HUMBOLDT STATE UNIVERSITY
CIS 291 – Data Structures in C++
Spring Semester – 2005 – REVISED 1-18-05

Lecture:	Thursday	12:00 - 1:50 pm HGH 203		
Lab:	Tuesday	12:00 - 1:50 pm GH 215		
Instructor:	Sharon Tuttle, Ph.D.		**Office:**	237E NHW
E-Mail:	**st10@humboldt.edu** or sharon.tuttle@humboldt.edu		**Phone:**	826-3381 (Office/Message)

Web Page: follow link from **http://www.humboldt.edu/~st10**;
note the link from the course web page to the Moodle site for this course, also.
(you can also reach your Moodle account via **http://learn.humboldt.edu**)

Office Hours: **Monday, Wednesday** **10:00 am - 12:00 pm,**
or by appointment.

Course description:

Techniques for representing and manipulating data structures using C++. Static and dynamic properties of data structures. Represent structured information such as stacks, queues, trees, linked lists, and graphs. Efficient algorithms for creating, finding, altering, and removing structured data. Prereq: grade of C in CIS 230 or IA. Weekly: 2 hrs lect, 2 hrs lab.

Required text: none

Course software:

You are expected to use the GNU C++ compiler installed on cs-server.humboldt.edu for all course assignments. You will also be making some use of the UNIX operating system, and some tools such as ssh, sftp, and pine.

Note that you may access cs-server.humboldt.edu by using ssh (secure shell) and sftp (secure ftp); ssh may be downloaded for free from:

http://www.humboldt.edu/~its/techguides/software/software.shtml

Note, also, that ssh and sftp are installed on axe, redwood, and sorrel, so if you can reach one of them, you can generally also reach cs-server.

Finally, note that Dev-C++ --- which you may have used in CIS 230 in Fall 2004 --- uses the GNU C++ compiler. Therefore, if you were to develop preliminary versions of your course assignment coding in Dev-C++, it ought to run on cs-server as well. HOWEVER, it is YOUR responsibility to VERIFY that this is so for EACH assignment before turning in its code.

Grading breakdown:

Homeworks		25%
Lab exercises/quizzes/etc.		15%
Exams	Exam #1:	15%
	Exam #2:	20%
Final Exam		25%

Grading scale:

A: >=93	A-: 90-92	
B+: 87-89	B: 83-86	B-: 80-82
C+: 77-79	C: 73-76	C-: 70-72
D+: 67-69	D: 60-66	
F: <60		

Final exam: **TUESDAY, MAY 10th, at 12:40 pm ---
in HGH 203**
(NOTE the above date and time BEFORE making your end-of-semester travel plans...!)

If you would like me to e-mail your grades to you during the course of the semester, you must give me permission in writing on the form given out on the first class session.

Course Policies begin on the next page.

Course Policies:

*	**...related to lecture and lab sessions:**

Regular attendance (at **BOTH** the scheduled **LECTURE AND LAB** sessions) is encouraged and expected. If you should happen to miss a lecture or a lab, then you are responsible for finding out what you missed. "I wasn't there that time" is never an acceptable excuse. Lecture notes are not posted, although many of the projected examples will be made available on the course web site.

There will be graded lab exercises completed and turned in during lab; there may be quizzes and/or other graded activities in lecture that would be turned in during the lecture session as well. If you miss lab or lecture, you will not be able to make these up.

As mentioned above, during lab sessions, there will be lab exercises due during that lab session. Once these are completed, the remaining lab session time may be used to continue work on homework assignments and/or to ask questions about course-related topics. You should not expect to be able to finish homework assignments during the lab sessions --- like any programming course, you should expect to put in a (potentially) large amount of time outside of formal lecture and lab sessions working on homework assignments and practicing concepts discussed.

*	**...related to homework assignments:**

Homework assignments are expected to be completed using the GNU C++ compiler on cs-server (although if I can run your code successfully on cs-server, I don't really care if you did some code development in some other setting. It is, however, YOUR responsibility to VERIFY that your code runs on cs-server for EACH assignment before turning in its code.)

Each homework assignment will state clearly **how** it is to be turned in, and you will not receive credit for it unless you turn it in accordingly. (There **will** be variations in how this will be done for different homework assignments.)

Each homework assignment will be clearly marked with due dates (note that a single homework could have multiple parts with multiple due dates). **No homework assignments will be accepted late. If you wish to receive any credit for a homework assignment, then you must turn in whatever you have done, even if it is incomplete, by the deadline. Partial credit is usually preferable to no credit.** Please note that "the computer/network/etc. going down" is no excuse --- if you leave a homework assignment for the last minute and there are technical problems, you still must turn in whatever you have by the deadline.

E-mailed portions of homeworks must have an e-mail time/date stamp at or before the appropriate deadline; on-paper portions of homeworks (if any) must be given to me, in person, by the appropriate deadline or turned into my mailbox in the Computing Science department office by that deadline **with a time/date notation signed by the department secretary.** On-paper portions of homeworks that I find leaning against my office door will **NOT** be accepted. (If anybody in the CS department is willing to accept it and mark it with a date turned in, that is acceptable, too; they should be able to put it in my box even if the department office is closed, at least during normal school hours.) Note that e-mail from off-campus service providers sometimes has bizarrely-delayed time-date stamps; if yours suffers from this problem, then you must use your axe e-mail to submit assignments.

You are expected to carbon-copy (CC:) to yourself a copy of EACH item submitted by e-mail that is to be graded; this is your only "receipt" that you have turned it in at a particular time. These CC'd copies should be retained at least until a grade has been posted to the course Moodle site for that work. If there is an e-mail glitch or other hardware/software/network problem, you may be asked to submit to me a copy of your CC'd copy; if you do not have it, you will not receive credit for that particular file. This copy is for your protection!

Please note that if you turn in a version of a homework assignment **early**, you are certainly allowed to turn in one or more improved versions before or at the deadline; the latest version will be graded. (This is to encourage experimentation and improving of your work, if inspiration strikes!)

You are expected to start homework assignments promptly after they are made available from the course web page. You are encouraged to send me e-mail with questions you have as you work on these assignments; if you wait until a class meeting to ask such questions, you may not have time to complete the assignment.

Note that programs may be graded on style as well as on whether they run properly and whether they meet homework requirements. (Programs are expected to meet homework handout specifications precisely; when one eventually works within a team on large software projects, following specifications precisely is vital, and may mean the difference between a working product and one that just sits there.) Discussion on programming style will be ongoing throughout the semester.

Some course work may be graded based on whether it has been attempted --- other course work may be graded for correctness, style, and whether it meets homework/lab exercise specifications. You will not know in advance which will be the case.

* **...related to exams:**

Tentative exam dates are given in the course calendar attached. No make-up exams will be given, except by special prior arrangement. Note that the Testing Center's services were severely curtailed in Fall 2003, due to budget cuts

* **...related to ethics:**

All course work --- homeworks, tests, lab exercises, etc. --- are to be the work of each student, **individually**, **unless** it is **explicitly** stated otherwise at the beginning of that course work's description. In general, this is not a group or team programming course. When group work is permitted, the names of all students involved must be included on the work submitted.

For individual work, the general rule of thumb is that you should not show your code to any other student until after **both** of you have turned in the final versions of that code; the only exception is that a student who has completed his/her final version of some code may look at another student's version of that same code in order to help him/her to interpret a compiler or run-time error or to help to ferret out a bug within that code. No student should tell or show another how to write the code for an assignment.

Collaboration or cooperation on assigned course work where it is not explicitly permitted as an option violates course policy. **If two or more students are working together on such an assignment, that is cheating.** You may discuss **general concepts** of the course with one another, of course --- such discussion is encouraged. Studying together for tests is particularly encouraged. But, homework assignments are to be **individual** work. **General** concepts of homework assignments may be carefully discussed --- a useful rule of thumb is that **none** of the students in the discussion should be writing **anything** down during such discussion, or looking at **anyone's** computer terminal --- but, for example, if one student is telling (or, worse, showing) another how to do a particular problem, those students are going too far. (**General** discussion of the meaning of a particular error message is probably all right.) **Copying or modifying of another's computer files, or providing computer files to another, related to homework assignments is definitely over the line, and never justified.**

Note that it is your responsibility to ensure that course assignment files are read-protected. If you are careless about this, and someone else copies your work, you will share the penalty. (In particular, be very careful about leaving work on shared network drives in campus labs, or in UNIX directories that are not read-protected.)

I will not tolerate cheating; work showing significant duplication will receive **negative** credit, as will work done by anyone other that the person handing it in. (One receives no credit when required work is not turned in, of course --- copied work deserves **less** than that, thus **negative** credit.) A person providing a file for copying receives the same **negative** credit as the copier. The University's policies on academic honesty will be enforced.

* **...related to other areas:**

You are required to have a working e-mail account that you check regularly. Course-related announcements will normally be sent during the semester via the course mailing list linked to the class roll on Moodle; if Moodle is down or unavailable, however, such course-related announcements may be sent via the course mailing list linked to the class roll on Banner. It is your responsibility, not mine, to ensure that both Banner and Moodle have a working e-mail address for you. I cannot change your default e-mail address in Banner; you must do so. (I am not yet sure how this works in Moodle.) Note that changing your e-mail address on Banner probably does **not** automatically change it on Moodle, nor vice versa.

You are expected to check the course web page and the course Moodle site regularly --- course handouts, homework assignments, and example code from lectures will be posted there, along with course-related announcements as needed, and possibly more. Grades will be posted to the Moodle course site --- you are expected to monitor these, as well, and let me know of any discrepancies.

Incompletes are rarely given and are only given in the case of a true emergency. They are certainly not appropriate for students who find they have fallen behind with assignments, missed a test, or taken on too much academic, work, or family responsibility. For these situations, a drop would be appropriate.

Please note that programming courses can be notorious time-eaters --- you can only learn programming by practicing it, and it takes some much longer than others to master it. In addition, note that the course builds in complexity during the course of the semester. Attending lecture and lab, starting assignments early, practicing programming as much as possible, and asking questions is your best defense!

Please do not hesitate to ask me questions in class, in office hours, by e-mail or by phone (although note that I check e-mail more frequently than voice-mail). And, good luck this semester!

Approximate Course Schedule: (subject to change)
begins on the next page

> Note that this syllabus and the schedule following are subject to change. If you are absent from lecture or lab, it is your responsibility to check on announcements made while you were absent.