University of California – Berkeley
Department of Electrical Engineering and Computer Sciences
CS61c: Summer 2006 – Andy Carle, Instructor
http://inst.cs.berkeley.edu/~cs61c/su06/

1. Introduction

CS61c is focused on the components, interfaces, and operations of modern microprocessors and systems. To build understanding of these topics, we will introduce you to the low-level programming languages C, MIPS, and machine language. We will also make use of a digital logic simulator, Logisim, and several other visualization tools for describing machine fundamentals such as digital design, CPU design, pipelining, caches, virtual memory, and input/output. If you have questions about what we'll be covering, don't hesitate to ask!

2. Do You Belong Here?

The summer session version of this course is a bit different from the regular semester version. We cover all of the usual material, but we do it in half the time. This makes the course very fast. If you fall behind, you will find it almost impossible to catch up. At the same time, the summer course has no restrictions on enrollment. During the regular semester, CS 61B is a prerequisite for this course. However, over the summer, we will waive this requirement. Therefore, anyone, regardless of prior experience, may enroll in the course (until it fills.) We encourage anyone who's curious or interested to take this course, even if they aren't computer science majors!

With that said, this course will be difficult and time-consuming. Your nominal classroom hours are roughly 12hrs/week – however, you can expect to be spending about another 20hrs/week on readings and assignments. If you have other time commitments, such as a summer job or another summer course, you may find yourself stretched too thin. In short, this course will be like a full-time job, so please plan accordingly.

3. Textbooks

The textbooks for this course are *Computer Organization and Design: The Hardware/Software Interface* (hereafter COD) by Patterson and Hennessey, **third edition**, and *The C Programming Language* (hereafter K&R) by Kernighan and Ritchie, second edition. The second edition of COD is vastly different in many ways from the new text and is not recommended. You may find an additional reference on C or UNIX essentials to be helpful.

4. Assignments

- **Reading Assignments** are required readings from COD and K&R. These readings will fill in details not provided in lecture and should round out your understanding of the material.
- **Laboratory Exercises** are relatively simple tasks designed to be completed in your assigned lab section and further build understanding of course materials. You will receive credit for lab by being *checked off* by your lab TA, usually during the lab session.
- **Homework assignments** are more difficult problems designed to solidify and test your understanding of course materials. Most will be of a "book problem" nature, but some will involve extensive programming or other computer use. There will generally be two homework assignments each week (only one in weeks when a project is due), each due approximately one week after they are assigned. These assignments will generally be submitted online and graded by the course readers.
- **Projects** are the most time-intensive assignments in this course, designed to teach you the skill of developing a large program, assess your understanding of the course material, and teach you new details. Expect to spend a lot of time on these. These assignments will generally be submitted online.
- **Exams** will be used to test your understanding of course material and encourage a well-rounded understanding of everything from lecture, assignments, and the readings. We will have two midterm exams and a comprehensive final.

5. Grading

Your course grade will be computed using a system with a total of 300 points:

- **20pts Labs**
- **40pts Homework**
- **60pts Projects (probably 4)**
- **90pts Midterms (2)**
- **90pts Final**

Final letter grades will be determined using the following breakdown:

A+: 291 – 300	A: 271 – 290	A-: 261 – 270
B+: 251 – 260	B: 231 – 250	B-: 221 – 230
C+: 211 – 220	C: 191 – 210	C-: 180 – 190
D: 140 – 179		F: < 140

This is an absolute scale. This means that there is no curve, so you are not in direct competition with the rest of the class. I reserve the rights to bump the entire class up by a few points or alter the grade cutoffs slightly if I decide an exam or the grading standard was too tough, but any changes I make will be strictly to your advantage. This is subject to change before the first midterm.

6. *Late Assignments*

There is very little time to stop and catch up during the summer session. As this applies to the course staff as well as students, **no late submissions will be accepted on homework assignments.** If you fall behind on a homework, I suggest forging ahead to the next assignment instead of spending your time arguing with us about turning one in late. Due to their increased complexity, **projects may be turned in up to 24 hours late, but late submissions will only be eligible for 2/3 of the normal maximum score**. Be aware that the instructional computing servers tend to slow down as a 61c deadline approaches, sometimes becoming completely unresponsive. It is to your advantage to finish assignments early.

7. *Where to Go for Help*

CS61c covers material that is very unlike the material in 61a and 61b. This new material, particularly the Computer Engineering topics, can prove tricky for students who have concentrated on programming thus far in their academic careers. Every semester, bright students who did well in previous CS courses struggle with 61c because they refuse to ask for help. Don't let this happen to you! First, make liberal use of the class newsgroup (details on the course website). Other students and the course staff watch it carefully and are happy to help you out. Ask questions in discussion sections and lecture. Attend office hours (any office hour that is convenient, not just your own TA's) and ask questions there. Finally, if you've exhausted the other options available, feel free to email your TA or Andy with your question.

8. *Academic Honesty*

We like to encourage collaborative studying in 61c. No one enrolled in this course is likely to be an expert in all of the diverse topics covered, so studying in groups for exams and to understand the reading is certainly wise. Unfortunately, to hold each of you accountable for your own knowledge and understanding of the course material we must draw the line somewhere. In CS61c, this line is drawn at anything that will eventually be turned in. Collaboration is not acceptable while working on any homework, project, or exam. This means that you may not: share code with another student, look at another student's code (from this semester or any other), ask another student to debug a portion of your code, share details about or collaborate while coming up with an algorithm to solve an assignment, or collaborate in any way on an exam. For more details about what is and is not acceptable, refer to the "Policy on Academic Honesty," attached and available on the course website.

If you are found to have cheated on an assignment you will be given negative points. E.g. if you cheat on a homework worth 10 points, you will be given a -10. The "giver" and the "receiver" of any illegitimate assistance will be considered equally culpable. All cases of academic dishonesty will be referred to the Office of Student Judicial Affairs.