

CS 213: Introduction to Computer Systems

Spring 2001

1. Organization

Instructors:

Guy E. Blelloch	Bruce Maggs
WeH 7125	WeH 4123
x8-6245	x8-7654
blelloch@cs.cmu.edu	bmm@cs.cmu.edu
Mon 2:30-3:30pm	Mon 3:30-4:30pm

TAs:

Umut Acar	Shaheen Gandhi	Urs Hengartner	David Koes
WeH 4130	WeH 3108	WeH 4103	WeH 3108
x8-3053	x2-3001	x8-7571	422-8325
umut@cs.cmu.edu	sgandhi+@andrew	uhengart@cs.cmu.edu	dkoes@andrew
Tue 3-4pm	Wed 1:30-2:30pm	Tue 10:30-11:30am	Tue 7-8pm

Class Secretary:

Keith Ledonne
WeH 7116
x8-7660
ledonne@cs.cmu.edu

Lecture:

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

Recitations:

A	Mon	10:30-11:20	OSC (Old Student Center) 203	Urs
B	Mon	11:30-12:20	OSC (Old Student Center) 203	Urs/Umut
C	Mon	12:30-1:20	OSC (Old Student Center) 203	Umut
D	Mon	1:30-2:20	OSC (Old Student Center) 203	Shaheen
E	Mon	2:30-3:20	OSC (Old Student Center) 203	David

Web page:

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

Professor's Randy Bryant and David O'Hallaron are in the process of writing a textbook for this course. We will be using a very early draft of this book, which will be available at the bookstore hopefully by the end of the first week of classes. In addition to this draft book, we will make all the overheads from class available online.

We will also use the following book as a reference text on C.

Samuel P. Harbison and Guy L. Steele Jr., *C: A Reference Manual, Fourth Edition*, Prentice Hall, 1995.

If you have some other book on C, it will probably suffice, but be warned that the Harbison and Steele book has much more precise definitions than many other books.

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 (in groups of up to 2) 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. Labs will be handed out in class on Thursday and due a couple of weeks later on a Wednesday.

Homework assignments will be 1-week efforts (that you work on individually) that involve solving a series of smaller problems. The solution to many of these will involve writing small C programs. Homeworks are designed to be drills to help you get practice for exam problems. Homeworks will be handed out at class on Tuesday and due the following Monday.

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. Blelloch and Prof. Maggs 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