CSI 310 – Data Structures – Spring 2006
http://www.cs.albany.edu/~sdc/CSI310 (Official location for announcements)

Course Policies

Instructor: S. Chaiken
Office Hours: T, Th: 1:30PM-2:30
LI 96H, 442-4282 Wed: 9:30-11:00AM
sdc@cs.albany.edu other times drop-in if I’m not busy

Prerequisite: CSI201 or equivalent CS1 course in fundamental programming (variables, arrays, control statements, basic problem solving) with C++, beginning programming/testing/debugging under Unix.

Teaching Assistants: TBA Scheduled office hours will to be held in lab room HU-025.

Required Texts:

1. Michael Main and Walter Savitch, “Data Structures and Other Objects Using C++”, 2nd or 3rd edition, Addison-Wesley. Web material supporting this text, including complete file copies of the example programs that are explained piece-by-piece in the text. Some people complain, with justification, the whole solution examples are not printed in the text. This web resource is the answer to those complaints.

2. Your CS1 C++ textbook and/or other references for C++ and other programming fundamentals. The book by Walter Savitch, “Problem Solving with C++, the Object of Programming”, Addison-Wesley, has been used at U. Albany for CSI201 and is recommended. If your CS1 course was in Java, you should be able to make the transition to C++ with the help of this book. You might want to attend the CSI201 lectures this semester too.

3. Other handouts, online materials and Web references that will be provided through the lectures and labs throughout the semester. Some readings in “Writing Solid Code” by Steve Maguire from library reserve will be assigned.

4. Project assignments are written as software specifications, with pedagogical explanations and expositions. Every word on project assignments is required reading. The professor may refuse to give personal explanations of details that appear in written form which the student hasn’t read.

5. Some content from Peek, et. al., “Learning the Unix Operating System,” 5th edition, or equivalent material from ITSUNIX helpdesk, Unix online manual “man”, Web resources or other books. The topics include basic Unix command line shell usage and scripting, and working with files and directories in particular. Details will be specified in handouts, assignments, labs and lectures.

1 Learning Objectives

1. To develop fluency and problem solving skills in computer programming to levels expected after the second semester of University Computer Science major study. This includes the
writing of preconditions, postconditions and invariants to precisely document interfaces and implementations.

2. The principles, practice and elementary analysis of data organization in certain (more or less non-obvious) ways that are well-known and understood by computer scientists, professional programmers and software designers. The power, efficiency and, for some beginners, the learning difficulty of these linked and/or arrayed data structures results from data that refers to data.

Within these data structures, there are pointer and/or array index DATA whose purpose in to locate OTHER DATA. (The other data is accessed, referred to, or retrieved in short, constant time; no searching is needed.) The other data is often structured and includes its own pointers and/or indexes to function in the same linked or arrayed manner. Additional efficiency results from other contraints on data organization, for example, the arrays, lists or trees being sorted or ordered.

We will also introduce examples of similar data organization from the Web, the Internet and computer operating systems/environments.

3. The powerful property defining, problem solving, algorithm structuring and computer coding technique of recursion: Defining something in terms of itself in a way that avoids the philosophical problem of circularity. (By the way, abstract mathematical approaches to recursion are taught in Albany’s CSI210 course.)

Important but secondary learning objectives include some principles of object oriented design and programming, as supported by some features of C++. They also include additional understanding and proficiency with computing environments (such as command line shell controlled Unix systems as opposed to more popular and easy to use graphical environments) that expose and enable programmers to control, customize and script the steps relevant to their tasks. Technical proficiency with a particular programming language (C++) and operating environment (Solaris Unix) is included in the curriculum. Finally, we will introduce comparisons to our subject issues within C, Java, XML and other programming and markup or specification languages.

A. Laboratory Exercises and Technology Tutorials: The schedule for lab sessions and staff office hours will be linked from the course web page.

The lab periods will be used for learning and discussion exercises conducted by the assistants plus assigned self-paced tutorials. The self-paced tutorials will cover the application of technology such automated builds, a graphical debugger, and program analysis tools to project topics. More advanced tutorials will supplement the programming and data structures course topics with items like graphics.

Each technology tutorial exercise will be posted on or before the Tuesday of the week it should be begun. That way, the lab instructors can provide some assistance to get you started.

The due time for electronically submitting the reports on the tutorials and other specified work of the lab is the end of the week (Friday, 11:59PM) following the week of the exercise. If classes are not in session that Friday, the due time will be one week later. Analysis shows you will have at least 11 days to submit the results after receiving the assignment.

You must register for some lab session. If the session is overfull, the professor will accomodate you entering a different session.
Help, explanations and attendance reports can be done with any of the course staff (not just your lab instructor), subject to the due time defined above.

Quizzes will sometimes be given in lectures to monitor student preparation, progress, learning needs and attendance. Points for attendance and acceptable quiz answers will be included in the lab exercise score.

2 Lab Exercises 1 and 2 Assignment

The first Lab exercise is provided on a paper handout on the first lecture.

The second is a tutorial to (1) make and “cd to” a subdirectory, (2) copy (from directory “/acsi310/Lab1”) the 3 files named throttle.h, throttle.cxx and demo2.cxx featured in Chapter 2 of the textbook (“DSO”), (3) compile and test them in the lab under the Albany Unix cluster system, (4) learn and practice some basic uses of the ddd graphical debugger, (5) get started using emacs to create edit files beginning with a shell script to automate “building” the demo2 executable program file, and (6) make, compile, and observe with the debugger modifications of demo2.cxx to explore and experience variables, values, and their behavior under copying under C++ with ddd.

Complete instructions are provided on a separate handout for Lab 2.

IMPORTANT: If you do not have an Albany Unix cluster account now, GO ASAP after class to the helpdesk office (LC-27) for help with getting an account. Tell them you are taking CSI310 and need quick start handouts on common unix commands, getting started on Unix, X-Win32, Emacs, and remote access to Unix.

The Lab (HU-025, basement level) sessions and office hours begin the first week of class–Tomorrow, Jan. 25 and 27.

All students, such as transfer students and others who did not take Albany’s CSI201, and especially those whose programming experience is not C++ under Unix, are expected to get a Unix account and become ready to do these 4 steps before their Lab next week. They should seek help from the student assistants in LC-27 and practice with access and use of Albany’s itsunix systems in LC-3 and LC-4.

Additional help/office hour resources will be posted on the course web page as soon as they are available.

B. Evaluation:

Midterm – Mar. 16, 2006 (Thurs) : In class – 25%
Final – May 15, 2006 (Mon) : 3:30 PM to 5:30 PM – 35%
Laboratory Exercises and Quizzes – – 10%
Programming Projects (5 or 6) – – 30%

Final letter grades will be based on cutoffs applied to your score computed as above and modified according to the following policy:

40% on each of at least two of the 5-6 programming projects, a 40% total score AND a final or average exam grade of at least 40% is the minimum for passing the course (with a D-). 60% on
each of at least three of the 5-6 programming projects, a 50% total score AND a final or average 
exam grade of at least 60% is the minimum for a C. However, I reserve the right to assign a higher 
grade based on a record of substantial improvement.

I will set a grading scale so that C (2.0) indicates the minimum necessary preparation for pass-
ing CSI333 (Programming at the Hardware/Software Interface or “Assembly Language”, the 3rd 
Albany CS Major programming sequence course). Similarly for CSI311 (Principles of Programming 
Languages), if you also get a C or better in CSI210.

Estimated grades based on the midterm exam score will be announced with the midterm results.

The “C” (2.0) or better grade in CSI310 is a prerequisite for the two required CS 
major courses CSI333 and CSI311. (Except if you declared your CS major prior to Sept. 1, 
2002) If your are a CS major, CSI310 is your most important course so far and you should 
aim for an “A”. (Ask any professional programmer.) If you get a B, that is fine. If not, consider 
switching majors. If you get below C, you are probably not in the right major: Your choices are 
to switch or try CSI310 again.

CSI310 is a required course for the CS minor, and it is an elective for ISP and a few other 
majors. For these academic programs, just passing, not a C or better grade, is required.
Details regarding the exams will be announced later. A previous year’s midterm is posted on the 
Web and this year’s will have a similar style.

C. Programming Projects: There will be five to six programming project assignments. These 
assignments will be graded using Albany’s ITS/Academic Computing SUN/Sparc/Solaris Unix 
cluster systems, so you are expected to fully test them there before submitting them for grading. 
Submissions that do not compile and link get ZERO points, automatically! See the 
outline for doing incremental software development included in the project 1 assignment.
(You can log on to these machines over the Internet from computers all over the world, including 
your dorm, LC-3 and LC-4. A few SUN Ultra5 workstations are available to you in LC-3/4 for 
“Power User Experience” but their number is limited. Although the programming work can be 
done on your own computers, especially if you install a unix variant such as Linux or FreeBSD, 
problems (hardware, software, network access) with your system will not be accepted as excuses 
for late or missing programming project or lab completion work.

Programming guidelines and submission information will appear in a separate handout.

Very Important: If you do not turn in syntax error free and generally working programs for at least 
two of the programming projects assignments will result an E grade for the course, regardless of 
laboratory exercises and exam grades.

D. Policy on Cheating:

1. Cheating in an exam will result in an E grade for the course. Further, the students involved 
will be referred to the University Judicial System.

2. The code and any written reports or answers for programming assignments; and written 
answers for lab exercises must be written by yourself. You are welcome to discuss the class 
material, the problems and ideas for solutions; but each person is expected to write the code 
and answers he or she submits independently, without copying.

Cheating in a programming assignment or lab exercise will result in a ZERO for that re-
quirement for all the students involved. Students who cheat in two or more programming
assignments/ lab exercises will receive an E grade for the course.

A report of the every cheating incident will also be made to the Office of Undergraduate Studies in accordance with the University regulations concerning “Penalties and Procedures for Violations of Academic Integrity” in the Undergraduate Bulletin.

We may do automated code comparisons between submissions of current students together with submissions from prior course offerings to detect copying.

3. Violations of Computer and Network Usage policy on laboratory systems are regarded as academic integrity violations, like cheating.

E. Policies on Computer and Network Usage:

1. Attempts to use ECL or ITS computers in violation of the regulations set forth in all the Web pages linked from http://www.albany.edu/its/policies/ may result in account suspension, course failure or referral to University disciplinary action. The implications of the responsible use requirements of shared computer and network infrastructure is an element of this course’s curriculum and so course failure or grade reduction can be imposed by the instructor as a sanction against the violation of laboratory rules and procedures. Failure to comply with instructions given to you by system administration or course instructional team staff members in regard to Lab usage is disruptive and is grounds for sanction.

Willful illegal, malicious or disruptive use, or attempts to disguise one form of computation as another will be taken particularly seriously.

You are also warned against disclosing your password to anybody either willfully or by accident. Another person knowing your password can really get you into trouble. Similarly, if your password is compromised or you change the permissions of your files so that others can read them, they might copy your work and you might get caught when we compare submissions.

2. Ignorance of the quota -v command to monitor your Unix account disk quota and the steps you must take to reduce disk space usage are likely to result in you account becoming unusable when you need it most. If this happens, corrective actions might take several days. Lateness of programming projects or lab exercises will not be excused, nor can any “urgent” system administration actions be taken.

F. Make-up Exams: Make-up exams will be given only for valid and verifiable excuses (e.g. a major medical situation). In such a case, it is your responsibility to contact the instructor ahead of time if at all possible.

G. Policy on I grades: A grade of I will only be given for genuine extenuating circumstances that are beyond your control after the midterm point. Both of the following conditions must be met:

1. Your work must be in good standing as of March 16, 2006, the day of midterm exam, defined as follows: You must have an average score of at least 50% on the programming assignments and at least 50% on the lab exercises due up to that point; and further, your midterm grade
must also be equivalent to at least a C. Therefore, if you miss the midterm or have performed poorly on programming assignments or lab exercises, you are not eligible for an I grade.

2. Written documentation must, upon request, be supplied about the extenuating circumstance either by you or by the University administration. The Dean of Undergraduate Studies and her assistants are there to assist you and will write letters to your professors that request appropriate accomodations.

Under no circumstances will the condition for completing an I grade be that the entire course be retaken later without a new registration.

H. Disabilities, etc: Accomodations will be made for clients of the Office of Disabled Student Services upon adequate prior notice and according to that office’s policies.

Students with genuine continuing hardship situations, or any disability related problems with Lab usage should confer with the professor before February.

I. Attendance: You are responsible for all material presented in the lectures. Some of that material will not be presented anywhere else.

Sometimes quizzes will be given in lectures. They will be counted into the lab scores. No makeup will be given except for genuine, documented extenuating circumstances.

Make sure you have a trusted friend to lend you lecture notes if you are going to miss a lecture.

J. Other Notes:

1. During their office hours, the instructor and the teaching assistants for this class will be glad to help you with the course material and the programs.

2. In addition to the regular office hours, you can also set up an appointment to meet with your instructor and the teaching assistants.

3. We will answer “reasonable” questions by email; which means definite questions that have definite and short answers; and which are at the academic level of the course and on course topics. We may edit and post on the Web questions and answers we think would be helpful to others (but I will email you a personal reply too.) We will remove your name to preserve privacy, unless you specifically state in your message that you would like your name included if we decide to post the question. We might ignore any others. We will not write or copy long explanations that repeat material we presented or are available for you to read.