

CS 2213/22111: ADVANCED PROGRAMMING  
SPRING 2009, COURSE SYLLABUS

CS 2213.001:	Mon./Wed./Fri.	12:00-12:50pm	in BB 3.01.12
CS 2211.001:	Mon.	1:00-1:50pm	in SB 3.02.10A
CS 2211.002:	Wed.	1:00-1:50pm	in SB 3.02.10A

### Instructor

Instructor: Dr. Jeffery von Ronne  
Phone: 210-458-5667  
Email: vonronne@cs.utsa.edu  
Office Location: S.B. 4.01.34  
Office Hours: Wed. 3-4pm, Thu. 3-4pm, and by appointment

### Textbooks

The textbooks for this class are:

1. **Sobell:** *A Practical Guide to Linux Commands, Editors, and Shell Programming*. Prentice Hall, 2005. ISBN-13: 978-0131478237. **(optional)**
2. **Kernighan and Ritchie:** *C Programming Language (2nd Edition)*. Prentice Hall, 1988. ISBN-13: 978-0131103627. **(this or King is required)**
3. **King:** *C Programming: A Modern Approach, Second Edition*. Norton, 2008. ISBN-13: 978-0-393-97950-3. **(this or K&R is required)**
4. **Kalicharan:** *Data Structures in C*. CreateSpace. ISBN-13: 978-1438253275. **(required)**

This class will be using the Linux command-line environment. If you are comfortable running programs from the Linux/UNIX command-line and editing text using vi, then you can probably skip Sobell. Otherwise, **Sobell** is recommended.

The first half of the semester will be an introduction to C. It will be necessary to get either **Kernighan and Ritchie** or **King** by the second week of classes.

The second half of the semester will be spent implementing and using data structures in C. **Kalicharan** will be used as the text for this material.

### Drop Date

March 23 is the deadline for sophomores and above to drop this course with a grade of 'W.'

## **Communications**

Important announcements will be made in class and/or through email to the students' CS department accounts (instructions for accessing these accounts will be given). The instructor normally responds to emails to vonronne@cs.utsa.edu within 1 business day and is happy to answer questions over email. The CS department will set up an email list for the class, and the instructor/TA may forward questions of general interest to the email list.

## **Prerequisites**

Students are assumed to already have prior experience programming in Java, including mastery of local variable declarations, if and while statements. Students are expected to have prior or concurrent experience using basic data structures in Java.

Formally:

1. CS 1713 and CS 1711: Introduction to Computer Programming II (Prerequisites)
2. CS 2123 and CS 2121: Data Structures (Corequisites)

Concurrent enrollment in both CS 2213 and CS 2211 is required.

## **Course Description and Goals**

An implementation level view of data structures in a specific language with an emphasis on pointers and memory management. Dynamic data structures such as dynamic lists, heaps, 23-trees, graphs, etc. are considered.

The goals of this course are to provide students with:

1. a detailed understanding of basic dynamic data structure implementation,
2. an understanding of the C memory model,
3. intermediate-level proficiency in C programming,
4. basic proficiency with the UNIX/Linux command-line environment, and
5. the ability to use and adapt basic data structures in C programs and develop C programs in the Linux command-line environment.

## Course Outline

### 1. C Programming

- (a) compiling, simple programs
- (b) basic types, expressions, printf, control
- (c) functions and program organization
- (d) arrays, pointers, and strings
- (e) structures
- (f) dynamic memory

### 2. Data Structures

- (a) linked lists
- (b) trees
- (c) heaps
- (d) graphs (time permitting)
- (e) hashing (time permitting)
- (f) matrices (time permitting)

CS 2211 will include hands-on instructions on tools which are necessary and useful for doing programming in C in Linux as well as discussing general programming/testing/debugging techniques and discussing common problems.

### Assignments, Quizzes, Exams

There will be approximately ten assignments, six 10-minute quizzes, 1 midterm exam, and 1 comprehensive final exam.

Quizzes will be announced at least a week in advance and cover conceptual knowledge and understanding of material. Make up quizzes will not be given except in the case of excused absences as provided for by university-wide policy. The lowest quiz score will be dropped.

Assignments will generally involve the application and synthesis of material in order to write C programs that perform a specified task. Assignments must be submitted using subversion by 5pm on the due date. A one week extension may be requested by emailing the TA and instructor before the due date with an explanation of why the assignment could not be completed on time; a 20% penalty will be assessed.

A midterm covering the C programming language will be given on February 25. A comprehensive final will be given at the designated time (1:30-4:00pm on Tuesday, May 5). Make up exams will not be given except in the case of excused absences as provided for by university-wide policy.

## Course Grades

Final grades will be assigned using the weighted average of the following components:

Quizzes	10%
Programming Assignments	30%
Midterm Exam	25%
Comprehensive Final Exam	35%

A total score of 90% or higher will earn an 'A,' a total score of 80% or higher will earn at least a 'B,' a total score of 70% or higher will earn at least a 'C,' and a total score of 60% or higher will earn at least a 'D'. The instructor reserves the right to adjust the grading scale for the difficulty of work and assign higher grades than those listed above.

There will be a single grade assigned for both CS 2213 and CS 2211.

## Attendance

Attendance in the lecture and recitation is expected and some of the required material may be provided only in lecture or in the recitation.

## Scholastic Dishonesty

Students are expected to behave ethically and **work individually on assignments, quizzes, and examinations.**

Students are allowed to discuss general strategies for solving assignments with other students and tutors. It is no longer a "general strategy" if it gets to the level of detail of what would be done in individual lines of code. And discussions of "general strategy" should not be taking place while either party is editing their source code. Students may also seek help from lab tutors in trying to understand why their code doesn't work. They may also seek help and guidance of any kind from the TA and instructor. Students may not, however, look at any other student's code for an assignment. If collaboration is disclosed at the time of submission, points may be deducted, but no scholastic dishonesty will be considered to have occurred.

If students consult any code not provided by the instructor or TA or from the course text books, they must document this fact (including a copy of or URL for the consulted material) in their submission. Depending on the extent of the outside code use, points may be deducted, but as long as the use is completely disclosed in the submission, no scholastic dishonesty will be considered to have occurred.

Please refer to UTSA's Student Code of Conduct for a description of expected behavior and the potential penalties for scholastic dishonesty.