CS 213: Introduction to Computer Systems

Randal E. Bryant(*) and David R. O'Hallaron(**)

(*) School of Computer Science
(**) School of Computer Science and Dept of Electrical and Computer Engineering
Carnegie Mellon University
Fall 2002

1 Organization

Instructors:

Randal E. Bryant	David R. O'Hallaron
WeH 4220	WeH 8125
x8-8821	x8-8199
Randy.Bryant@cs.cmu.edu	droh@cs.cmu.edu
Wed 10:00–11:00am	Tue 10:30–11:30am

TAs:

Rajesh Balan	Shimin Chen	Andrew Faulring	Anubhav Gupta	Annie Luo
Section E	Section C	Section A	Section D	Section B
WeH 8205	WeH 8019	NSH 2504	WeH 8218	WeH 8402
x8-3778	x8-5143	x8-4074	x8-7555	x8-3076
rajesh@cs	chensm@cs	faulring@cs	anubhav@cs	luluo@cs
Thu 4-5pm	Wed 3-4pm	Thu 5-6pm	Thu 3-4pm	Thu 6-7pm

Class Assistant:
Rosemary Battenfelder
WeH 4218
x8-3853
rosemary@cs.cmu.edu

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

Recitations:

A	Mon	10:30–11:20	OSC (Old Student Center) 200	Andrew Faulring
B	Mon	11:30–12:20	OSC (Old Student Center) 200	Annie Luo
C	Mon	12:30–1:20	OSC (Old Student Center) 200	Shimin Chen
D	Mon	1:30–2:20	OSC (Old Student Center) 200	Anubhav Gupta
E	Mon	2:30–3:20	OSC (Old Student Center) 200	Rajesh Balan

Web page: `www.cs.cmu.edu/afs/cs/academic/class/15213-f02/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.

Oversimplifying matters somewhat, 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 (both CS and ECE). Even more important, you will have learned skills and knowledge that will help you throughout your career.

3 Textbook

The primary textbook for the course is

> Randal E. Bryant and David R. O'Hallaron, *Computer Systems: A Programmer's Perspective*, Prentice Hall, 2003.

In addition, we require you to have the following reference book on the C programming language:

> 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. 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. Preparing for and participating in the recitations.
3. Laboratory assignments.
4. Reading the text.
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.

The textbook contains both *practice problems* within the chapter text and *homework problems* at the end of each chapter. The intention is that you work on the practice problems right as you are reading the book. The answers to these problems are at the end of each chapter. Our experience has been that trying out the

concepts on simple examples helps make the ideas more concrete. In addition, the schedule (at the end of this document and on the class web page) shows specific homework problems with each lecture topic. The intention is that you try these out and discuss them in the next recitation. You will find that you will get much more out of recitation if you have done some advance preparation.

The only graded assignments in this class will be a set of 7 labs. Some of these are fairly short, requiring just one week, while others are more ambitious, requiring several weeks. Most labs will be handed out in class on Thursday and due either one or two weeks later on a Wednesday.

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-f02/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 Alone on Assignments

Unlike previous terms, you will work on the first six laboratory assignments by yourself. However, you will work on the last assignment (writing a web proxy) with a partner, since the grading will be done by a

personal demo to one of the teaching staff.

Handing in Assignments

All assignments are due at 11:59pm (one minute before 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 details 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 Prof. O'Hallaron. For example, suppose an assignment is due at 11:59pm on Wed. If you hand it in between midnight and 11:59pm Thursday, you will be docked 20%. If you turn it in between midnight and 11:59pm Friday, you will be docked 40%. You won't be able to turn it in at all after 11:59pm Friday.

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 60% 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 lab will count 6–12% 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 10%, plus a final counting 20%.

Grades for the course will be determined by a method that combines both curving and absolute standards. The total score will be plotted as a histogram. Cutoff points are determined by examining the quality of work by students on the borderlines. Individual cases, especially those near the cutoff points may be adjusted upward or downward based on factors such as attendance, class participation, improvement throughout the course, final exam performance, and special circumstances.

Cheating

Each lab assignment must be the sole work of the student turning it in. Assignments will be closely monitored by automatic cheat checkers, and students may be asked to explain any suspicious similarities. The following are guidelines on what collaboration 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 an 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.
- Helping others with high-level design issues.
- Helping others debug their code.

Be sure to store your work in protected directories.

The usual penalty for cheating is to be removed from the course with a failing grade. We also place a record of the incident in the student's permanent record.

7 Facilities: Intel Computer Systems Cluster

Intel has generously donated a cluster of 25 Linux-based Pentium III servers, specifically for CS 213, that we will use for all labs and assignments. The class Web page has details.

8 Class Schedule

Figure 1 shows the tentative schedule for the class. The reading assignments are all in the new book. You can check on the class web page for suggested readings in K&R. The schedule also indicates suggested homework problems, the lab activities, and the lecturer for each class.

Any changes will be announced on the class news group. An updated schedule will be maintained on the class web page.

Class	Class	Day	Topic	Reading	Problems	Labs	Lecturer
1	08/27	Tue	Overview	1			Both
2	08/29	Thu	Bits and Bytes	2.1	2.44, 2.45,	L1 Out	REB
3	09/03	Tue	Integers	2.2–2.3	2.49, 2.54		REB
4	09/05	Thu	Floating Point	2.4–2.5	2.59, 2.60, 2.61		REB
5	09/10	Tue	Machine Prog I - Overview	3.1–3.5	3.31		REB
6	09/12	Thu	Machine Prog II - Control	3.6	3.34	L1 Due, L2 Out	REB
7	09/17	Tue	Machine Prog III- Procs	3.7			REB
8	09/19	Thu	Machine Prog IV- Data	3.8–3.11	3.36		DROH
9	09/24	Tue	Mach. Prog V - Advanced	3.12–3.13, 3.16	3.24		REB
10	09/26	Thu	Program Optimization I	5.1–5.6		L2 Due, L3 Out	REB
11	10/01	Tue	Program Optimization II	5.7–5.16	5.14, 5.17, 5.18		REB
12	10/03	Thu	Memory Hierarchy	6.1–6.4	6.21, 6.23, 6.24	L3 Due	DROH
13	10/08	Tue	Exam 1	6:00-7:30pm, DH 2315			
14	10/10	Thu	Cache Memories	6.5–6.8	6.25, 6.26, 6.27	L4 Out	DROH
15	10/15	Tue	Linking	7	7.8, 7.12		DROH
16	10/17	Thu	Except. Control Flow I	8.1–8.4	8.10, 8.11		REB
17	10/22	Tue	Except. Control Flow II	8.5–8.8	8.19		DROH
18	10/24	Thu	Time Measurement	9	9.10, 9.11	L4 Due, L5 Out	REB
19	10/29	Tue	Virtual Memory	10.1–10.6	10.11, 10.12, 10.13		REB
20	10/31	Thu	P6/Linux Memory System	10.7–10.8	10.14	L5 Due, L6 Out	REB
21	11/05	Tue	Dynamic Storage Alloc I	10.9	10.15, 10.16		DROH
22	11/07	Thu	Dynamic Storage Alloc II	10.10–10.13	10.18		DROH
23	11/12	Tue	Exam 2	6:00-7:30pm, DH 2315			
24	11/14	Thu	System-level I/O	11	11.7		DROH
25	11/19	Tue	Internetworking	12.1–12.3		L6 Due, L7 Out	DROH
26	11/21	Thu	Network Programming	12.4			DROH
27	11/26	Tue	Web Services	12.5–12.8	12.10		DROH
	11/28	Thu	Thanksgiving				
28	12/03	Tue	Concurrent Servers	13.1–13.3	13.22, 13.23		DROH
29	12/05	Thu	Thread Programming	13.4–13.8	13.24	L7 Due	DROH

Figure 1: CS 213 Class Schedule