CS 241 Data Organization using C
Fall 2019

Course Instructor:

Joel Castellanos
e-mail: joel@unm.edu
Office: Room 2110 of the Farris Engineering Center (FEC).
Office Hours: Tue/Thurs: 5:00 to 6:00 PM and by appointment

Lab Instructors:

Viacheslav Zhuravlev
email: vzhuravlev@unm.edu
Office Hours: Wednesday 4:00 - 5:00 PM, Farris Engineering Center, room 2045
Dustin Loughrin Office Hours by appointment
email: dloughrin@unm.edu

Textbook:

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

Description:

CS-241 is an introduction to the C Programming language, an introduction to using a command-line interface of the Linux operating system, and an introduction to machine level data organization and memory allocation. Students taking this course should already be familiar with basic concepts of computer programming such as variables, conditional control flow and loops. Developing mastery of these fundamental concepts is one of the goals of CS-241. Students in CS-241 author many C programs: Lab assignments are short and usually variants on examples in the textbook. Projects are more interesting and touch on a wide range of computer applications which have included encryption, numerical analysis, databases, scientific visualization, artificial intelligence, genetic algorithms and games. Many examples used in this course involve implementation of standard data structures such as linked lists and trees. However, mastery of such data structures is not within the course's domain.

The primary goals of CS-241 are for the student to be able to:

1) Read and apply the C syntax covered in the textbook (The C Programming Language by Kernighan and Ritchie).

2) Without a computer, determine the output of C language source code involving triply nested loops, conditional control flow, function calls, pointers, arrays, arithmetic, logical and bit operators, structures and memory allocation.

3) Use a Linux command-line environment to manipulate files, and directories, and to edit, compile, run and debug C programs. This includes the use of simple makefiles and a low level debugger such as valgrind.

4) Implement, in C, any given algorithm with a complexity level equivalent to that of quicksort or a doubly linked list with accuracy, efficiently and clarity.

Grading:

- 40% Programming Projects.
- 30% Exams (midterm and final)
- 20% laboratory programming assignments (attendance required).
- 10% Lecture quizzes (approximately 30, i-clicker, attendance required).

Late projects/assignments will receive a 5% per day penalty.
Assignments can be turned in after the due date, but not after the cut-off date.
The cut-off date is two class meetings after the due date.

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

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:

- Coping code from another person or having someone else write your code.
- Coping 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 source code (.c and .h files) for assignments must be attached in Blackboard Learn in order to receive credit. Assignments have grading rubrics and, for the most part, are graded by compiling them and running them on given data files. If your program does not compile or expects the wrong number of spaces or commas, then your program will miss most of the points on the rubric. **Therefore, you get a much higher grade if you turn in a program that works perfectly with some of the input data than a program that works *almost* correctly with all of the input.**

Lab Attendance

Lab class meets once per week in a computer lab. Lab attendance is taken both at the beginning and end of class. If you are absent, leave early or arrive more than ten minutes late, then you will be marked as absent. Each student may miss up to three lab classes during the semester without there being any direct effect on the grade. Each additional

missed lab class will result in -2% to the student's final lab grade average. There are six lab sections at different times during the week. If for some reason you cannot attend your regularly scheduled lab class but are able to attend one of the other lab classes ***during the same week***, then that other lab can count as your lab attendance.

NOTE: Associated with each lab class there is usually a 20 point lab assignment or a larger point project assignment. If you attend the lab class associated with a particular assignment, then the minimum grade you will receive on that assignment will be 10 points - regardless of its quality or lateness (up to the 7-day late cutoff).

NOTE: Before attending a different lab section, check with that section's lab instructor to make sure there is an open space for you.

NOTE: In order to receive credit for attending a different lab section, ***it is your responsibility*** to make sure the lab instructor of that section ***counts you as present while you are in the lab*** class (NOT after the fact). Your name will not be on that instructor's roster. You must make sure to speak to the lab instructor during the lab class, telling him or her first and last name, and in what section you are registered.

NOTE: The three lab classes that every student may miss without having final grade points deducted are designed to cover sports travel that prevents attending a different lab during the same week, short-term illnesses and other such events. A student that needs to miss many classes due to an extended or reoccurring illness or hospitalization will need to request a grade of *Incomplete* for the semester. With this, arrangements can be made for missed lab attendance and work to be completed during the following semester.

If you feel you need extra help or would simply like to attend lab section in addition to your own, then you are encouraged to do so. First, however, please contact the lab instructor of the extra lab you want to attend to make sure that there is enough space.

Title IX:

In an effort to meet obligations under Title IX, UNM faculty, Teaching Assistants, and Graduate Assistants are considered “responsible employees” by the Department of Education (see pg 15 - http://www2.ed.gov/about/offices/list/ocr/docs/qa-201404-title-ix.pdf). This designation requires that any report of gender discrimination which includes sexual harassment, sexual misconduct and sexual violence made to a faculty member, TA, or GA must be reported to the Title IX Coordinator at the Office of Equal Opportunity (oeo.unm.edu). For more information on the campus policy regarding sexual misconduct, see: https://policy.unm.edu/university-policies/2000/2740.html

ADA:

In accordance with University Policy 2310 and the Americans with Disabilities Act (ADA), academic accommodations may be made for any student who notifies the instructor of the need for an accommodation. If you have a disability, either permanent or temporary, contact Accessibility Resource Center at 277-3506 for additional information.