

CS 1723, Data Structures Spring Semester, 1997 *Course Information and Tentative Syllabus*

Course: CS 1723, Data Structures.

Time, Place, Dates:

Section 001 MWF 10-10:50 am, 3.02.18 BB.

(Students must also take either:

CS 1721-001: M 9-9:50 am, 3.02.28 BB or

CS 1721-002: W 9-9:50 am, 3.02.28 BB.

First class meets Monday, January 13, 1997.

Final exam: Wednesday, May 7, 7:30-10:15 am.)

Instructor:

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

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

Prerequisites:

The official prerequisites are: CS 1713 and MAT 1214. In practice you need programming experience in C, including especially the use of functions with parameters. The topics of C pointers, and recursion are not on the CS 1713 syllabus, and will be covered “from scratch” in this course. C structs will also be covered from scratch. There will be a review of other parts of C.

Class Newsgroups and On-line Access:

This course will use the newsgroup `utsa.cs.1723-1` for announcements by the instructor, and `utsa.cs.1723-1.d` for discussions by students. You should send questions to me by E-mail. I will send you an individual answer, but I will also post an answer to `utsa.cs.1723-1`, with your name and identifying information stripped off.

I plan to carry out much of the work for this course over the network. All materials, including assignments, sample programs, and even this document, will be available on runner in the directory:

```
runner$ cd ~wagner/pub/CS1723
```

Textbooks (required):

- ☛ *Data Structures and Program Design in C*, Second Edition, Kruse, Leung, and Tondo, Prentice-Hall, 1996. (This is the main text for the course. We may not always be following this text exactly, but it will be a useful reference for all the material. Much of the course material will be presented using handouts.)
- ☛ *The C Programming Language*, Kernighan and Ritchie, Prentice-Hall, Second Edition, 1988. (This is the standard “bible” on the C language, written by one of C’s originators. The text also covers data structures material. The C code in this book, including its formatting, can be used as a model for the code for the course. This is a sophisticated text that can be difficult to understand.)

Course Objectives:

- Introduction to data structures such as arrays, structs, lists, stacks, queues, trees, and their variations, including linked-list versions.
- Introduction to recursion and its uses.
- Discussion of hashing techniques.
- Elementary searching and sorting methods.
- Details of the C programming language and of the Unix environment. This includes especially C pointers, structs, and strings.

- Introduction to object-oriented programming concepts, such as abstract data types, information hiding, objects (with associated data and functions).
- Discussion of storage management, including C stack-based storage, C static storage, and storage created with `malloc` and accessed through pointers. Also simulation of pointer storage using arrays (allocation, use and deallocation).
- Introduction to algorithm design and analysis techniques.
- Introduction to various applications of data structures.

Course Requirements (with *tentative* percentages of grade):

- Programming (7–9 assignments at roughly 5% each = 45% of grade): There will be up to 9 programming assignments, each handed out one or two weeks before the due date. Some will be harder than others, so the credit for these assignments will vary. These must be well-documented and well-structured, as illustrated in your textbooks. Usually 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 (2 at 15% each = 30% of grade)
- and Final Exam (25% of grade): The final will be nearly twice the length of the other exams. It will be comprehensive, but with emphasis on material covered after the second hour exam.

Late Assignments:

Programming assignments must be completed on or before the due date shown in the Syllabus. Assignments will still be accepted with no penalty if they are completed by midnight on the due date, as shown by the date on the listing, and are turned in by noon the next day. Assignments completed after midnight on the due date will not be accepted.

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.

Laboratory classes:

The laboratory classes are there to review material about the C language and to go over questions and problems related to the programs. The laboratory instructors will keep track of attendance and participation. You will get a separate grade in the lab, not necessarily the same grade as in the lecture.

CS 1723, Tentative Course Syllabus, Spring 1997

	Dates	Kruse	K&R	Topics Covered
1	Jan 13, 15, 17	1-2	1-4	Intro, course organization. C review, esp. operators and precedence, functions and parameters. [Classes start Jan 13]
2	Jan 22, 24	3.1 4.1-4.3	5	Stacks and queues, using arrays. static and auto storage [Jan 20 is MLK Day] [Program 1 (3/2 problem): Due Jan 22.]
3	Jan 27, 29, 31	4.5, 4.6 3.1.6, 3.2	6	Dynamic memory allocation, linked lists, recursion.
4	Feb 3, 5, 7	same	6	Same as week 3. [Program 2 (Simulation using queues): Due Feb 5.]
5	Feb 10, 12, 14	5.3	5	Strings in C, pointers, pointer arithmetic. [FIRST EXAM : Feb 14.]
6	Feb 17, 19, 21	-	7	Files in C. [Program 3 (Translation to RPN): Due Feb 19.]
7	Feb 24, 26, 28	9	6	Binary trees, recursion.
8	Mar 3, 5, 7	9	6	More binary trees, recursion. [Program 4 (All words in a string): Due Mar 5.]
9	Mar 10, 12, 14	6,7	5,6	Searching and sorting. [Last day to drop: Mar 14.]
	Mar 17-21			Spring Break.
10	Mar 24, 26, 28	6,7	5,6	More searching and sorting. [Program 5 (Concordance): Due Mar 26.]
11	Mar 31, Apr 2, 4	-	5	2-dim arrays and pointers. [SECOND EXAM : Apr 4.]
12	Apr 7, 9, 11	-	5	structs and pointers. [Program 6 (Recursive bdry recog.): Due Apr 9.]
13	Apr 14, 16, 18	-	4	C preprocessor.
14	Apr 21, 23, 25	-	-	Object-oriented programming in C. [Program 7 (To be determined): Due Apr 23.]
15	Apr 28, 30	-	-	Leeway. [Study: May 1, 2.]

Programs due: Wednesdays.

Exams: Fridays.

CS 1721, Tentative Laboratory Syllabus, Spring 1997

	Dates	Topics Covered
1	Jan 13, 15, 17	Review of C, examples with structs.
2	Jan 22, 24	Storage classes (static, auto). [Monday students attend Wed.] [Jan 16 is MLK Day]
3	Jan 27, 29, 31	Working with linked lists.
4	Feb 3, 5, 7	More work with linked lists.
5	Feb 10, 12, 14	Working with strings and pointers.
6	Feb 17, 19, 21	Files in C.
7	Feb 24, 26, 28	Binary tree examples.
8	Mar 3, 5, 7	More binary tree examples.
9	Mar 10, 12, 14	A specific sort algorithm. [Last day to drop: Mar 14.]
	Mar 17–21	Spring Break.
10	Mar 24, 26, 28	Another sort algorithm.
11	Mar 31, Apr 2, 4	2-dim arrays and pointers.
12	Apr 7, 9, 11	Structs and pointers.
13	Apr 14, 16, 18	C preprocessor.
14	Apr 21, 23, 25	C++
15	Apr 28, 30	C++ [Study: May 1, 2.]