CSI 310 – Data Structures – Spring 2003
http://www.cs.albany.edu/~sdc/CSI310 (Official location for announcements)

Course Policies

Instructor: S. Chaiken
Office Hours: T, Th: 11:00AM-12:00
Li 96H, 442-4282
sdc@cs.albany.edu
Wed: 9:00-11:00AM
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: Goutam Paul(LI-96Q, goutam@cs.albany.edu, 442-4286), Ben Carle (LI-95D, 437-4435, brc@cs.albany.edu) and possibly another. Office to be announced and posted on the Web.

Required Texts:


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.

The main learning objectives are: (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 beginning analysis of organizing data 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 learning difficulty for some beginners subtlety of these linked and/or arrayed data structures results from the pointer and/or array index DATA whose purpose in to locate (i.e., access, refer to, or retrieve in short, constant time) other DATA. The latter is often structured and includes its own pointers and/or indexes to function likewise. Additional efficiency results from other contraints on data organization, for example, arrays, lists or trees being sorted or ordered. (3) The powerful problem solving and computer code organization 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.

**A. Laboratory Exercises:** The schedule for lab sessions and staff office hours will be linked from the course web page. The lab exercises are designed to be done The due time for completing each lab exercise is the end of the week (Friday) following the week of the exercise.

You must register for some lab session, but you can attend and be graded in any (one or more!) session. (In the unlikely event a session is overloaded, students registered in that session who arrived on time will be seated.)

Help, explanations, check-offs and submissions for pre-lab and post-lab assignments can be done with any of the course staff (not just your lab instructor), subject to the due time for you defined above.

Laboratory exercises are an integral part of the course. Each exercise, assigned the week before, will begin with a prelab assignment to be completed before you do the exercise in the lab. A short presentation will be given at the beginning of each lab section meeting. The checkpoint of starting the exercise in the lab (which will be a little more than attendance) must be done in the lab. You will then work on completing the exercise with any reasonable assistance of the TA, instructor and/or other students. If you do not complete the exercise in the lab session, you can go do it in a subsequent one, on your own, or with the assistance of a TA or the instructor during office hours. Each exercise handout will include a “checkpoint count grading form” that we will sign when we record the grade. Your lab grade is based on number of exercises started in the lab, and the number of checkpoints achieved. We expect you to keep the handout and grading form so that when we do the final count at the end of the semester, you can verify it.

The first lab exercise will be to (1) make and “cd to” a subdirectory, (2) copy (from directory \texttt{/cs310/Lab1}) the 3 files named \texttt{throttle.h}, \texttt{throttle.cxx} and \texttt{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 \texttt{ddd} graphical debugger, (5) get started using \texttt{emacs} to create edit files beginning with a shell script to automate “building” the \texttt{demo2} executable program file, and (6) make, compile, and observe with the debugger modifications of \texttt{demo2.cxx} to explore and experience \texttt{variables, values}, and their behavior under copying under C++ with \texttt{ddd}.

**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 CS310 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) will be open during the 11:15AM-12:10 and the 2:30PM-3:25 session today for you to get assistance with steps (1), (2), (3) and (5). One TA will hold office hours (announced in class) for this at other times this week and on Monday. All students, such as transfer students and others who did not take Albany's CS1201, 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.

**B. Evaluation:**
Midterm - Mar. 13, 2003 (Thurs) : In class  - 25%
Final - May 14, 2003 (W) : 1:00 PM to 3:00 PM - 35%
Weekly Laboratory Exercises (around 10) - - 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 60% 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 passing 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.

Details regarding the exams will be announced later.

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 future handout for incremental software development. (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 requirement 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 APPENDIX G: POLICIES GOVERNING STUDENT USE OF COMPUTING AND NETWORKING FACILITIES AT THE UNIVERSITY AT ALBANY, http://www.albany.edu/judicial_affairs/standardsofconduct.html#IXG 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 your 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 13, 2003, the day of midterm exam; that is, you must have an average score of at least 50% on the programming assignments and at least 50% on the lab exercises completed up to that point. 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 the University administration.

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: Accommodations 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: Although lecture attendance will not be taken, you are responsible for all material presented in the lectures. Some of that material will not be presented anywhere else.

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

It will not be possible for your instructor to conduct makeup classes.

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 on course topics. We might ignore any others.