

The University of Texas at San Antonio
Division of Computer Science
San Antonio, Texas 78249

CS 1713, Introduction to Computer Science
Spring Semester, 1998
Course Information

Course: CS 1713-001, Introduction to Computer Science.

Times, Places, Dates:

Section 003 TR 9:30–10:45 am, 3.04.20 HSS.

(Students must also take:

CS 1711-003, TR 8:00–8:50 am, 3.04.20 HSS.

First class meets Tuesday, January 13, 1998.

Final exam: Monday, May 4, 10:30–1:15 pm.)

Instructor:

Neal R. Wagner, Office: 3.02.16 SB, Telephone: (210)458-5550.

E-mail: wagner@cs.utsa.edu, wagner@runner.utsa.edu

Web-page: <http://www.cs.utsa.edu/faculty/wagner.html>

Course Description:

From the UTSA Catalog: *Introduction to basic concepts of computer science. Functional components of computers, data representation, problem solving methods, algorithm development, and programming using a high-level programming language.* In practice, a lot of course time will be spent on the C language and on the Unix operating system.

Prerequisites: MAT 1093, Precalculus.

The prerequisite MAT 1093 is strictly required. In particular, you should be familiar with logarithms, exponentials, and trigonometric functions.

This course is intended for computer science majors, prospective computer science majors, and others who want the challenge of our most difficult introductory programming class. Previous programming experience is not required, but even science-oriented computer science majors often find the course difficult and time-consuming. There are other introductory computer courses for other majors, such as engineering (CS 2073), life sciences (CS 1073), and education and general education (CS 2083). The College of Business also has introductory courses.

Textbooks (required):

1. *The Art of Programming: Computer Science with C*, by Steven C. Lawlor, West, 1996.
2. *Unix System V: A Practical Guide*, 3rd Edition, by Mark G. Sobell, Benjamin-Cummings, 1995.

Course Requirements (with tentative percentages of grade):

Programming (45% of grade): There will be up to eleven programming assignments, each

handed out one or two weeks before the due date. These must be well-documented, in a way which will be explained in class. Initially you will turn in a single listing giving the source program and any required output. **Note:** The programming assignments are a basic course requirement, like engineering lab work, and not at all like homework in a calculus class. You *must* complete most of the programs in order to pass the course.

Two In-class Exams (15% of grade each, or 30% total) and Final Exam (25% of grade): The final will be nearly twice the length of the other exams.

Laboratories:

The laboratory meets Tuesday and Thursday at 8am. All students should show up for each Tuesday lab. There will be presentations, required work, quizzes, questions answered, etc. Students who do not complete all the work or have further questions should come to the Thursday lab session for additional help.

Late Assignments:

For full credit, programming assignments must be completed and turned in at the start of class on the Thursday shown in the Syllabus. If this deadline is missed, then assignments may be turned in at the start of class on the next Tuesday and will receive a 25% penalty. Assignments should only be turned in at the start of class on Thursday or the following Tuesday, and not between classes. Answers will often be provided during class on the Tuesday after the assignment is due, so they cannot be accepted after the start of class on Tuesday.

It is important that you start work on each assignment early enough so that you can complete it on time. The material from these assignments builds on concepts learned in earlier assignments. Once a student falls behind in this course, it is difficult to catch up.

Scholastic Dishonesty:

From the UTSA Catalog: *The integrity of a university degree depends on the integrity of the work done for that degree by each student. The University expects a student to maintain a high standard of individual honor in his/her scholastic work.*

Scholastic dishonesty: includes, but is not limited to, cheating on a test or other class work, plagiarism (the appropriation of another's work and the unacknowledged incorporation of that work in one's own written work offered for credit), and collusion (the unauthorized collaboration with another person in preparing course work offered for credit).

In practice, for this course, you may discuss assignments in general terms, but you are not allowed to share any details of actual algorithms or of program code. You may help someone else debug their program as long as you do not start substituting in your own code when there are problems. Turning in a copy of someone else's program, even a copy with extensive changes made to it, is a very serious offense in this course.

Missed Classes:

You are urged to get to know other class members so that you can find out what happened in class if you have to miss.