CS 213: Introduction to Computer Systems

Randal E. Bryant and David R. O'Hallaron

School of Computer Science
Carnegie Mellon University

Fall 1998

1. Organization

Instructors:

Randal E. Bryant	David R. O'Hallaron
WeH 7128	WeH 8125
x8-8821	x8-8199
Randy.Bryant@cs.cmu.edu	droh@cs.cmu.edu
Tue 10:30–11:30am	Tue 10:30–11:30am

TAs:

Chris Colohan	Larry Greenfield	Kip Walker
WeH 5101	WeH 3130	WeH 8218
x8-8139	x8-8735	x8-7555
colohan+@cs.cmu.edu	leg+@andrew.cmu.edu	kwalker@cs.cmu.edu
Wed, 3:00–4:00pm	Wed, 12:00-1:00pm	Wed, 2:00–3:00pm

Class Secretary:
Joan Maddamma
WeH 7121
x8-7656
jfm@cs.cmu.edu

Lecture:
Tue Thu 9:00-10:20, Wean Hall 7500

Recitations:

A	Mon	10:30–11:20	Student Center 203	Chris Colohan
B	Mon	11:30–12:20	Student Center 203	Chris Colohan
C	Mon	12:30–1:20	Student Center 203	Larry Greenfield
D	Mon	1:30–2:20	Student Center 203	Kip Walker
E	Mon	2:30-3:20	Student Center 203	Kip Walker

Web page: `www.cs.cmu.edu/afs/cs/academic/class/15213-f98/www/`
Newsgroup: `cmu.cs.class.cs213`

2. Objectives

Our aim in CS 213 is to help you become a better programmer by teaching you the basic concepts underlying all computer systems. We want you to learn what really happens when your programs run, so that when things go wrong (as they always do) you will have the intellectual tools to solve the problem.

Why do you need to understand computer systems if you do all of your programming in high level languages? In most of computer science, we're pushed to make abstractions and stay within their frameworks. But, any abstraction ignores effects that can become critical. As an analogy, Newtonian mechanics ignores relativistic effects. The Newtonian abstraction is completely appropriate for bodies moving at less than 0.1c, but higher speeds require working at a greater level of detail.

Our 21X sequence works as follows: 211 is based on a simplified model of program execution. 212 builds further layers of abstraction. 213 introduces greater detail about system behavior and operation. This greater detail is needed for optimizing program performance, for working within the finite memory and word size constraints of computers, and for systems-level programming.

The following "realities" are some of the major areas where the abstractions we teach in 211/212 break down:

1. *Int's are not integers, Float's are not reals.* Our finite representations of numbers have significant limitations, and because of these limitations we sometimes have to think in terms of bit-level representations.

2. *You've got to know assembly language.* Even if you never write programs in assembly, The behavior of a program cannot be understood sometimes purely based on the abstraction of a high-level language. Further, understanding the effects of bugs requires familiarity with the machine-level model.

3. *Memory matters.* Computer memory is not unbounded. It must be allocated and managed. Memory referencing errors are especially pernicious. An erroneous updating of one object can cause a change in some logically unrelated object. Also, the combination of caching and virtual memory provides the functionality of a uniform unbounded address space, but not the performance.

4. *There is more to performance than asymptotic complexity.* Constant factors also matter. There are systematic ways to evaluate and improve program performance

5. *Computers do more than execute instructions.* They also need to get data in and out and they interact with other systems over networks.

By the end of the course you will understand these "realities" in some detail. As a result, you will be prepared to take any of the upper level systems classes at Carnegie Mellon. Even more important, you will have learned skills and knowledge that will help you throughout your career.

Good luck!

3. Textbook

This is a totally new course for which no complete textbook exists In fact, your instructors will probably end up having to write it! In the interim, the following will serve as the text for the course:

> Brian W. Kernighan and Dennis M. Ritchie, *The C Programming Language, Second Edition*, Prentice Hall, 1988.

This the classic *K & R* book, the standard against which all reference manuals are compared. Although it only provides partial coverage of the course material, it is an essential part of every computer scientist's library.

4. Course Organization

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

1. Attending the lectures.
2. Participating in the recitations.
3. Homework and laboratory assignments.
4. Reading the text and supplementary handouts.
5. Exams

Attendance will not be taken at the lectures or recitation sections. You will be considered responsible for all material presented at the lectures and recitations. Lectures will cover higher-level concepts. Recitations will be more applied, covering important "how-to's", especially in using tools that will help you do the labs. In addition, the recitations will help clarify lecture topics and describe exam coverage.

There will be two types of assignments in this class. *Lab* assignments will be multi-week efforts providing in-depth understanding of some aspect of computer systems. Labs will involve some combination of C and assembly programming, and some will involve performance measurements. *Homework* assignments will 1-week efforts that involve solving a series of smaller problems. The solution to many of these will involve writing small C programs.

5. Getting help

For urgent communication with the teaching staff, it is best to send electronic mail (preferred) or to phone.

If you want to talk to a staff member in person, remember that our posted office hours are merely times when we guarantee that we will be in our offices. You are always welcome to visit us outside of office hours if you need help or want to talk about the course. However, we ask that you follow a few simple guidelines:

- Prof. Bryant and Prof. O'Hallaron normally work with their office doors open and welcome visits from students whenever their doors are open. However, if their doors are closed, they are busy with a meeting or a phone call and should not be disturbed.
- The TAs share offices with other students. To avoid disturbing these students, please send mail or zephyr before visiting a TA outside of office hours so they can arrange to meet you.

We will use the Web as the central repository for all information about the class. The class home page is at

`www.cs.cmu.edu/afs/cs/academic/class/15213-f98/www/`

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

We have also set up a news group for this class, `cmu.cs.class.cs213`. 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.

6. Policies

Working in Groups

For both homeworks and labs, you may work in groups of up to 2 people. It is up to you to form and regulate your own groups. If you are not happy with your partner, you are free to find another partner. You may also work by yourself.

Handing in Assignments

All assignments are due at 12:01am (one minute past midnight) on the specified due date. All handins are electronic, usually consisting of one or more files that are to be copied to a specified directory. The writeup for an assignments will provide about the handin procedure for that assignment.

Penalties for Late Assignments

Late assignments will be docked 20% each day for the first two days. Assignments more than 2 days late will not be accepted, unless you have arranged for an extension in advance with your instructors. For example, suppose an assignment is due at 12:01am on Thursday. If you hand it in between 12:02am Thursday and 12:01am Friday, you will be docked 20%. If you turn it in between 12:02am Friday and 12:01am Saturday, you will be docked 40%. You won't be able to turn it in at all after 12:01am Saturday.

Making up Exams and Assignments

Missed exams and assignments more than 2 days late can be made up, but only if you make prior arrangements with Prof. O'Hallaron. However you should have a good reason for doing so. 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.

Appealing Grades

After each exam, homework, and assignment is graded, Prof. O'Hallaron will send each of you a personalized email with your grade (as well as all of your previous grades). You have seven calendar days from the date he sends the email to appeal your grade.

If you have questions about the grade you received on an assignment (homework or lab), please talk first to the person in charge of the assignment, who will be clearly identified in the writeup. If you are still not satisfied, please come and visit Prof. O'Hallaron.

If you have questions about an exam grade, please visit Prof. O'Hallaron directly.

Final Grade Assignment

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

- **Assignments:** The assignments will count a combined total of 50% of your score. The exact weighting of the different assignments will be determined near the end of the course based on our perception of the relative effort required. In any case, each homework will count 3–5%, while each lab will count 8–12% of your score. Since small differences in scores can make the difference between two letter grades, you'll want to make a serious effort on each assignment.

- **Exams:** There will be two in-class exams, each counting 12.5%, plus a final counting 25%.

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, improvement throught the course, final exam performance, and special circumstances.

Cheating

All of your assignments allow collaboration, but only with the other member of your project group. Each assignment must be the sole work of the group turning it in. Assignments will be closely monitored, and students may be asked to explain any suspicious similarities. The following are guidelines on what collaboration outside of your group is authorized and what is not:

What is Cheating?

- *Sharing code or other electronic files*: either by copying, retyping, looking at or supplying a copy of a file.

- *Sharing written assignments*: Looking at, copying, or supplying a assignment.

What is NOT Cheating?

- Clarifying ambiguities or vague points in class handouts or textbooks.

- Helping others use the computer systems, networks, compilers, debuggers, profilers, or other system facilities.

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.

7. Facilities

We will use a cluster of Alphas for the labs and assignments. The class Web page has details.

8. Class Schedule

Figure 1 shows the tentative schedule for the class. The notation "Hi" indicates a homework assignment, while "Li" indicates a lab. Any changes will be announced on the class news group. An updated schedule will be maintained on the class web page.

Class	Date	Day	Topic	Reading	Asst	Lecturer
1	08/25	Tue	Overview			Both
2	08/27	Thu	Bits and bit operations	2.9	H1 Out	REB
3	09/01	Tue	Integer Representations	2.2–2.4,2.7 B11		REB
4	09/03	Thu	Integer Arithmetic	2.5–2.6, 2.12		REB
5	09/08	Tue	Machine Model			REB
6	09/10	Thu	Alpha ISA basics	3.2–3.7	H1 Due, H2 Out	REB
7	09/15	Tue	Alpha procedures			REB
8	09/17	Thu	Structured Data	5.1–5.12	H2 Due, L1 Out	REB
9	09/22	Tue	Heterogeneous Data	6.1–6.8		REB
10	09/24	Thu	IEEE Floating Point Format			REB
11	09/29	Tue	Object Code and Processes	B8, B9		DROH
12	10/01	Thu	Asynchronous processing		L1 Due, H3 Out	DROH
13	10/06	Tue	Exam #1			
14	10/08	Thu	Memory management	8.7	H3 Due, L2 Out	DROH
15	10/13	Tue	Memory referencing errors			DROH
16	10/15	Thu	Garbage Collection			DROH
17	10/20	Tue	Memory technology			REB
18	10/22	Thu	Caches		L2 Due, H4 Out	REB
19	10/27	Tue	Virtual Memory			REB
20	10/29	Thu	Memory system performance		H4 Due, L3 Out	DROH
21	11/03	Tue	Code optimization			DROH
22	11/05	Thu	Performance Evaluation			DROH
23	11/10	Tue	Benchmarking			DROH
24	11/12	Thu	Networking technology		L3 Due	DROH
25	11/17	Tue	Exam #2		L4 Out	
26	11/19	Thu	TCP/IP			DROH
27	11/24	Tue	Network routing			DROH
	11/26	Thu	Thanksgiving			
28	12/01	Tue	Application: BDDs			REB
29	12/03	Thu	Application: Quake modeling		L4 Due	DROH

Figure 1: Tentative Class Schedule