COMP 264: Introduction to Computer Systems (Section 001)
Fall 2015 Course Information & Syllabus

Instructor: R. I. Greenberg

Mailing Address: Computer Science Department
Loyola University
Water TowerCampus, Lewis Towers 520
820 N. Michigan Ave.
Chicago, Illinois 60611-2147

Office Hours: In Cudahy Science 408: Mon./Wed./Fri. 12:30–1:30pm, Mon./Wed. 2:50–3:50.
These are the guaranteed times to find me except as announced in advance or at `http://bit.ly/RIGcal`. You should also feel free to look for me at other times or make appointments.

Phone: (773)508-3782 on Tuesday/Wednesday/Thursday
Email: rig@cs.luc.edu **Home page:** http://rig.cs.luc.edu/~rig
Departmental tutoring schedule: See `http://www.luc.edu/cs/schedules/tutoringhours`.

Lectures: Mon./Wed./Fri. 10:25–11:15am in Crown 103.
Sometimes lecture notes or a summary may be available on the web. Other than that, 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.

Course Objectives: This course is designed to provide students with an understanding of the hierarchy of abstractions and implementations that comprise a modern computer system. The course is particularly geared towards topics of interest to a programmer, i.e., topics that affect the performance, correctness, or utility of user-level programs. Since this investigation is best carried out using the C programming language, the course will include some instruction in C for programmers familiar with Java.

Prerequisites: COMP 170. (It is also helpful to have taken COMP 150 and/or 163.)

Textbook: Randal E. Bryant and David R. O'Halloran. *Computer Systems: A Programmer's Perspective.* Pearson Prentice Hall, 3rd edition, 2015.

Course Requirements: There will be homework, two tests, and a final exam. The weightings within the semester grade will be: Homework: 25%, Tests 1–2: 20% each, 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 discussed in the same class on occasion. Homework will generally be submitted through a specified online mechanism. Anything that does get submitted on paper would need to go under the door of my office before class.

Exams: The midterm exams, tentatively scheduled for week 6 and week 12, are 50 minutes long. The final exam is scheduled for 9:00–11:00 Monday, December 7.

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 or online, cite your source completely and write up the solution in your own words.

Tentative Course Outline and Approximate Schedule:

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

1. (8/24) Administrivia, computer representation of information (1.1).
   Program translation (1.2–3).
2. (8/31) Basic overview of computer organization, operating system, networks, etc. (1.4–10).
3. (9/9) Information representation, boolean algebra, bit and logical operations (2.1).
4. (9/14) Integer representation (2.2).
5. (9/21) Floating point representation (2.4).
   Exam review.
6. (9/28) Exam I (9/28) on Chapters 1–2.
   Integer arithmetic (2.3). Floating point operations, and rounding (2.4–5).
7. (10/7) Machine-level program representation: historical perspective and program encodings (3.1–2).
   Data manipulation (3.3–5).
8. (10/12) Control (3.6).
9. (10/19) Procedures (3.7).
   Arrays (3.8).
   Heterogeneous data structures (3.9).
10. (10/26) Buffer overflow (3.10.3).
    Pointers, GDB, and possibly more (3.10,15).
    Floating-Point Code 3.11.
11. (11/2) Optimizing program performance: Intro (5.1–6).
12. (11/9) Catchup, review.
    Exam II (11/11) on Chapter 3.
    Instruction scheduling and pipelining (5.7).
13. (11/16) More on performance optimization (5.8–15).
    Storage technologies (6.1).
14. (11/23) Locality, memory hierarchy, cache memories, cache-friendly code (6.2–5).
15. (12/1) Virtual memory (9.1–6) and/or catchup/review.

Essential and Important IDEA Objectives:

In accordance with the IDEA course evaluation system, the following are the most important learning objectives:

Essential:

- Gaining factual knowledge (terminology, classifications, methods, trends)

Important:

- Learning to apply course material (to improve thinking, problem solving, and decisions)
- Learning fundamental principles, generalizations, or theories