CS 3100 Data Structures  
Fall 2001

Instructor: Edward L. Lamie, Professor of Computer Science
Office: P-284  Phone: 667-3183  Email: lamie@altair.csustan.edu
Office Hours: Monday and Wednesday 11:15 AM to 12:15 PM, Tuesday 9:30 AM to 12:30 PM

BASIC SCHEDULE INFO:

Class meets Monday, Wednesday, and Friday from 10:10 AM to 11:08 AM in P-107.

Each class session is a mixture of laboratory, lecture, and discussion. The goal is to do whatever is necessary to learn the material. My expectation is that everyone will attend all the classes and keep current with everything that is happening in class.

PREREQUISITE:

It is important that you are adequately prepared to take this course. Check with me if you have not passed:

   CS 2500 (Computer Programming II).

TEXTS:


PROGRAMMING LANGUAGE:

Our text is C++ oriented, and I will present most code samples in C++. If you want to do a programming assignment in a different computer language, speak with me in advance to determine whether we can reach an agreement on some ground rules.

COMPUTER EQUIPMENT

SUN Ultra Workstations:

You will need an account that gives you access to all the Computer Science Department Sun Ultra 10 and Sun Ultra 30 computers. These computers have been specially networked so that you get the same environment, including home directory of your personal files, regardless of which Sun Ultra you log on to.

If you don't seem to have access to the Ultras, then let me know right away, preferably by email. Give me your full name and the name of the class. I'll see to it that an account is created for you, and I'll give you the account information at the next class meeting.

We'll have a special lab session to teach you what you need to know in order to use the Ultras for class assignments. The Ultras are located in the Computer Science Lab: P-288. P-288 is in the Professional Schools Building, in the extreme northwest corner of the north wing.
I will test the programs that you write for this course by compiling and executing them on a Sun Ultra. Therefore you must write programs that will compile and run without errors on these machines. If you are accustomed to a different computing environment, you will have to be careful about this.

Generally, the Ultras are available for access and use on a 24-hour basis. However the campus computer labs do not remain open at all hours. Therefore sometimes network and modem connections will be the only means available for connecting to the Ultras. For more information about network and modem connections, see the sections below titled "MODEM CONNECTIONS" and "INTERNET CONNECTIONS". Also see the document titled "Login Directions" on the class web site.

P-288 will be open mid-morning until about 5:00 PM. During these times, you can be physically present in the CS Department Lab while using a CS Department computer. This can be very beneficial because you then have the opportunity to work and communicate with fellow students and members of the faculty.

Due to budgetary constraints beyond our control, the hours during which P-288 is open are subject to change on short notice. Please check the postings of the lab hours at the entrance to the lab or in the "message of the day" that is printed on your screen when you log on to your Sun Ultra account.

The CS Department is looking for volunteers and work-study students to help keep the lab open longer hours. For further information, ask our system administrator, Julie Gorman: P-288C, 667-3273, email: julie@eos.csustan.edu.

While you are in the CS Lab, you will be able to access Sun Ultra workstations by logging in directly at the console, or by using telnet from some other kind of computer in the lab, such as a PC or a Macintosh.

Telnet makes it possible for many people to use a given workstation simultaneously. Let me know if you need a demonstration of how to get a telnet connection. (You can probably learn how just by asking someone in the lab.) You can also access the Sun Ultras via a telnet connection from most any computer on the campus local network, such as those in P-106, P-120, L-125 and L-145. Generally those labs are open during the same hours that the campus library is open. Please check postings at the labs or ask lab personnel in order to get further information about lab hours. Lab assistants (wearing red vests) should be able to show you how to telnet to a CS Department workstation.

MODEM CONNECTIONS:

Some toll-free telephone numbers are available for making modem connections from Turlock, Modesto, Merced, and Stockton. Check with the computer center (OIT) help desk to get the latest information. The help desk is in L-150, and the telephone number is 667-3687.

INTERNET CONNECTIONS:

You may want to purchase Internet connectivity. This is not a requirement and not necessary for success, but it can be a great time-saver and a convenience once you have gotten past the hurdle of setting everything up. There are several Internet service providers (ISPs) that serve this area. Chances are that members of the class can make good recommendations. We can devote some time to this topic on the first day of class.

If you get an Internet connection, make sure you get telnet and ftp capability so that you can login remotely to your computer account at the college and also transfer files back and forth. To be ready to do telnet and ftp, you may need to get some special add-on software from your ISP.
COURSE OBJECTIVES:

The highest aim of this course is for you to learn how to create good abstract data structures and good algorithms for computer programs. One way to work toward that aim is to study the abstract data structures and algorithms that are widely-known to be "good" for common programming problems, seeking to understand what it is that makes them good. Such a study is another goal of this course.

To understand what makes one data structure or algorithm good for performing a given task, and another not as good, it is necessary that we know how to analyze the potential efficiency of an algorithm using the counting methods of discrete mathematics, elementary probability and statistics, and, occasionally, calculus. So analysis of algorithms is a course objective too.

It is also quite important that you take on and complete challenging programming assignments -- ones that require considerable organizing and/or problem-solving skill on your part.

To learn good techniques of software development is another goal of this course. This includes the use of our campus computers and network.

Finally, successfully completing this course is an important objective for computer science majors. It is an important prerequisite for many other CS courses.

COURSE CONTENT:

We'll start out by quickly reviewing the chapters from Carrano and Prichard's textbook titled *Data and Abstraction: Walls and Mirrors* that were covered in CS 2500. After that, we'll carefully work through most of the remaining chapters of the textbook.

TESTS & GRADING:

There will be two in-term exams and a final exam. Each exam will count 20% of your grade. The other 40% of your grade will come from your score on programming assignments. I will use fractional (plus/minus) grading for this course.

There will be 4 or 5 programming assignments.

I will require you to design each program using a top-down design methodology, modular data structures, and information hiding. You will use the top-down method not only to design the instructions of your program, but also to design the data structures, the documentation, and the test data. A programmer using the top-down method develops the program level by level. The programmer starts with a very simple first-level program design, and works down, step-by-step, to more detailed levels.

I will typically require you to turn in two or three "levels" of your programming assignments so that I can check your progress in employing the top-down design method. For example, for a programming assignment worth 100 points, the top and second levels might count 25 points each and the final level 50 points.

EXCEPTION:

In order to pass this course, you must pass both the programming part and the classroom test part. Also, you must turn in a working solution to each of the programming problems. Late assignments are not permitted, so solutions to programming problems have to be turned in by the (final) due date for the project in order to receive credit.

LATE ASSIGNMENTS:

Due to many constraints, I cannot accept late assignments. You get credit only for what you turn in on time.
On the other hand, we can have class discussions about due dates and I may be willing to move a due date back if doing so can be justified to my satisfaction. If you are uncomfortable with a scheduled due date, please bring this to my attention as soon as possible, and not at the last minute.

If you are not finished with an assignment on the due date, please turn in whatever you have done. It may be possible to get up to about half credit, depending on the quality of the work and the circumstances. If you turn in nothing, you get no credit.

NOTE: Please bear in mind that you can fail this course just by missing some due dates. If you get bogged down on some aspect of an assignment, get some help from me. Speak to me after class, during my office hours, or send me a message by email. Do this well before the assignment is due.

SPECIAL PROBLEMS:

If you think you are going to be unable to take a test or turn in an assignment on time due to circumstances beyond your control, then let me know at the very earliest time possible. I'll try to make some fair arrangement with you.

RESPONSIBILITY FOR YOUR OWN WORK:

I encourage you to discuss programming assignments with your classmates or whomever you wish. It is very appropriate and worthwhile for students to get together to test their understanding of an assignment, to discuss advantages and disadvantages of alternative approaches to solving the problem, to talk over the pros and cons of different ways of organizing the program, and to give each other pointers to helpful published materials.

You are welcome to use algorithms or segments of source code that you get from me, from textbooks, or from publications available in our library or through inter-library loan. (Don't expect all of it to work perfectly! You'll have to adapt and even debug some of it.) Aside from the foregoing exceptions, you must be the sole author of all the source code that goes into the programs you turn in for this course.

During exams you are allowed to come forward and ask me to clarify the meaning of a test question. It will be up to me to decide how to answer, or whether to answer at all. The members of the class will be told any new information that arises from such an exchange. Aside from that, you must write exams without the help of other people or the help of any reading materials.

If there is compelling evidence that programs, or exams were not done according to the above criteria, course credit will be withheld.

PARTICIPATION:

I hope to be accessible, helpful, and responsive to your needs during this course. You can do well, and I want that.

To optimize your likelihood of success, participate fully in the class: Attend all classes. Faithfully listen and take notes. Finish all assignments on time. Review your notes before each class. Reply to questions or remarks addressed to you. Ask questions. When you are not getting anywhere on some problem, and it's not the kind of thing you can get resolved in class, seek my help outside of class.

Class web site:

http://www.cs.csustan.edu/~lamie/cs3100/main.htm

Read the following documents on the class web site:
Login Directions
Hello World Exercise
Program Assignment Rules
How To Make A Test Script
Sample submissions and tests