CS 1723, Data Structures—Fall Semester, 1998
Course Information and Tentative Syllabus

Time, Place, Dates:

CS 1723, Section 001 MWF 1-1:50 pm, 3.04.06 BB.
(Students must also take: CS 1721-001: W 12–12:50 pm, 3.04.06 BB
First class meets Monday, August 24, 1998.
Final exam: Wednesday, December 9, 1:30-4:15 pm.)

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, but quickly. There will be a review of other parts of C. You also need to know the rudiments of Unix from a user perspective.

Class Newsgroups and On-line Access:

This course will use e-mail on runner for announcements by the instructor and for answers to student questions. You should send questions to me by e-mail. I will send you an individual answer, but in most cases I will also send the question andan answer to all students, with the 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:** *The C Programming Language*, Kerhighan 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. Though an excellent book, this text is sophisticated and can be difficult to understand.)
- **Possibly helpful:** *Data Structures in C*, Drozdek and Simon, PWS, 1995. (This is a useful reference for the course, but I will not be following it. Instead, much of the course material will be presented using handouts.)
- **Possibly helpful:** *Introduction to Algorithms*, Cormen, Leiserson, and Rivest, MIT Press, 1990. (Another useful reference for the course that I will use on occasion. This book is also used for the Algorithms course, CS 3343.)

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. This includes especially C pointers, structs, and strings.
- Familiarity with the Unix environment.
- 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, about 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 may 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 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).
- 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.

Laboratories:

The laboratory meets Monday and Wednesday at 9 am. All students should show up for each Monday 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 Wednesday lab session for additional help.

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.

Late Assignments:

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

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.

CS 1723, Tentative Course Syllabus, Fall 1998

	Dates	D & S	K&R	Topics Covered
1	Aug 24, 26, 28	1	1-4	Intro, course organization. C review, esp. operators and precedence, functions and parameters. **[Classes start Aug 24]**
2	Aug 31, 2, 4	6	5	Stacks and queues, using arrays. static and auto storage **[Program 1 (3/2 problem): Due Sep 4.]**
3	Sep 9, 11	5, 7	6	Dynamic memory allocation, linked lists, recursion. **[Sep 7 is Labor Day.]** **[Program 2 (RPN evaluation): Due Sep 11.]**
4	Sep 14, 16, 18	same	6	Same as week 3. **[Program 3 (Queues): Due Sep 18.]**
5	Sep 21, 23, 25		5	Strings in C, pointers, pointer arithmetic. **[Program 4 (Strings): Due Sep 25.]**
6	Sep 28, Oct 30, 2	-	7	Files in C. **[FIRST EXAM : Oct 30.]**
7	Oct 5, 7, 9	7, 8	6	Binary trees, recursion. **[Program 5 (Translation to RPN): Due Oct 9.]**
8	Oct 12, 14, 16	7, 8	6	More binary trees, recursion. **[Program 6 (All words in a string): Oct 16.]**
9	Oct 19, 21, 23	11	5,6	Searching and sorting. **[Last day to drop: Oct 23.]**
10	Oct 26, 28, 30	11	5,6	More searching and sorting. **[Program 7 (Concordance): Due Oct 30.]**
11	Nov 2, 4, 6	-	5	2-dim arrays and pointers. **[SECOND EXAM : Nov 4.]**
12	Nov 9, 11, 13	-	5	structs and pointers. **[Program 8 (Recursive bdry recog.): Nov 13.]**
13	Nov 16, 18, 20	-	4	C preprocessor.
14	Nov 23, 25	-	-	Object-oriented programming in C. **[Program 9 (Sorting/searching): Nov 25.]** **[Nov 26 is Thanksgiving]**
15	Nov 30, Dec 2	-	-	Leeway. **[Study: Dec 3, 4.]**
	Dec 9			**[Final Exam, 1:30–4:15 pm]**

Programs due: Fridays.
Exams: Wednesdays.

CS 1721, Tentative Laboratory Syllabus, Fall 1998

	Dates	Topics Covered
1	Aug 26	Review of C, examples with structs.
2	Sep 2	Storage classes (static, auto). **[Jan 19 is MLK Day.]**
3	Sep 9	Working with linked lists.
4	Sep 16	More work with linked lists.
5	Sep 23	Working with strings and pointers.
6	Sep 30	Files in C.
7	Oct 7	Binary tree examples.
8	Oct 14	More binary tree examples.
9	Oct 21	A specific sort algorithm. **[Last day to drop: Oct 23.]**
10	Oct 28	Another sort algorithm.
11	Nov 4	2-dim arrays and pointers.
12	Nov 11	Structs and pointers.
13	Nov 18	C preprocessor.
14	Nov 25	Leeway
15	Dec 2	Leeway **[Study: Dec 3, 4.]**
	Dec 9	**[Final Exam, 1:30–4:15 pm]**

Labs are used to help with assignments and to answer questions, as well as to cover special material.