

HUMBOLDT STATE UNIVERSITY
CIS 130 – Introduction to Programming
Fall Semester – 2005

Lecture: Wednesday 12:00 - 1:50 pm FR 107
Labs: (each student is registered for ONE of the following:)
Monday 12:00 - 1:50 pm GH 215
Friday 12:00 - 1:50 pm GH 215
Instructor: Sharon Tuttle **Office:** 236 NHW
E-Mail: st10@humboldt.edu **Phone:** 826-3381 (Office/Message)
or sharon.tuttle@humboldt.edu
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: **M, Tu, W, Th** 3:00 - 4:00 pm
or by appointment.

Course description:

An introduction to programming, including programming discipline, problem decomposition, algorithm design, modularity, cohesion, coupling, control structures, simple data structures, testing, error detection approaches, and documentation.

Prerequisites:

- * Math code 40, and
 - * CIS 110 or three units from 171, 172, 173, or 174. [CIS 110 can also be taken concurrently with CIS 130]
- OR
Instructor consent.

Please note: if you are a CIS major, you will need to satisfy the CIS 110 requirement for the CIS major even if you have received consent to take CIS 130 without taking CIS 110 first. Please consult your academic advisor.

Required texts:

How to Design Programs, Felleision, Findler, Flatt, Krishnamurthi, chapters 1-4 (pp. 1-46) modified to use C++ by the course instructor, with permission from MIT Press. [HtDP packet]

- * You need to purchase this from the course instructor (to pay for printing and MIT usage fees).

Introduction to Programming with C++, compiled by Amoussou, Tuttle, Pearson Custom Publishing. [2nd text]

- * This will be used for the second half of the semester. A used version will work fine for this course.

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.

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++, a free C++ environment available in the NHW 244 lab, 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 submitting its code.

Grading breakdown:

| | | |
|-----------------------|-------|---|
| Homeworks | 25% | Grading scale [SUBJECT TO POLICIES BELOW] 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 |
| Lab exercises | 12.5% | |
| In-lecture activities | 12.5% | |
| Exams Exam #1: | 15% | |
| Exam #2: | 15% | |
| Final Exam: | 20% | |

Important grading policies:

1. To receive a **C** in this course, the **average** of your Exam #1, Exam #2, and Final Exam must equal or exceed **70%**.
2. To receive a **C** in this course, the **average** of your homeworks must equal or exceed **70%**.
3. If requirements #1 and #2 are met, then your semester grade will be computed according to the grading breakdown percentages given above. (Note that you could still receive a grade lower than C in this case, if you are borderline and your lab exercise and in-lecture activities grades are poor.)
4. If requirements #1 and #2 are not met, your grade will be either the letter grade computed according to the grading breakdown percentages given above or **C-**, whichever is lower.

Final Exam

The Final Exam for this course is scheduled for **10:20 am on MONDAY, DECEMBER 12th** in **Forestry 107**. Note this time and date BEFORE making your end-of-semester travel plans.

Additional Course Policies

5. You are expected to read this syllabus and be prepared to sign a statement that says you have received and understand these policies.
6. I check my e-mail (sharon.tuttle@humboldt.edu or st10@humboldt.edu) at least once a day on weekdays. Include **CIS 130** in the Subject: line for expedited handling.
7. If you would like me to e-mail certain course grades to you during the semester, then you must give me permission in writing on the course information form.
8. Regular attendance at lecture and at your registered lab session is 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. Graded lecture and lab activities missed cannot be made up later.
9. Graded lecture and lab activities will be given during most class meetings. The lowest two lecture activities' grades and the lowest two lab exercise grades will be dropped from your grade. That means you can be absent from the graded in-lecture and in-lab activities twice each without direct penalty, for whatever reason (although you are, of course, still responsible for whatever material is covered on those days, and it is your responsibility for determining what that material is). Illness does not allow you the privilege of making up in-lecture activities or lab exercises --- plan to be sick no more than twice. [terminology blatantly borrowed from A. Burroughs CIS 315 syllabus]
10. You are expected to prepare (read and study) assigned readings before class and to participate in class discussions. Projected examples will be utilized frequently during discussion. You should understand that there will be material in the reading that will not be discussed in lecture/lab, and material in the lectures/labs that may not be found in the reading. You are responsible for both.

11. As previously mentioned, during lab sessions, there will be lab exercises due during that lab session. Once a lab's lab exercise is complete, the remaining lab time should be used to continue work on the current course homework assignment, to practice course concepts, 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 college-level course, you should expect to put in a (potentially) large amount of time outside of scheduled class meetings (lectures and labs) doing the assigned reading, working on homework assignments, and practicing concepts discussed.

12. Remember the general rule of thumb for college-level courses --- to be successful in a course, you should plan to spend at least 3 hours outside of class for each 1 hour of college course credit. That implies an estimate of 9-10 hours a week spent outside of class for this course.

However, you should also be warned that:

- * Programming courses can be notorious time eaters --- occasionally, a problem with your program will take large amounts of time to locate and fix. Starting early enough so that you have time to ask me questions when you run into problems can help with this.
 - * You can only learn programming by practicing it, and it takes some much longer than others to master it. Practicing programming as much as possible helps.
 - * The course will intensify as the semester progresses --- as you are able to do more, you will be expected to do more.
13. Each homework assignment must be submitted as is specified on that assignment in order to be accepted for credit. This may vary for different homework assignments. Often, parts of homework assignments will be submitted using a special tool on cs-server. Code that does not run on cs-server will not receive credit; remember that it is your responsibility to verify that your code runs on cs-server for each assignment before turning in its code.
 14. Each homework assignment will be clearly marked with one or more due dates (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.

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.

You may turn in multiple versions of homework files before the deadline; I will grade the latest pre-deadline submission unless you inform me otherwise. This is to encourage you to turn homework parts in early (since you will know that you can always turn in an improved version if further inspiration strikes).

15. The tool that you will be using to submit homework parts results in a file that serves as your "receipt" for having submitted items. You are expected to retain these "receipt" files at least until a grade has been posted to the course Moodle site for that homework. If there is a system glitch or other hardware/software/network problem, you may be asked to make me a copy of one or more receipt files; if you do not have them, then you will not receive credit for the files involved. These receipt files are for your protection!
16. The successful student in this course will 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 class meetings or even office hours to ask such questions, you may not have time to complete the assignment.

It is not a problem if you send me a question and then end up answering it yourself before you receive my answer; likewise, it is not a problem if you end up sending me several questions (as you work on different parts of an assignment while awaiting earlier answers).

17. Note that programs may be graded on **style** as well as on whether they run properly and whether they precisely meet the homework specifications and requirements. Discussions on programming style will be ongoing throughout the semester.

Programs are expected to meet homework handout specifications precisely; when one eventually works within a team on large software projects, following the specifications precisely is vital, and can mean the difference between a working product and one that just sits there.

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 specifications. You will not know in advance which will be the case.

18. Exam dates are given in the course schedule below. 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.
19. All course work is to be the work of each student, **individually**, **unless** it is **explicitly** stated otherwise at the beginning of that course work's description. Except for explicit exceptions, this is **not** a group or team programming course.

You are expected to be familiar with Humboldt's policies on academic honesty:
http://studentaffairs.humboldt.edu/judicial/academic_honesty.php

For individual homework assignments, students may discuss general approaches **as long as no one involved in the discussion is writing anything down or typing anything during such discussions**. Students may also help one another in determining causes of program bugs, or in determining the meaning of compiler error messages. However, students may not work together to complete homework assignments, one student should not instruct another in how to write the code for an assignment, and **any type of copying or modifying of another person's computer files, OR of providing computer files to another, related to homework assignments is definitely over the line, and never justified**. This applies to copying of documentation and comments as well as to copying of program code.

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/LINUX directories that are not read-protected.)

Learning takes hard work; when students turn in others' work as their own, it is a slap in the face to those seriously interested in learning. Not turning in an assignment results in no credit for that assignment, of course, but that is an honest grade. Work that violates the course honesty policy deserves a lower grade than that, and therefore the course policy is that work violating this policy will receive **negative** credit. A person providing a file for copying receives the same **negative** credit as the copier. Repeat offenses will be handled according to University policies.

20. You are required to have a working e-mail account that you check regularly, and it is your responsibility to set Banner and Moodle so that such an address is your preferred e-mail address. 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. Note that changing your e-mail address on Banner probably does **not** automatically change it on Moodle, nor vice versa.
21. You are expected to check the course web page and the course Moodle site regularly --- course handouts, homework assignments, example code from lectures, and possibly more will be posted to the public course web page, and grades will be posted to the course Moodle site. You are expected to monitor your posted grades and let me know about any discrepancies.
22. Incompletes are rarely given and only in the case of a true emergency. They certainly are not appropriate for students who find they have fallen behind on assignments, missed a test, or taken on too much academic, work, or family responsibilities. For these situations, dropping the course would be appropriate.

22. You are **encouraged** to ask me questions in class, in office hours, and by e-mail. The most successful students are those who are not afraid to ask questions early and often (I will gently let you know if you are overdoing it), who do the assigned reading, who attend lecture and lab regularly, who start assignments early, and who practice programming as much as possible.

Approximate Course Schedule: (subject to change)

Note that this syllabus and this schedule 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.

Exam #1: Wednesday, October 5th
Exam #2: Wednesday, November 16th
Final Exam: Monday, December 12th, at 10:20 am

Tentative topics ordering (subject to change):

- * Intro to course
- * Intro to numbers and arithmetic expressions - Reading: [HtDP packet] through Section 2.1
- * Intro to parameters, functions, word problems, and error types - Reading: [HtDP packet] Sections 2.2, 2.3, 2.4
- * Designing programs and intro to the design recipe - Reading: [HtDP packet] Section 2.5
- * Auxiliary functions - Reading: [HtDP packet] Sections 3, 3.1
- * Named constants - Reading: [HtDP packet] Sections 3.2, 3.3
- * Conditional expressions, bool type, boolean expressions, functions that test conditions, intervals and interval boundaries - Reading: [HtDP packet] Sections 4, 4.1, 4.2
- * if-statements and conditional functions - Reading: [HtDP packet] Sections 4.3, 4.4
- * Local variables, assignment statements, the function main(), #include statements for standard libraries, intro to simple cin and cout, #include statements for your own functions, compiling your own functions and programs - Reading: [HtDP packet] Sections 5, 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7
- * cout formatting - Reading: [2nd text] Chapter 12
- * Intro to while loops - Reading: [2nd text] Chapter 10
- * Intro to for loops and arrays - Reading: [2nd text] Chapters 16 and 18
- * Intro to switch statements - Reading: [2nd text] Chapter 17
- * Pass-by-value parameters and pass-by-reference parameters - Reading: [2nd text] Chapter 14
- * [if time permits] File input/output - Reading: [2nd text] Chapter 21