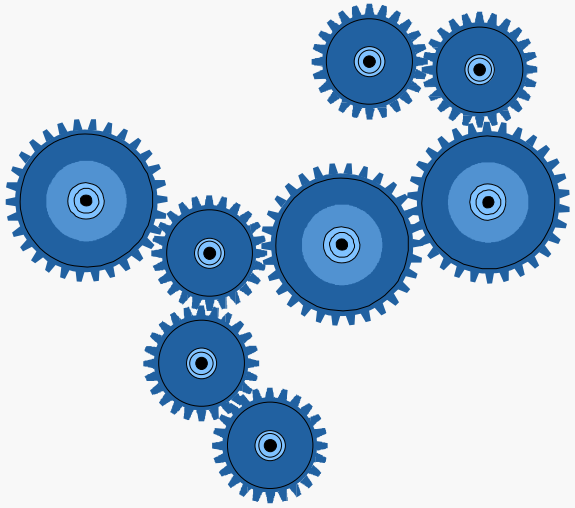
Welcome to Computer Science 2704 Object-Oriented Software Design and Construction



Instructor Information

Instructors:

Instructor:	William D McQuain
Email:	`mcquain@cs.vt.edu`
Office:	631 McBryde Hall
Office Hours:	9:00 – 10:30 MWF
	and by appointment

Instructor:	N Dwight Barnette
Email:	`barnette@cs.vt.edu`
Office:	624 McBryde Hall
Office Hours:	3:00 – 4:00 MWF, 2:00 – 3:00 TTh
	and by appointment

Course Properties

Course Description

Credits: 3

Prerequisites: CS 1704 (formerly 2574) or ECpE 2574

Prereq: CS Majors and Minors must have completed the prerequisite with a grade of C or higher (a C- is not acceptable).

Students are also expected to have attained proficiency in the procedural aspects of the C++ programming language and to have some prior exposure to the basic aspects of C++ classes.

There will be absolutely NO exceptions to these requirements.

Objectives:

Object-oriented programming concepts are studied and basic skills in software design are developed. Sound practices for design, construction, testing, and debugging of object-oriented software systems are emphasized. Object-oriented features of the C++ programming language are examined. The primary principles and language features studied are: objects, classes, inheritance, and polymorphism.

Texts & References

Required:

UML and C++: a Practical Guide to Object-Oriented Development, 2nd Ed., by Richard C Lee and William M Tepfenhart, Prentice-Hall, ©2001

C++ How to Program, 3rd Ed., by Deitel & Deitel, Prentice Hall, ©2001

Recommended:

CS 2704 Course Notes, Fall 2001 Edition, by Keller, McQuain and Barnette, ©2001 (as available)
http://courses.cs.vt.edu/~cs2704/

Other Useful References:

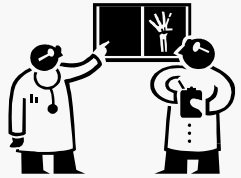
Programming and Problem Solving in C++, N. Dale, C. Weems & M. Headington, Jones and Bartlett Pub., ©2000

Data Abstraction and Problem Solving in C++: Walls and Mirrors, 2nd Ed., by Carrano, Helman and Veroff, Addison Wesley, ©1998

Object-Oriented Software Design and Construction with C++, by Dennis Kafura, Prentice Hall, © 1998

CS 1704 Course Notes, Spring 2001 Edition, N D Barnette & W D McQuain, ©2001
http://courses.cs.vt.edu/~cs1704/

Evaluation



Evaluation and Grading:

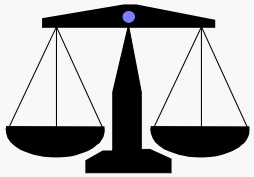
Point Distribution

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

Item	Weight	Tentative Dates
Project Testing and SE	50%	TBA
Homework and Quizzes	10%	TBA
Tests (two)	20%	TBA
Final Exam	20%	11:05 – 1:05, Wednesday December 19

Grade Scale:

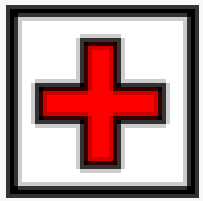
The usual 10-point scale will apply (subject to any curve). A final average of 90% will guarantee an A-, 80% will guarantee a B-, and so forth.



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.

Class Organization



Sources for Help/Questions etc.

- CS 2704 Classmates:
  - CS 2704 Listserv for announcements by instructors
  - CS 2704 website message board for discussion
- CS 2704 TAs
- CS 2704 Instructors

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.
- CS 2704 ListServ
  - Used for announcements from the course instructors

Lecture Instruction

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

Development System

Test Environments

- All programming assignments submitted are required to compile under either Microsoft Visual C++, version 6.0 or the GNU g++ compiler installed on the Linux machines in McB 124.
- Programs will be tested under either Windows NT or Linux.
- 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 Demonstrations

- For the major projects, students will demonstrate their implementation to a TA in the McBryde CS Dept. Computer Lab.
- Students may <u>not</u> bring their systems to the labs to demonstrate their programs.
- Any code changes (made at the demo) will incur a penalty equal to the late penalty at the time of the demo.

Damage Control

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 are mechanical devices. Systems fail. Plan for it. It is inevitable!