



# CS 241 Data Organization using C

Spring 2009

---

## Course Instructor:

Joel Castellanos

e-mail: joel@unm.edu

Office:

Room 321 of the Farris Engineering Center (FEC).

Located near the northeast corner of Central Avenue and University Boulevard. This is building #119 in section I-2 of the campus map.

Office Hours:

Mon & Wed: 10:30-11:30 AM

Tues & Thurs: 8:00-9:00 AM and by appointment

## Lab Instructor:

Dongye Meng

e-mail: dymeng@cs.unm.edu

Office: FEC 301A

Office Hours: Friday: 11 AM - 1 PM

## Textbook:

Kernighan, Brian W. & Ritchie, Dennis M. The C Programming Language, 2nd ed., ISBN: 0-13-110362-8.

## Description:

This course is an introduction to the C Programming language for students who already have solid skills in applying basic programming techniques such as loops, conditional control flow, and variable arrays in some other language (such as Java, Matlab, Python, etc.). The course covers memory organization of data storage and its relation to computation and efficiency. Topics include: linked vs. contiguous implementations, memory management, the use of indices and pointers, and an introduction to issues raised by the memory hierarchy. The course also covers fundamental data structures such as stacks, linked lists, and trees. Programming assignments provide practice with programming styles that yield efficient code. Computational experiments investigate the effect of storage design choices on the running time of programs. This class also covers the use of Linux command line tools for file organization, navigation, compilation, low level debugging and Makefiles.



## Grading:

- 50% Programming Projects (approximately 12).
- 20% Exams (midterm and final)
- 15% laboratory programming assignments (attendance required).
- 15% Lecture quizzes (approximately 30, i-clicker, attendance required).

Late projects/assignments will receive a 5% per day penalty.

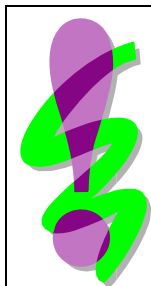
## Working Together:

Working together and helping one another on all projects (but not on exams and quizzes) is highly encouraged. This includes discussion of project *specification*, *algorithms*, *data structures*, and *test cases*. It does not include code. Each person must author his or her own code.

## Cheating:

Cheating will be dealt with very harshly, and includes:

- Copying code from another person or having someone else write your code.
- Copying code from the Internet or another source. (If there's some code that you would really, really like to use, please check with us before you do it.)
- Attempting to disassemble, decompile, or otherwise reverse engineer compiled example programs.
- Allowing another person to copy your code.
- Leaving your code (paper or electronic copies) where others can find it. You responsible for the security of your intellectual property.
- Use of an external libraries other than those included with gcc version 4.2.3 without documenting it. *Note: If you do document usages of external libraries, it will not be considered cheating. However, you still might not receive full marks if the library covers too much of the assignment. It is best to check with one of the instructors before using an external library.*
- Violation copyright or license agreements on external libraries. If you use external library code, it is your responsibility to understand and comply with the appropriate copyright and license issues.
- Violation the University [policy on acceptable computer use](#).



Not being able to explain how some significant part of your code works will result in a zero for the assignment. It does not matter if the reason you do not understand your code is because you did not do the work or because you got your code working by trial and error. If I suspect someone of cheating, the first thing I do is ask that person to explain the code. This is not a quiz you ever want fail. Too much code in the real world is build and maintained by trial and error. It makes for a house of cards. It is not a good way to produce code nor is it a good way to learn.



### Submitting Assignments:

All assignments must be in WebCT in order to receive credit for them. If WebCT is down, then you can e-mail the assignment to the lab instructor in order to prove it was done on time. However, it must be inside WebCT before you can receive credit for it.

### Syllabus:

Week	Topics	Chapter
1 - 2	Types, Operators, Expressions, Scope, Control Flow, Intro to Functions, and Bit Manipulation.	K&R: Chap 1-3
3 - 4	Functions & Program Structure	K&R: Chap 4
5 - 6	Pointers, Arrays, Structures, Linked Data Structures	K&R: Chap 5-6
7 - 8	I/O & System Interface	K&R: Chap 7-8
9 - 10	Linear Data Structures, Efficient debugging techniques Lists, Strings, and Dynamic Memory Allocation	Supplemental reading
11 - 12	Hashing and other efficient data structures	Supplemental reading
13 - 14	Sorting, memory management	Supplemental readings
15 - 16	Makefiles, Debugging, Profiling and performance tuning, Review	Supplemental reading

---