CMSI 182

Introduction to Computer Science

http://myweb.lmu.edu/dondi/fall2006/cmsi182

Fall 2006 — Pereira 109
TR 1:35pm–2:50pm, 3 semester hours
Office Hours: TR 3–6pm or by appointment

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

Course Objectives

To present a survey of computer science as a field of study, ranging from its theoretical foundations to its practical applications. The course work includes much learning by doing: you will be asked to do work that is similar to what computer scientists — in industry as well as the academe — do everyday. At the same time, the course seeks to show how this work fits into the "big picture" of computer science as a discipline.

Course Requirements

A willingness to listen, to step out of one's "intellectual comfort zone," and to participate actively in class discussions. Some familiarity with computers, operating systems, Web technologies, and mathematics is helpful but not absolutely necessary.

Materials and Texts

- J. Glenn Brookshear, *Computer Science: An Overview*, Ninth Edition, Addison Wesley, 2007 (!).
- Assorted handouts, articles, and sample code to be distributed throughout the semester.

The following texts are recommended and not required — but they *will* fill in a lot of details in case you're interested:

- David Harel with Yishai Feldman, *Algorithmics: The Spirit of Computing*, Third Edition, Addison Wesley, 2004.
- Patrick Niemeyer & Jonathan Knudsen, *Learning Java*, Third Edition, O'Reilly, 2005.

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

Graded coursework consists of homework (25%), 1 midterm (25%), 1 programming portfolio (25%) and 1 final exam (25%). Letter grades are determined as follows: ≥ 90% gets an A– or better; ≥ 80% gets a B– or better; ≥ 70% gets a C– or better. The instructor 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 October 5. The final exam is scheduled for December 14. All tests are open-paper-everything; no sharing. "Open computer" might also be allowed depending on the scope, subject matter, or circumstances. You may neither solicit nor give help while the exam is in progress. Late and/or missed tests are handled on a case-to-case basis; in all instances, talk to me about them.

Programming Portfolio

Many of the homework assignments you will get will involve programming, and when submitted as homework, will be graded as homework (i.e., credit/no credit). However, at the end of the semester, you will be asked to resubmit a subset of these assignments in a *programming portfolio* — a

showcase of sorts for your newfound computer science skills. *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. The code will be graded along these criteria:

1. *Design (30%):* How good is the overall structure of the code? Is it clear, flexible, and easy to maintain? Is it elegant or innovative? How well does it apply the principles of "separation of concerns" and "one change, one place?"
2. *Functionality (30%):* How well does the code work? Does it fulfill requirements? Are its results accurate or correct? Does it perform its tasks in a reasonable amount of time? How well do unit tests validate the code?
3. *Naming (20%):* Are program entities — classes, subroutines, variables, etc. — clearly and consistently named? Do their names correspond to their functions and roles?
4. *Comments (20%):* Are comments provided where appropriate? Are they clear and well-written? Does the code take advantage of any special support for comments provided by the project language or platform (e.g., JavaDoc)?

The programming portfolio is due on December 14. Late portfolios will not be accepted.

Attendance

I am not a stickler for attendance, but I do like having a full class. Remember that the university add/drop with 100% refund deadline is September 1. The deadline for withdrawal or credit/no-credit status is November 3.

University Policy on Academic Honesty

Loyola Marymount University expects high standards of honesty and integrity from all members of its community. Applied to the arena of academic performance, these standards preclude all acts of cheating on assignments or examinations, plagiarism, forgery of signatures or falsification of data, unauthorized access to University computer accounts or files, and removal, mutilation, or deliberate concealment of materials belonging to the University Library.

Course Schedule

This schedule may change based on the actual ebb and flow of the class; deadlines, exams, and university dates (italicized) are less likely to change than lecture topics.

September	Overview; data storage and manipulation; operating systems; your first programs
September 1	*University add/drop deadline for full refund*
October	Networking and the Internet; algorithms and programming languages; software engineering
October 5	Midterm
October 23–24	*Undergraduate holidays; no class*
November	Data abstractions; database systems; artificial intelligence; theory of computation
November 3	*University withdraw/credit/no-credit deadline*
November 23–24	*Thanksgiving; no class*
December 14	*Final exam, 11am; programming portfolios due*

You can view the class calendar on the Web at *http://ical.mac.com/dondi/LMU*. If you have an iCalendar-savvy client (i.e., Mozilla Calendar, Ximian Evolution, KOrganizer, Apple iCal, etc.), you can subscribe to the class calendar at *webcal://ical.mac.com/dondi/LMU.ics*. On-the-fly updates and adjustments to the class schedule will be reflected in this calendar.