CS/ECE 347, Spring, 1996
Course Information

1. Organization

Instructors:

Randal E. Bryant
7128 Wean Hall
x8-8821
Randy.Bryant@cs.cmu.edu
M 9:30–10:30

David R. O'Hallaron
8125 Wean Hall
x8-8199
droh@cs.cmu.edu
F 1:30–2:30

Teaching Assistants:

Scott Davies
5103 Wean Hall
x8-7571
scottd@cs.cmu.edu
T 2:00–3:00

Ralph Melton
7120 Wean Hall
x8-7673
ralph@cs.cmu.edu
W 1:00–2:00

J. Alan Miller
3130 Wean Hall
jmbd@andrew.cmu.edu
Th 10:30–11:30

Secretary:

Joan Maddamma
7121 Wean Hall
x8-7656
Joan.Maddamma@cs.cmu.edu

Lecture:
TTh 3:00-4:20, Porter Hall 100

Recitations:

A	M	10:30	Scaife 324	Miller
B	M	12:30	Wean 5205	Melton
C	M	3:30	Baker 235a	Davies

2. Objectives

CS 347 presents the principles, characteristics, and trends of computer systems design at a level appropriate for all computer scientists and computer engineers. It expands on the role of a traditional computer architecture course, focussing largely on the hardware design, to include aspects of the complete system, comprising the hardware, operating system, compilers, and application software.

Our intention is to present material that will benefit all computer scientists, even those who are more concerned with software development than with hardware design. By learning more about what is inside "the box," software developers can make more effective use of computers, often achieving dramatic performance gains.

Students will also gain an understanding of the hardware technology that has fueled the rapid progress of computer systems. Since the technology will continue to change and improve, it is important to see how these changes will affect the characteristics of future computer systems.

After completing the course, students should have gained a firm grounding in the following:

Machine-level programming. With the advent of optimizing compilers, there is seldom need to write large amounts of assembly code. On the other hand, it is important to be able to understand the code generated by these compilers, and to write small sections of code. We will use the MIPS instruction set as our main case study, mostly using assembly code generated by the Gnu C compiler GCC.

Instruction Set Design. In addition to MIPS, we will study the general characteristics of other instruction sets, including the Intel x86 and the IBM PowerPC.

Pipeline Processor Implementation. Modern processors are designed to execute instructions in a pipelined, and possibly superscalar fashion. The characteristics of these pipelines and the code scheduling capabilities of an optimizing compiler can greatly affect the performance the processor achieves.

Memory Subsystem. The combination of primary memory, caches, and virtual memory are intended to provide the illusion of high speed access to a large memory space. Understanding how this subsystem operates is critical to efficiently implementing applications with high memory requirements.

I/O and Storage Subsystem. With the advent of multimedia applications, we are seeing stringent, real-time requirements placed on the various I/O and display devices, as well as the busses connecting them to the CPU. Understanding the organization and performance characteristics of this subsystem is therefore critical to current and future software developers.

Parallel Computing. Increasing application performance by using resources operating in parallel has been attempted in a number of different ways for nearly 50 years. Although the track record for these efforts is mixed, there are many indications that parallelism will play a increasingly critical role as the technology and the techniques for exploiting it mature.

3. Textbook

J. L. Hennessy, and D. Patterson, *Computer Organization and Design : The Hardware Software Interface*, Morgan Kaufmann, 1994.

4. Course Requirements

Your participation in the course will involve five forms of activity:

1. Attending the lectures.
2. Participating in the recitations and completing the recitation assignments.

3. Laboratory assignments.
4. Reading the text and supplementary handouts.
5. Exams

Attendance will not be taken at the lectures, but you will be considered responsible for all material presented, much of which is not covered to the same depth in the textbook.

The recitations will play a critical role in the course, both for reinforcing the material presented in the lectures, as well as to give directions and assistance on the laboratory assignments. To promote active learning, there will be a recitation assignment almost every week. These assignments will require applying some of the ideas presented during the lectures, and they can be turned in only in conjunction with attending the recitation. Your course grade will depend partially on these assignments, as well as your TA's evaluation of your participation in the recitations.

The laboratory assignments will be multi-week efforts providing in-depth understanding of some aspect of computer system design. They will involve some combination of C and assembly programming, as well as performance measurements.

The textbook covers much of the material taught in the course. In the schedule shown in this document, you will see the reading assignments associated with each section of the course. It would be a good idea to read over each of these assignments twice—a brief reading before the lecture to become familiar with the basic concepts, and a more thorough reading later to understand the details.

5. Policies

5.1. Handing in Assignments

Recitation assignments are due at your recitation section. You must submit a typewritten (or computer-generated) document at the end of the class, and you must attend the recitation to receive credit. You should do these assignments entirely on your own.

Laboratory assignments are due at 3:00 pm on the specified due date. Late assignments will not be accepted. It is your responsibility to get your assignments done on time. Be sure to work far enough in advance to avoid unexpected problems, such as illness, unreliable or overloaded computer systems, etc. Each assignment will clearly state whether you are to work on it individually or collaboratively.

5.2. Final Grade Assignment

Each student will receive a numeric score for the course, based on a weighted average of the following:

Recitation Assignments. Of the 12 assignment scores, we will drop the lowest 2 and count the remaining for 15% (1.5% each) of the total.

Recitation Participation. Your recitation TA will assign you a grade counting 5% of your score based on the consistency and quality of your participation in the recitations.

Laboratory Assignments. There will be three labs counting a total of 40% of your score. The exact weighting of the different labs will be determined near the end of the course based on our perception of the relative effort required. In any case, each lab will count 10–15% of your score.

Exams. There will be two in-class exams, each counting 10%, plus a final counting 20%.

Grades for the course will be determined by a curve. The total score will be plotted as a histogram, and then approximate cutoff points for the different letter grades will be determined. Individual cases, especially those near the cutoff points may be adjusted upward or downward based on factors such as attendance, class participation, and special circumstances.

5.3. Cheating

For individual assignments, and exams, we will enforce a strict policy with respect to cheating. An individual assignment must be the sole work of the student turning it in. These assignments will be closely monitored, and students may be asked to explain any suspicious similarities. For some of your laboratories, you will be working in small teams. In such cases, you are naturally allowed to collaborate with your team members, but not with others. The following are guidelines on what collaboration is authorized and what is not.

Not Cheating:

- Clarifying ambiguities or vague points in class handouts or textbooks.
- Helping another student use the computer systems, networks, compilers, or other system facilities.

Cheating:

- *Sharing code or other electronic files*: either by copying files, retyping another person's files, looking at a copy of someone else's code, or by allowing someone else to copy or look at your code.
- *Sharing written assignments*: Looking at or copying someone else's assignment, or letting others look at yours.

Be sure to store your work in protected directories.

The penalty for cheating will depend on the severity of the offense and the student's past record in this regard. *At the very least* the student will be given a score of 0 for the assignment.

6. Class Schedule

Warning: the following schedule is subject to change. Changes will be announced on the class news group. An updated schedule will be maintained on the class WWW pages.

CS 347, Spring '96 **Updated January 15, 1996**

Class	Date	Day	Topic	Reading	Assignments
1	01/16	Tu	Overview, Technology	1	
2	01/18	Th	Data formats & referencing	3.1–4, 3.7, A.1, A.10	
3	01/23	Tu	Conditional & Loops	3.5, A.10	
4	01/25	Th	Procedures	3.6, 3.9–10, A.6	L1 Out
5	01/30	Tu	Performance & Benchmarking	2	
6	02/01	Th	Benchmarking		
7	02/06	Tu	Floating Point Instructions		
8	02/08	Th	Data Structures	3.11, A.2–5	
9	02/13	Tu	Other Instruction Sets	3.8, 3.12–14, App. E	
10	02/15	Th			L1 Due
11	02/20	Tu	Exam #1		
12	02/22	Th	Building Blocks	5	L2 Out
13	02/27	Tu	Basic Pipelining	6.1–6.3	
14	02/29	Th	Data Hazards	6.4–6.6	
15	03/05	Tu	Control Hazards	6.7	
16	03/07	Th	Exception Handling	6.8–6.12	
17	03/12	Tu	Fixed Point Arithmetic	4.1–4.7	
18	03/14	Th	Floating Point Arith.	4.8–4.11	L2 Due
19	03/19	Tu	Advanced Processors		
20	03/21	Th	Exam #2		
	03/26	Tu	Spring Break		
	03/28	Th			
21	04/02	Tu	Caches	7.1–7.2	L3 Out
22	04/04	Th			
23	04/09	Tu	I/O, Storage	8	
24	04/11	Th			
25	04/16	Tu	Virtual Memory	7.3	
26	04/18	Th			
27	04/23	Tu	Memory System Summary	7.4–7.7	
28	04/25	Th	Parallel Computers	9	
29	04/30	Tu			L3 Due
30	05/02	Th			

7. Facilities

7.1. MIPS Processors

The various DECStations around campus are based on the MIPS architecture. These machines are fast disappearing, but there are still some in the Wean Hall cluster, and the server machines (`unix1.andrew`, etc.) are MIPS machines.

7.2. Electronic Communication

We will use the World-Wide Web (WWW) as the central repository for all information about the class. You can reach this page via the link labeled "course, lecture, and seminar home pages" from the CS home page. The complete URL is:

```
http://www.cs.cmu.edu/afs/cs/academic/class/15347-s96/www/home.html
```

Using the Web, you can:

- Obtain copies of any handouts or assignments. This is especially useful if you miss class or you lose your copy.
- Read clarifications and changes made to any assignments, schedules, or policies.
- Find links to any electronic data you need for your assignments

For urgent communication with the teaching staff, it is best to send electronic mail (preferred) or to phone. We have also set up a news group for this class, `cmu.cs.class.cs347`. This group will be used by members of the teaching staff to post announcements and clarifications. You may also post to this group to make queries.