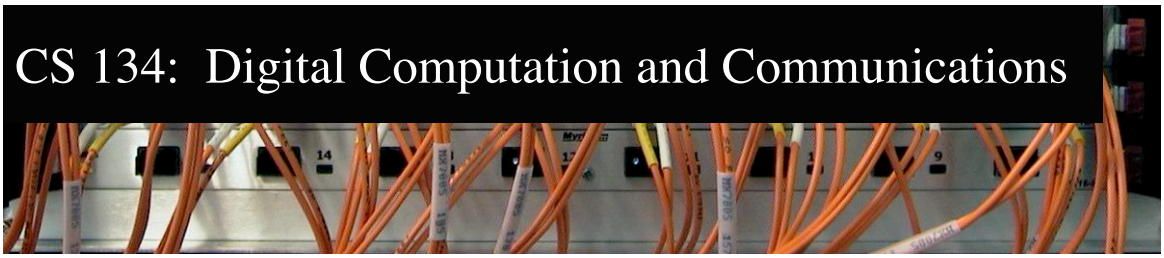
CS 134: Digital Computation and Communications



SPRING 2007

Instructor:	Brent Heeringa	Tom Murtagh
Office:	TCL 306	TCL 309
Phone:	597-4711	597-2369
Email:	heeringa@cs.williams.edu	tom@cs.williams.edu
Office Hours	Mon. & Wed: 3:30 - 5:00 Fri: 1:30 - 2:30	Tue., Wed., & Thurs.: 1:30 - 3:30

Course Description

A digital revolution has transformed the way we communicate and process information. Digital cameras have replaced film, CDs have replaced LPs, DVDs have replaced analog VCR tapes, and communications through email, chat systems, and the Web have become part of daily life. This course explores the principles that underly such digital information processing and communication systems.

The representation of information in discrete, symbolic form is ultimately what makes a system digital. We will examine digital techniques for representing information. How can a beam of light traveling through an optical fiber represent a sequence of 0s and 1s? How can numeric data capture the tones we hear when we listen to a digital recording? As we explore these questions we will examine the tradeoffs involved in the design of schemes for representing information.

All digital information processing and communication systems are driven by precise rules or algorithms expressed as computer programs. We will develop an appreciation for the nature and limitations of such algorithms by exploring abstract algorithms for complex processes and by learning the basics of computer programming in Java. Programming topics covered will include objects, classes, methods, conditional and iterative control structures, text/string processing, arrays, and recursive lists. Programming projects will include network applications like IM chat clients, tools to process and compress digital images, and simple network servers. While the programming assignments for the course will focus on the application of programming to communication, the programing skills students develop will be applicable in many other areas. In particular, this course is designed to provide the programming skills required to complete CSci 136 and/or CSci 237.

Text

Students should purchase a copy of the course packet from the Computer Science Department. Additional readings will be distributed in class or through the course web site.

Labs

Room 217A in Thompson Chemistry (TCL) is the primary lab for CSCI 134. You can do all of the programming exercises on these computers. In addition, the software we will use for this course is freely available on the web if you wish to install it on your own computer.

Lab sessions will be held each week. These sessions provide a time during which your instructor can actively assist you in the development of programs. Lab descriptions will be handed out in advance, and you will be expected to have planned your approach to the assignment before the beginning of lab.

You will submit your laboratory programs electronically. Laboratory programs will be accepted up until 11:00 P.M. on the second night after your lab session. Laboratory programs will not be accepted after

the due date, but in computing your grade the lowest score on **submitted** assignments will be dropped. Therefore, it is to your advantage to submit assignments on time, even if they are incomplete.

Test Programs and Exams

In addition to the weekly laboratory exercises, there will be three exams. Two of these are traditional in-class written exams: one given at midterm and the other during the final exam period. The times and places for these exams will be announced on the web site. The third exam is a programming assignment called a "test program" completed during the final weeks of the semester. The test program should be treated as a take-home exam. No assistance is allowed. Late test programs will be accepted, but with a penalty of 10% per day. Grades will be determined as follows:

Laboratory programs:	20%	Midterm	20%	Test Program:	20%
Homework Assignments:	20%	Final exam:	20%		

Computer Science Honor Code

The Honor Code as it applies to non-programming assignments is outlined in the Student Handbook. For programming assignments in Computer Science courses, the honor code is interpreted in very specific ways. When a program is assigned, your instructor will identify it as a "practice," "test," "laboratory," or "team" program. The Honor Code applies differently to each with respect to collaboration or assistance from anyone other than the TAs or instructors:

Test Programs. Any assignment designated as a test program is to be treated exactly as a take-home, open-book test. You are allowed to read your textbook, class notes, and any other source approved by your instructor. You may not consult anyone other than your instructor. The instructor encourages the asking of questions, but reserves the right not to answer, just as you would expect during an exam. Guideline: Any work that is not your own is considered a violation of the honor code.

Laboratory Programs. Laboratory programs are expected to be the work of the individual student, designed and coded by him or her alone. Help locating errors is allowed, but a student may only receive help in correcting errors of syntax; help in correcting errors of logic is strictly forbidden. Guideline: Assistance from anyone other than the TAs or instructors in the design or coding of program logic will be considered a violation of the honor code.

Team Programs. Team programs are laboratory or test programs to be worked on in teams of two or more students. You are allowed to discuss team programs with your partners, but work with others is otherwise restricted by the appropriate rules above. Guideline: Any work that is not the work of your team is considered a violation of the honor code.

If you do not understand how the honor code applies to a particular assignment, consult your instructor. The Department of Computer Science takes the Honor Code seriously. Violations are easy to identify and will be dealt with promptly.

Suggestion: To protect your work dispose of printouts carefully, and avoid leaving your programs on hard disks in labs and other public storage areas.

Computer Ethics

Students should be aware of the Computer Ethics outlined in the Student Handbook. Violations including uninvited access to private information and malicious tampering or theft of computer equipment or software are subject to disciplinary action.

Lecture Topics and Readings

The table below shows the topics we plan to cover in each lecture. Students should consult the course web page for updates to this schedule and for the readings that should be completed for each topic.

	Date	Topic
1.	2/2	Being Digital
2.	2/5	Encoding in Binary
3.	2/7	Providing a GUI Interface
4.	2/9	Network Communication in Java
5.	2/12	Huffman Codes for Text
6.	2/14	Conditional Execution
7.	2/19	Transmitting Binary Signals
8.	2/21	Transmission Delays and Framing
9.	2/23	Processing Strings in Java
10.	2/26	Iteration
11.	2/28	Java Class Definitions
12.	3/2	Java Class Definitions
13.	3/5	Ethernet Operation
14.	3/7	Recursive Class Definitions
15.	3/9	Recursive Class Definitions
16.	3/12	Ethernet Performance
17.	3/14	Ethernet Performance
18.	3/15	Midterm
19.	4/2	Switched Networks
20.	4/4	Indexing & Image Manipulation
21.	4/6	Arrays & Image Manipulation
22.	4/9	Internetworking
23.	4/11	IP Configuration and Forwarding
24.	4/13	Image Compression
25.	4/16	Image Color Quantization
26.	4/18	Routing Algorithms
27.	4/20	Heuristics & Intractability
28.	4/23	TCP & Transport Protocols
29.	4/25	TCP Retransmission
30.	4/27	TCP Connection Maintenance
31.	4/30	Error Detecting Codes
32.	5/2	Error Detecting Codes
33.	5/4	Error Detecting Codes
34.	5/7	Network Address Translation
35.	5/9	Network Address Translation
36.	5/11	Review