# COMP 462: Advanced Computer Architecture (Section 806)
# Spring 2004 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:** MWF 11:30 am – 12:20 pm in DH-735.
Sometimes lecture notes or a summary will 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.

**Office Hours:** In Damen 329C: 10:15–11:15 on Monday, Wednesday, and Friday.
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:** Presents key principles underlying the design of digital computers. Includes an advanced presentation of principles for uniprocessors, with particular emphasis on pipelining and memory-hierarchy design. Will also touch upon multiprocessor design.

**Prerequisites:** COMP 260 or comparable background. Anticipated background includes: (1) Understanding of basic computer organization, including familiarity with such components as CPU, ALU, multiplexors, registers, main memory, caches, and buses, (2) Familiarity with the roles of compilers, assemblers, and operating systems, (3) some familiarity with assembly language, (4) familiarity with the representation of numbers in digital computers, and, possibly, (5) capability of writing and running simple C programs on UNIX systems.

**Required Text:** David A. Patterson and John L. Hennessy. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, third edition, 2002.

**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 20%, Exam II 25%, 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 week 5 and week 10, are each in two 45 minute parts. The final exam is scheduled for 8:30–10:30 am on Wednesday, April 21.

**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 on a weekly basis. (When selected sections or subsections are listed, it is assumed that you will include the introduction of the corresponding chapter or section.)

1. (1/5) Administrivia, introduction, computing trends, hardware costs. Sections 1.1–4.
   Computer performance. Amdahl's Law. CPU performance equation. Sections 1.5–6. Memory hierarchy basics as an example for performance calculations. (Not in text in this form.) (Sections 1.7–8 also provide some good perspective not covered in lecture.)

2. (1/12) Fallacies and Pitfalls. Section 1.9. (Sections 1.10–11 also provide some good perspective not covered in lecture, as is generally true of the chapter concluding sections we will not get to.) Instruction sets: Introduction and addressing. Sections 2.1–4.
   Instruction sets: Operands, operations, encoding. Sections 2.5–10.

3. (1/21) MIPS architecture. Section 2.12.
   The role of compilers. Section 2.11

4. (1/26) Pipelining. Introduction, a simple MIPS implementation and pipelining, basics of pipeline performance. Sections A.1 and A.3 (except "Implementing the Control for the MIPS Pipeline").
   Pipeline hazards: structural, data, and control. Sect. A.2 (except "Performance of Branch Schemes").

5. (2/2) Performance of branch schemes. Section A.2. Implementing the control for the MIPS pipeline, improved branch handling in MIPS and pipelining difficulties such as exceptions. Sections A.3–4.
   Exam I on Chapters 1–2.

6. (2/9) Pipelining: multicycle operations. Section A.5
   Dynamically scheduled pipelines. Section A.8. (Additional perspective not directly covered in lecture may be found in Sections A.9–11.)

7. (2/16) Dependences, hazards, and dynamic scheduling. Sections 3.1–2.
   Dynamic scheduling, branch prediction, etc. Sections 3.3–5.

8. (2/23) Scheduling and loop unrolling, static branch prediction, VLIW. Sections 4.1–3.
   Advanced compiler support for ILP. Section 4.4

9. (3/8) Memory-hierarchy: Introduction, cache basics, cache performance. Sections 5.1–3.
   Memory-hierarchy: Reducing miss penalty and miss rate. Sections 5.4–5. Memory-hierarchy: Reducing miss penalty or miss rate via parallelism, reducing hit time. Section 5.5–6.

10. (3/15) Exam II on Appendix A and Chapters 3–4.
    Memory-hierarchy: Main memory, memory technology, virtual memory basics. Sections 5.8–10.

11. (3/22) Storage systems: devices, buses. Sections 7.1–3.
    Storage systems: RAID, I/O performance. Sections 7.4–5, 7–8.

12. (3/29) Multiprocessors: Introduction. Sections 6.1. Multiprocessors: Centralized shared-memory architectures and distributed shared-memory architectures. Sections 6.3 and 6.5.
    Synchronization. Section 6.7.

13. (4/5) Interconnection networks. Sections 8.1, 4, and 5.

14. (4/14) Review for final exam. Perhaps do Vector Processors on 4/7 or 4/14: Sections G.1–2.