a B-, and so forth.



Curve

A grade curve may or may not be employed in this course. The application of a curve is dependent upon class performance on tests and homework. The decision to utilize a curve rests entirely with the course instructor.

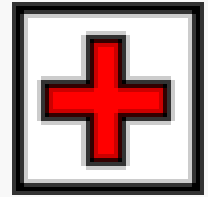
Sources for Help/Questions etc.

CS 1044 Classmates:

CS 1044 Forum for questions

CS 1044 TAs & Instructor

CS 1044 Listserv for announcements by instructor, etc.



General C++ Language Help

USENET Newsgroup: `alt.comp.lang.learn.c-c++`

A panel of "experts" will respond to questions.

We DO monitor the group.

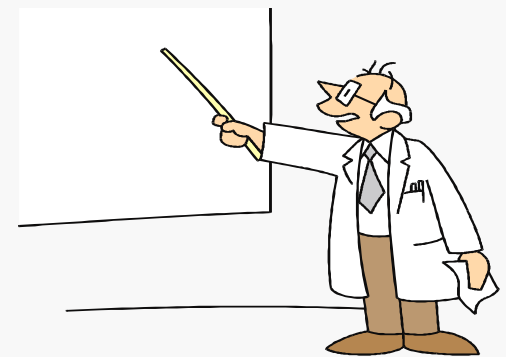


Lecture Instruction

Lectures will consist of presentations, applications, problems and solutions, ideally interspersed with classroom discussion.

Lab Instruction

Lab sessions will involve a variety of assignments designed to develop skills and illustrate lecture topics.



Test Environments

- All programming assignments submitted are required to compile under Microsoft Visual C++ .NET or the CS 1044 Sandbox, or both.
- Programs will be tested under Windows NT or 2000.
- It is the student's responsibility to ensure that his/her programs execute correctly in the appropriate environment; programs that do not will receive substantial deductions.



Program Evaluation

- Students are required to submit their source code files to the Curator system.
- Be sure to read the *Student Guide to Submitting* in the course notes pack. It describes how to prepare to submit a program to the Curator and discusses how the Curator scores your submissions.
- All submissions to the Curator are subject to the Virginia Tech Honor Code. Read the online Course Policy Statement for a detailed discussion



Backups

- **Students are responsible for making backup copies of all their work in this course.** Loss of work due to hard drive failure is **NOT** an acceptable excuse. Backup copies of files on the same hard drive are not backup copies. Backup copies of files on second hard drives are also risky. Backup copies should be maintained on two separate distinct storage mediums, (e.g., hard drives and floppies).
- Backup copies should be maintained until after the end of the term and students have received their course grade. (The Army lives by triplicate for a reason.)
- Remember: Computer systems contain mechanical devices. Systems fail. Plan for it. It is inevitable!



Deadlines

- **Assignments have deadlines.**
- **Deadlines are temperamental little beasts; hug one too tightly and it is likely to bite.**

