COSC 161—COMPUTER SYSTEMS I
COURSE INFORMATION

Revision 1[A]

You are responsible for knowing the full contents of this document!

1 The topics

This course is the first of a two-course series (the second of which is *Computer Systems II*), although you need not take the second course to get a complete experience from this first one. These courses explore how computer systems are structured, building up a collection of hardware and software layers that make a complete, general-purpose computing device.

For most computer users, a *computer system* is a mysterious box of “hardware”, plugged into a mysterious “network”, on which they use mysterious “software”. For many purposes, these users need to know little about the box, the network, and the software. They don’t need to understand how these components interact, nor how they work internally; they need only for the device, as a whole, to carry out its tasks.

Of course, **someone** must understand how each of these components work, and how the group of them work together. In fact, there are so many components and layers at work in a computer system when someone is performing a typical task (e.g., web browsing, word processing) that **no** single person knows exactly how every part of the system works.

Our goal is to understand how these components are structured and how they interact. We will find that each component can (and should) be broken into *layers*. Moreover, each layer is the result of combining operations and capabilities from the layer below, resulting in qualitatively different operations and capabilities. Somehow, by building one layer on top of another, complete, complex systems are built. We will see that hardware systems, software systems, and networks are all the result of this type of layering, and that certain conceptual ideas and problems repeat themselves across layers and systems.[1]

Since this course does **not** assume that you have any knowledge or experience with computer programming, it will focus primarily on *hardware systems*—the physical circuits and their structure that compose computing devices. When it is necessary, we will develop the programming and algorithmic background that we need.

Here is a brief list of **many** (but not all) of the topics that we will cover, in **roughly** the order that we will cover them:

- **Hardware:**
  - Boolean logic and Boolean algebra
  - Combinational logic circuits (gates)
  - Memory elements
  - Sequential logic and functions

[1] Do you find all of this talk of *layers*, *operations*, *capabilities*, and *systems* all a bit vague and incomprehensible? Don’t be surprised if you do. You might not have much of a sense of what they mean now, but by the end of the course, you will be able to re-read this paragraph and find that these terms have specific meanings that denote a great deal to you. If that weren’t true, it wouldn’t be much of a course, would it?

  - Binary arithmetic
  - Addition and multiplication circuits
  - Arithmetic Logic Units (ALUs)
  - Basic data path and control structures
  - Memory buses and caches

- **Software:**
  - Instruction set architectures (ISA's)
  - Machine code
  - Assemblers and assembly language
  - The booting sequence
  - How operating systems control the system

- **Networks:**
  - Digital and analog signaling
  - Error detection and correction

The nature of the labs and projects for this course will vary with the material throughout the semester. A good deal of time will be spent in our digital lab (Seeley Mudd 006), where we will create computing circuits from basic electronic parts. That is, we'll be working with chips, wires, buttons, and blinking lights. There will also be some time spent in the computer lab (Seeley Mudd 014) on assembly language programming. Additionally, as we develop more complex circuit designs, approaching the design of a general purpose processor, we will combine the use of the digital lab with circuit simulators used in the computer lab.

This course should be fun, because there is a great deal of hands-on experience with the material. It is also a great demystifying course, as you will have a much better understanding of the operation and principles behind the computers that surround us. Note, however, that it is a course with a great many details, as well as a course that is exceedingly cumulative. It will be critical that you stay on top of lectures, labs, and readings at all times.

2 Lectures, labs, and office hours

Lectures and labs: This class meets for lectures on **Monday and Wednesday** of each week, from **11:00 am to 11:50 am** in **Seeley Mudd 207**. Labs will meet, depending on the content of the project, in either **Seeley Mudd 006** (the digital lab) or **Seeley Mudd 014** (the computer lab) on **Friday** from **11:00 am to 11:50 am**.[2]

[2] The current class size will allow us all to meet in a single lab section during this time. However, if enrollments demand it, we will schedule an additional lab section that will meet at another time on Fridays; I will update that information here if needed.

You are expected to be present for **all of the lectures and labs**, and so missing either is strongly discouraged. I will not teach material twice, so if you miss a lecture or a lab, you're on your own. If you must miss lecture or lab due to an illness or other emergency situation, contact me and we will make some kind of accomodation. **If you have a conflict** with a lecture or lab for any non-emergency—an athletic event, a performance, a vacation—then **the choice is yours to miss or to attend**. If you choose to miss the class meeting, I do **not** want to know *why* nor even *that* you are missing class. You have elected, voluntarily, not to attend, and you must be prepared to obtain and learn the material that you missed on your own. I, of course, recommend that you choose to attend the class meeting when these conflicts arise. Do not underestimate the willingness of those who run extra-curricular programs to make accommodations for your academic demands.

I expect you not only to attend lectures and labs, but also to be attentive for them. The time will be best spent if it is interactive, and that requires that you be up-to-date on the class material, and that you be alert and prepared to participate.

Office hours: I will hold hours in my office (Seeley Mudd 406) weekely. I will schedule these times closer to the semester and update them here.

Furthermore, you can always send email to me if you want to **schedule a meeting at another time**, either in my office or in the lab. I strongly encourage you to ask questions both during and outside of class. The ability to do so is one of the big advantages of small liberal arts colleges, so use it!

3 Texts and materials

The text for this course will be available on these web pages. Specifically, I will be writing it as we go, developing a new textbook that is tailored for this class and for its sequel, *Computer Systems II*. Links to the text will be provided as needed.

4 Assignments, deadlines, and extensions

There will be a number of projects and and a couple of problem sets. The deadline for each will be stated clearly on the assignment, **down to the minute**. The assignment will also state the manner in which you are expected to submit or show your work. **Late submissions will receive failing grades**. Futhermore, **failure to complete any *one* of the labs or projects *may* result in a failing grade for the course**. These assignments are too important to the course not to be completed. If you do not complete the work for an assignment, you are better of **submitting partial work on-time than submitting a complete project late.** If you fall behind in the class, don't avoid the situation—come and talk with me, and we will figure out how best to get you back on track.

An extension for any assignment **must be requested, in writing** (email counts as *writing*), **at least 48 hours prior to the deadline**. The determination as to whether or not a particular situation merits an extension will be made on a case-by-case basis. Scheduled events are **not** sufficient reason to warrant an extension. Rather, extensions are intended for unusual circumstances that prevent you from planning your time well in order to meet the deadline. Note that a sudden onset of illness or other emergency situation that occurs less than 48 hours before a deadline will be treated as a special case.

5 Exams

There will be **two exams** in this course:

1. A one-hour, mid-term exam, given during week 7 of classes, during a lecture hour.
2. A three-hour final exam, given during a scheduled final examination slot. This exam will cover **all of the course material**. This final exam has been scheduled for **Sunday, December 18, from 2:00 pm to 5:00 pm, in Seeley Mudd 207** (our usual classroom).

6 Grading

Your final grade will be determined by a formula roughly like the one below:

- 50% labs/projects
- 15% mid-term exam
- 30% final exam
- 5% free bonus for staying up late and finishing the big projects

7 Academic dishonesty

You will be expected to **do your own work on all assignments and exams** in this course except where explicitly noted on group assignments. While I encourage you to interact with your classmates and discuss the material and assignments, there is a limit to the specificity of such discussions. I seek to make that limit clear here.

It is acceptable to discuss any assignment for the class with a classmate. You may even discuss your approach to a particular problem, or review relevant material for a problem with another person. However, you **may not show another student your work, nor see another student's work. If in doubt, *ask me***. If you are unusure whether or not a particular kind of communication would rise to the level of academic dishonesty, then you should contact me immediately and find out.

8 The big picture

There will be so many details to remember and understand in this course that it is easy to miss the forest for the trees. Don't forget, now and then, to consider the bigger picture: that from one simple level of capabilities, you can create another, fundamentally different, more complex level. As you repeat this process, you develop something so complex from components that are so simple that it seems nearly impossible that the former could arise from the latter. Only because we have seen the progression, step by step, do we see how it is done.

A Revision history

- **Revision 1 [2011-Dec-02]:** Updated with the final exam date, time, and location.
- **Revision 0 [2011-Jul-19]:** The initial, complete version.