

# CMSI 284

## COMPUTER SYSTEMS ORGANIZATION / SYSTEMS PROGRAMMING

<http://myweb.lmu.edu/dondi/spring2010/cmsi284>

**Spring 2010** — Doolan 219  
TR 1:35–2:50pm, 3 semester hours  
Office Hours: TR 3–6pm, or by appointment

John David N. Dionisio, PhD  
e-mail: [dondi@lmu.edu](mailto:dondi@lmu.edu), AIM: dondi2LMU  
Doolan 106; (310) 338-5782

### Objectives and Outcomes

This course uses the language of L. Dee Fink's *taxonomy of significant learning* for expressing its objectives and outcomes. Long after the course concludes, my hope is that you will:

- possess *foundational knowledge* of how computing systems are structured and operate at a level that is very close to the actual machines on which they run,
- be able to *apply* this knowledge by being capable of manually encoding and decoding bit representations, writing programs in C and assembly language, and making connections among these levels of abstraction, and
- readily *integrate* both the material from previous courses and material in courses to come into a comprehensive picture of how abstract concepts take concrete form when they approach the physical machine.

### Prerequisites/Prior Background

Programming proficiency in at least one high-level language; a prior course in data structures, such as LMU's CMSI 281. A familiarity with current hardware components and specifications is also beneficial but not absolutely necessary.

### Materials and Texts

This course does not have a preassigned textbook, with materials consisting solely of assorted handouts, articles, and sample code to be distributed throughout the semester. Most of this content comes from Prof. Ray Toal, for whom I am a temporary "sub" while he is on sabbatical.

Additional information is also available on the web; do not hesitate to look for further sources of information regarding the concepts, techniques, tools, and paradigms that we will discuss.

### Course Work and Grading

Course work consists of homework (25%), 1 mid-term (25%), 1 term portfolio (25%), and 1 final exam (25%). Numeric grades  $\geq 90\%$  get an A– or better;  $\geq 80\%$  get a B– or better;  $\geq 70\%$  get a C– or better. I may curve grades upward based on qualitative considerations such as degree of difficulty, effort, class participation, time constraints, and overall attitude throughout the course. Grades are never curved downward.

### Homework

Homework consists of questions, exercises, and programming assignments to be given throughout the semester. Homework is where you can learn from your mistakes without grading penalty: if you do the work and submit it on time, you will get full credit, regardless of correctness. What goes around comes around: the effort you put into your homework pays off in the tests and the portfolio.

The homework submission deadline is always the beginning of class on the designated due date; the due date is encoded in the homework number. Submissions after the deadline receive half credit, period. Extra credit homework may be assigned; fulfilling this is counted on top of the 25% allocation of homework to your final grade.

### Tests

The midterm is initially scheduled for February 25. The final exam is scheduled for May 6. The tests are meant to assess the foundational knowledge that you have acquired so far and, to some degree, your ability to apply it. Tests are open-paper-everything; no sharing. "Open computer" might also be allowed depending on the circumstances. You may neither solicit nor give help during exams. Late and/or missed tests are handled on a case-to-case basis; in all instances, talk to me about them.

## Term Portfolio

For many of the homework assignments that you will get, you may feel that if you “had another go” at the work, you can do much better than in the initial submission. For this course, you *can* have this “other go” at the end of the semester: you will be asked to resubmit a subset of these assignments in a *term portfolio* — a cumulative body of work that demonstrates how well you understand, apply, and integrate the course material. *This* will be graded more closely; presumably, by the end of the semester, you will know this stuff better, and will be able to clean up and improve your prior work.

Exercises will be graded using these criteria:

1. *Content (40%)*: Includes the requested information; correct answers and execution
2. *Organization (30%)*: Structures and presents information well
3. *Writing (20%)*: Precise language, notation, and expressions; clear statements and correct grammar where applicable
4. *Polish (10%)*: Evidence of proofreading and multiple reviews; no misspellings nor typos

Code will be graded along these criteria:

1. *Design (30%)*: Clarity, flexibility, and ease of maintenance; elegance and innovation; applies proper separation of concerns
2. *Functionality (30%)*: Works as intended; produces correct answers/results; performs in a reasonable amount of time; includes tests that demonstrate correct behavior
3. *Naming (20%)*: Clarity and consistency; names correspond to roles, types, or actions
4. *Documentation (15%)*: Abundance of comments in code; genuinely useful supplementary information (e.g., README, additional notes, etc.)
5. *Version control (5%)*: Sufficient frequency; informative commit log

The term portfolio is due on May 6. Late portfolios will not be accepted.

## Attendance

Attendance at all sessions is expected, but not absolutely required. If you must miss class, it is your responsibility to keep up with the course work. Note that the last day to add or drop a class without a grade of W is January 25. The withdrawal or credit/no-credit status deadline is March 26.

## Special Accommodations

Students with special needs who need reasonable modifications, special assistance, or accommodations in this course (such as a documented disability [physical, learning, or psychological]) should contact the Disability Services Office (Daum Hall, Room 224, x84535, <http://www.lmu.edu/dss>) as early in the semester as possible. All discussions will remain confidential. In addition, please schedule an appointment with the instructor early in the semester to discuss any accommodations for this course for which you have been approved.

## University Policy on Academic Honesty

Loyola Marymount University expects high standards of honesty and integrity from all members of its community. All students are expected to follow the LMU honor code, as stated in the *LMU Undergraduate Bulletin 2008-2010*, pp. 58–59 (online at <http://www.lmu.edu/Page13245.aspx#honorcode>).

## Topics and Important Dates

Specifics may change as the course progresses; university dates (italicized) are less likely to change.

<b>January</b>	Overview; computer system organization; numeric and character encoding
<i>January 25</i>	<i>Last day to add or drop a class without a grade of W</i>
<b>February</b>	Digital logic; C programming
February 25	Midterm
<b>March</b>	Processors; the IA-32 architecture; assembly language programming
<i>March 26</i>	<i>Withdraw/credit/no-credit deadline</i>
<i>March 29 to April 2</i>	<i>Spring break; no class</i>
<b>April</b>	System calls; executable files
May 6	<i>Final exam, 11am; term portfolios due</i>

You can view the class calendar on the web at <http://ical.me.com/dondi/LMU>, or via iCalendar at <webcal://ical.me.com/dondi/LMU.ics>.