15-213: Introduction to Computer Systems
Fall 2007
Syllabus

1 Course Details at a Glance

Lectures: Wednesdays and Fridays, 1:00-2:20pm, WeH 7500
Recitations: Mondays, WeH 5310
Section A: 10:30am-11:20am
Section B: 11:30am-12:20pm
Section C: 12:30pm-1:20pm
Section D: 1:30pm-2:20pm
Section E: 2:30pm-3:20pm

Instructors: Prof. Greg Ganger, CIC 2208, 268-1297, `ganger@ece.cmu.edu`
Office Hours: Thursdays, 4-5pm, CIC 2208

Prof. Todd C. Mowry, WeH 8105, 268-3725, `tcm@cs.cmu.edu`
Office Hours: Mondays, 4-5pm, WeH 8105

TAs: *Section A:* Matthew Wachs, `mwachs@cs.cmu.edu`
Office Hours: Wednesdays, 8:00-9:00pm, WeH 7220
Section B: Brett Simmers, `bsimmers@cmu.edu`
Office Hours: Tuesdays, 2:00-3:00pm, WeH 3108
Section C: Owen Yamauchi, `ody@andrew.cmu.edu`
Office Hours: Fridays, 3:30-4:30pm, WeH 3108
Section D: Austin McKinley, `amckinle@cmu.edu`
Office Hours: Fridays, 4:30-5:30pm, WeH 5205
Section E: Steven Okamoto, `syo@andrew.cmu.edu`
Office Hours: Thursdays, 2:00-3:00pm, DH 4302D

Class Admin: Jennifer Landefeld, WeH 8120, `jennsbl@cs.cmu.edu`

Web Page: `http://www.cs.cmu.edu/~213`
Message Board: `http://autolab.cs.cmu.edu`
Note: This is the only message board your instructors will be monitoring. We will not be using the Andrew or Blackboard message boards for this class.
Staff Email: `15-213-staff@cs.cmu.edu`
Handouts: *Electronic:* `/afs/cs.cmu.edu/academic/class/15213-f07/public`
Hardcopies: In bins outside WeH 8120.

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 need 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 seven labs. Some of these are fairly short, requiring just one week, while others are more ambitious, requiring several weeks.

5 Getting Help

We will use the Web as the central repository for all information about the class. Using the class web page, you can:

- Obtain copies of any handouts or assignments. This is especially useful if you miss class or you lose your copy.

- Find links to any electronic data you need for your assignments.
- Read clarifications and changes made to any assignments, schedules, or policies.
- Post messages to make queries about the course, specific labs, or exams.

The lab assignments and class message board are offered through a Web service called *Autolab*. See the Autolab web page at `http://autolab.cs.cmu.edu` for more information.

For urgent communication with the teaching staff, please send email to `15-213-staff@cs.cmu.edu`.

6 Policies

6.1 Working Alone on Assignments

You will work on all assignments by yourself.

6.2 Handing in Assignments

All assignments are due at 11:59pm (one minute before midnight) on the specified due date. All handins are electronic using the Autolab system.

6.3 Handing in Late Assignments

Each assignment will have an *end date*, which will typically be three days after the due date. Because we post solutions on the end date, *no assignments will be accepted after the end date.* Prior to the end date, it is possible to hand in assignments late, subject to the following policies.

Lateness is rounded up to full days: If the assignment is due at 11:59pm on Friday and you hand it in at 2am on Saturday, that still counts as being one day late (not $\frac{1}{12}th$ of a day late).

Grace days: Each student will receive a budget of five *grace days* for the course, which work as follows. If you hand in an assignment k days late but you still have at least k grace days left in your budget, there will be no late penalty for your assignment, but you will have used up k grace days from your budget.

What happens when you run out of grace days: Once you use up your grace days, you can still hand in assignments late (before the end date), but there will be a 15% late penalty per day. For example, if you hand in an assignment three days late but only have two grace days left in your budget, you will use up the two remaining grace days and there will be a 15% late penalty. If you have zero grace days left in your budget and hand in your assignment two days late, there will be a 30% late penalty.

Grace days are a tool to help you manage your time and to help smooth out burstiness in assignment due dates. We strongly recommend that you try to conserve grace days for the end of the term, however, when things get most hectic.

6.4 Notification of Your Grades

After each exam and lab assignment is graded, you will receive personalized email with your grade (as well as all of your previous grades).

6.5 Requesting a Re-Grade for an Assignment or an Exam

We will make the utmost effort to be fair and consistent in our grading. However, if you believe that you did not receive appropriate credit for an assignment or an exam, you may request a re-grade as follows:

- Submit your request *in writing* within seven calendar days of when the grade notification is sent to you via email, explaining in detail why you think that there was a mistake in the grading. Please note that verbal requests will not be processed: they must be in writing.
- For assignments, these requests should be submitted to the lead staff member (either an instructor or a TA) for the assignment, as indicated on the assignment handout. For exams, please submit your request to one of the instructors (Prof. Ganger or Prof. Mowry).
- When you submit a request for a re-grade, the entire assignment or exam may be re-graded (not just the parts that you specify). Your grade may go up or down (or stay the same) as a result of the regrade request.

Your request will be processed off-line, and we will respond to your request as quickly as possible (typically within a week). This re-grade policy is designed to correct legitimate mistakes in grading, but to discourage frivolous regrade requests (for the sake of being fair and consistent across the entire class).

6.6 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 a mid-term exam and a final exam, which count for 15% and 25% respectively.

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 near 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, and final exam performance.

6.7 Cheating

Each lab assignment must be the sole work of the student turning it in. Assignments will be closely monitored by automatic cheat checkers, including comparing turned-in code to the work of students from the same and previous semesters, and students may be asked to explain any suspicious similarities. These cheat checkers are very effective, so please don't try your luck. 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.

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. (Note that "debug" does not mean providing substantial help in solving the assignment.)

Be sure to store your work in protected directories.

7 Facilities: Intel Computer Systems Cluster

Intel has generously donated a cluster of 15 Linux-based 64-bit Xeon servers, specifically for 15-213, that we will use for all labs and assignments. The class Web page has details.

8 Class Schedule

Table 1 shows the tentative schedule for the class. The reading assignments are all from the CS:APP book. The schedule also indicates suggested homework problems, the lab activities, and the lecturer for each class.

Any changes will be announced on the class message board. An updated schedule will be maintained on the class Web page.

Table 1: 15-213, Fall 2007 Schedule.

Class	Date	Day	Topic	Reading	Problems	Assignments	Who
1	8/29	Wed	Overview	1			Both
2	8/31	Fri	Bits, Bytes, and Integers	2.1–2.3	2.44, 2.45, 2.49, 2.54	L1 Out	GG
3	9/5	Wed	Floating Point	2.4–2.5	2.59, 2.60, 2.61		GG
4	9/7	Fri	Machine Prog 1: Overview	3.1–3.5	3.31		TCM
5	9/12	Wed	Machine Prog 2: Control	3.6–3.7	3.34	L1 Due, L2 Out	TCM
6	9/14	Fri	Machine Prog 3: Data	3.8–3.11	3.36		TCM
7	9/19	Wed	Machine Prog 4: Advanced	3.12–3.13, 3.16	3.24		TCM
8	9/21	Fri	Program Optimization	5.1–5.15	5.3, 5.6	L2 Due, L3 Out	TCM
9	9/26	Wed	Memory Hierarchy	6.1–6.3	6.2, 6.3, 6.4		TCM
10	9/28	Fri	Cache Memories	6.4	6.9–6.17		TCM
11	10/3	Wed	Cache Performance			L3 Due, L4 Out	TCM
12	10/5	Fri	Linking	7	7.2, 7.3		GG
13	10/10	Wed	Except. Control Flow 1	8.1–8.4	8.1–8.3		GG
14	10/12	Fri	Except. Control Flow 2	8.5–8.8	8.19	L4 Due	GG
15	10/17	Wed	**Exam #1**			L5 Out	TCM
	10/19	Fri	**Mid-Semester Break**				
16	10/24	Wed	Dyamic Storage Alloc. 1	10.9	10.6, 10.7		TCM
17	10/26	Fri	Dyamic Storage Alloc. 2	10.10–10.13	10.18	L5 Due, L6 Out	GG
18	10/31	Wed	System-Level I/O	11	11.2, 11.3		GG
19	11/2	Fri	Virtual Memory	10.1–10.6	10.4		TCM
20	11/7	Wed	P6/Linux Memory System	10.7–10.8	10.14		TCM
21	11/9	Fri	Internetworking	12.1–12.3		L6 Due	GG
22	11/14	Wed	Network Programming	12.4	12.5	L7 Out	GG
23	11/16	Fri	Web Services	12.5–12.7			GG
			Thanksgiving Break				
24	11/28	Wed	Concurrency	13.1–13.4	13.4–13.6		GG
25	11/30	Fri	Synchronization 1	13.5–13.8	13.7, 13.9, 13.10		GG
26	12/5	Wed	Synchronization 2				GG
27	12/7	Fri	Exam Review			L7 Due	TCM
			Final Exam				