COMP 388: Foundations of Computer Science (Section 033)
Spring 2002 Course Information & Syllabus

Instructor: R. I. Greenberg
Department of Mathematical and Computer Sciences
Loyola University
6525 North Sheridan Road
Chicago, Illinois 60626-5385

Phone: (773)508-3991 **Email:** rig@cs.luc.edu **Home page:** http://www.cs.luc.edu/˜rig

Lectures: Monday and Wednesday 11:30 am – 12:45 pm in DH-441.
If you have to miss a class, get notes from another student; mine are typically pieced together from more than one place with a lot of metacomments, which makes it hard for anybody but me to follow them. Also get copies of any missed handouts (available on the web site). The handouts are numbered sequentially, starting with handout 0. On handout 0, you need to fill in some information and return it to me promptly so you can be on the email list and get access to the web site for the course.

Office Hours: In Damen 329C: Mon. and Wed. 9–10 am and 2–3 pm, and Tues. 10:30–12:00.
These are the guaranteed times to find me except as announced in advance. You should also be able to find me at lots of other times; feel free to look for me or make appointments.

Course Objectives: This course will provide graduate students with accelerated coverage of key topics in discrete mathematics (COMP 211) and algorithms (COMP 363) to prepare them for graduate courses in computer science. We will make a rapid passage through such topics as logic, sets, functions, relations, induction, modular arithmetic, elementary combinatorics, graphs and trees, elementary probability, boolean algebra, and finite-state machines. We will also cover asymptotic notation and recurrences and use these tools to analyze some algorithms for fundamental computing tasks such as sorting, searching in ordered data sets, and finding paths (or distances) in graphs.

Prerequisites: One semester of calculus (Math 161) and Structured Programming and Data Structures (Comp 271).

Required Text: Alfred V. Aho and Jeffrey D. Ullman. *Foundations of Computer Science.* Principles of Computer Science Series. W. H. Freeman & Co., C edition, 1995.

Course Requirements: There will be several homework assignments, two midterm exams, and a final. The weightings within the semester grade will be: Homework 20%, Exam I 25%, Exam II 20%, and Final exam 35%.

Homework: Only homework turned in by the due date is guaranteed to be graded. Any special circumstances that cause difficulty in meeting the deadlines should be brought to the attention of the instructor in advance. Homework must be handed in at the beginning of class, since solutions may be handed out in the same class on occasion. Homework turned in to my mailbox will generally not be graded, since I do not check the box continually and cannot generally verify that homework was turned in before solutions were distributed or discussed in class. If you cannot turn in homework in person, you should put it under the door of my office.

Exams: The midterm exams, tentatively scheduled for session 13 and session 20, are 75 minutes long. The final exam is scheduled for 10:20 am – 12:20 pm on Wednesday, May 1.

Collaboration: *No* collaboration is permitted on exams. *Collaboration* on homework is acceptable, but *copying* is not! (Safeguard your files and printouts.) You may discuss solution techniques with other students, but you must write up your solutions independently. If you obtain a solution through research, e.g., in the library, credit your source and write up the solution in your own words.

Tentative Course Outline and Approximate Schedule:

Recommended readings from the text are shown for each lecture. (When selected sections or subsections are listed, it is assumed that you will include the introduction of the corresponding chapter or section.)

1. (1/14) Administrivia, course overview.
2. (1/16) Propositional and predicate logic. Sections 12.2,12.4,12.8,12.9,14.2–4,14.7
3. (1/23) Induction. Sections 2.3–4.
4. (1/28) Loop invariants in programs. Recursive definitions and functions. Sections 2.2,2.5–7. Chapter 1 also contains desirable background.
5. (1/30) Sets. Sections 7.2–3. Relations and their properties. Sections 7.7 and 7.10.
6. (2/4) Functions and their properties, modular arithmetic. Section 7.7 and supplementation.
7. (2/6) Analysis of algorithms. Divide-and-conquer paradigm. Merge sort. Sections 3.2–3,2.8–9.
8. (2/11) Asymptotic notation. Sections 3.4–6.
9. (2/13) Recurrences. Section 3.11.
10. (2/18) Assignments, permutations, combinations. Sections 4.2, 4.3 (to just before "How Long Does it Take to Sort?"), 4.4–5, 4.7 (up to just before "Distributing Distinguishable Objects". **Note: the section on "Distributing Distinguishable Objects" is erroneous, and the "Comparison of Counting Problems" may be confusing.**
11. (2/20) Binomial coefficients, generalized permutations, pigeonhole principle, combining counting rules. Sections 4.5–6, 4.8, and some supplementation.
12. (2/25) Counting overflow, including relationship to sorting. Section 4.3. Review for Exam I.
13. (2/27) Exam I. Material covered through 2/13.
14. (3/11) Basic probability. Sections 4.9, 4.12–13 (with some supplementation).
15. (3/13) Linked lists, stacks, queues, binary search. Sections 6.2–8.
16. (3/18) Hashing. Section 7.6.
17. (3/20) Trees, especially binary. Binary tree traversals. Sections 5.2and 5.6. Sections 5.3 and 5.4 are optional (containing the book's first discussion of tree traversals but for general trees).
18. (3/25) Binary search trees. Section 5.7–8.
19. (3/27) Longest common subsequence, dynamic programming. Section 6.9.
20. (4/3) Exam II. Material covered 2/18 through 3/18.
21. (4/8) Graph representations, minimum spanning tree. Sections 9.2–3, 9.5 (without "Running Time of Kruskal's Algorithm"), and some supplementation.
22. (4/10) Depth-first search and applications. Sections 9.6–7.
23. (4/15) Breadth-first search and Dijkstra's algorithm. Section 9.8 (until just before "Data Structures for Dijkstra's Algorithm") and some supplementation.
24. (4/17) Some graph theory. Section 9.10 and some supplementation.
25. (4/22) Gates, circuits, logical expressions, Boolean algebras. Sections 13.2–4 and some supplementation.
26. (4/24) Finite state machines. Section 10.2 and a little supplementation. Possibly also regular expressions in Section 10.5.