CS 143 – Introduction to Computer Organization
Course Syllabus and Calendar – Winter 1999

Instructor: Don Colton

Brigham Young University—Hawaii Campus

Abstract

- **Course Number:** CS 143
- **Title:** Introduction to Computer Organization
- **Course Description:** Fundamentals of computer organization and operation. Memory structure, registers, arithmetic and logical functions, instruction formats, addressing modes, languages, and internal-external data representation. (Prereq: Algebra, recommended CS 142)
- **Textbook:** *Computer Organization and Design, The Hardware / Software Interface, second edition*, by: David A. Patterson and John L. Hennessy
- **Class Time:** MWF 11:00–11:50 AM
- **Classroom:** GCB 140
- **Instructor (me):** Don Colton
- **My email:** don@cs.byuh.edu
- **My Office:** GCB 104, Phone: 293-3478
- **My Office Hours:** Winter 1999: MWF 9–10, (subject to change) or drop-in or by arrangement (send email or call for appointment).
- **Teaching Assistant:** Andrew Thompson
- **TA Hours:** MTWTh 4-5, F 3-6, Sa 10-noon
- **Lab:** GCB 101

1 Why Take This Course?

Tools amplify our abilities. An airplane is a tool that lets us fly. A car is a tool that lets us travel quickly from place to place on the ground. A secretary or an assistant is a "tool" to which we can give instructions and get back results. What is a computer? Computers are not human. But they are probably the most human tool that man has ever made.

When is it safe to rely on results from a computer? What are its limits? When we ask it to do such-and-such, how long will it take? Will the answer be right? When the program "breaks," what might have caused the problem? If we really know cars and engines, we drive differently, with a trained ear and a sense of what is really happening. We can push the car to its limits and we know when to back off. With a computer, to really drive it well, it helps to have a similar sense for what the machine is really doing.

We will learn the simple, native language of computers. We will learn how computers think and some of the limits they have. We strive to truly understand computers, and we will make a very good start in this course. We will lay the foundation. We will gain an understanding of bits and bytes, of ands and ors and nots, of integers and floats and doubles, of addition, subtraction, multiplication, and division, of gates, latches, flip-flops, and memories. We will learn to think like a computer, and thereby realize the limitations on the thinking of all computers.

2 Prerequisites

The math we will typically do is limited to adding and subtracting by one, and multiplying and dividing by two. But more than that, we will "manipulate" numbers. We will take them apart and put them back together in different ways. We will see the world of mathematics as a computer sees it, which is probably just a little differently than you ever saw it before. We will also do some programming to see how small and simple things can be combined into complex and powerful results.

Knowledge of algebra and previous programming experience will be very helpful to you. Without them, it may take a bit more time for the classroom presentations and labs to sink in and make sense.

3 Course Content

The course focuses on demystifying computers. We will cover most of chapters three and four of the textbook, but we will also look at other parts of the book (Appendix A and chapters one and five).

4 Attendance

Due to INS (immigration) and VA (veterans) requirements the Vice President for Student Life is notified whenever a student misses four consecutive class periods.

5 Lecture Style

My most important goal is for you to develop intuition about the subject matter, and to get unstuck if you have become stuck. Accordingly, I devote as much time as necessary to the answering of your questions, especially

when those questions seem to be of general interest to the others in the class. (Questions of narrow interest may be deferred to my office.) Otherwise, the time is generally spent in discussion of some topic or other that is closely related to the material in the textbook. I may pose a problem to the class, and moderate as we work through it together. This can serve as a jump-start for understanding and mastering new material. My method of teaching is based on the view that learning is a shared activity between the teacher and the student, and that learning proceeds most quickly when interaction occurs.

6 Work Load

In the United States, the expectation for accredited university-level course work is that there be an average of three hours of work per week for every hour of credit awarded. In a lecture class this means one hour in class and two hours outside of class. It is my goal to keep the work load for an average member of the class at these levels.

Machine language and computer organization seem to be fairly difficult for many beginning students. I hope you will study and learn from the textbook. It is a popular text, and is in widespread use at most major universities in the US. The authors have a nice presentation style and they know their material. It is a good book. You will benefit from reading it. It is the plan of the BYUH computer science department to use this same book in the CS 380 course that some of you will take.

Reading is about 250 pages. It is assigned to specific dates by which you must complete it to receive credit. The course calendar lists by chapter and section the topics that will be discussed. Even if we do not discuss all of the assigned reading in class, you are responsible for it on the exams.

What is the hardest part of the class? We will memorize the basic machine language (about 50 words and what they mean) of the MIPS computer. We will learn to convert numbers and letters and machine language into binary and back. We will learn to perform the basic computer operations (e.g., and, or, add, subtract) just as the computer does. We will also learn to convert high-level statements (e.g., for-loops, while-loops, if-statements, array accesses) into machine language.

There will be nine lab assignments. These involve the writing of assembler programs, the testing of those programs, and the submission of the finished programs via email. For some people this can take a LARGE amount of time. (Especially the last lab.) If you are new to programming, you may need even more time to come up to speed on the whole challenge of running and debugging programs.

7 Grading

Grades will be computed on the basis of points earned on attendance, homework, quizzes, reading, programming labs, and tests (three midterms and a final exam). The weighting is as follows:

20%	reading (scaled from 255 points)
30%	programming (scaled from 310 points)
30%	three midterms
20%	final exam
100%	total

Extra Credit: Some extra credit assignments are available during the semester. The maximum extra credit that can be earned is 10%, even though more (or less) than that *might* be offered. 10% is enough to raise a B to an A.

There is no required "homework" in this class. If I give any homework, it will be for extra credit. The purpose of homework is to help you master the course material in a low-stress setting where resources such as the textbook can be consulted in a leisurely way. **Homework in this course can be done with the aid of other students, and answers can be compared.**

There are no required quizzes in this class. If I give a quiz, it will be for extra credit. The purpose of quizzes is to show me and you how well you are learning on a topic-by-topic basis. Often things on the exams show up on quizzes first.

Labs: The purpose of labs is to experience programming and grow thereby. Programming can be an extreme joy, where time ceases to exist (e.g., "just a minute" can be several hours, but you don't notice). It can be a great pleasure to cause a machine to obey your will, an inch at a time. Or it can be a nightmare, where nothing seems to work right, and the most insignificant things turn out to have far too much significance, and you pull out great clumps of your hair and hit you head against the wall and want to rush right over to your academic advisor and change majors to something you can actually enjoy instead of this misery. Labs reflect the true reality of computer science life. You should experience them.

All labs will be "open-neighbor" labs in the sense that you can confer with other students and lab assistants. You can show your code to others and look at the code others have written. You can also talk about your code, your approach, your difficulties, and your ideas. You can draw pictures and make analogies and ask me (even me) questions.

Due: Labs are due Saturday at noon, even though the calendar shows them as being due on Friday. You can have them graded in person by the T.A. (depending on his availability) or by emailing them to `<don@cs.byuh.edu>` with a subject line that has `cs143` as the first word. (That tells my email handler to forward the email to the T.A. and keep a copy for myself.)

Grading: Labs are graded by the T.A. Grades you feel are unfair can be appealed to me. Since the T.A. controls part of your grade, you should treat him nicely.

Late Labs: We allow late labs because labs are famous for taking longer than it looked like they might take. For the first week, late labs are worth 80%. After that it depends on a lot of things, so I will not make any promises for what they might be worth, but probably they are worth something. However, **no labs will be accepted after Saturday, April 10 at noon.**

Tests: The primary purpose of tests (examinations) is to gauge student learning by measuring performance in a timed and supervised situation. Some memorization may be required.

It is understood that such a situation creates additional stress for many students. For this reason testing is not used exclusively in the grading process. Each test will receive a scaled (normalized) score and a letter grade indicating the final course grade that would be earned by consistent performance at the level reflected on that test.

Testing Center: Except the final exam, most tests are given in the BYUH testing center. The day of the exam we will preview it in class. Attendance is not required. Taking the test counts as attendance in class. However you may find that getting a preview copy of the exam, and being able to ask me questions about it, are both good reasons to attend. You can take the exam that same day or the next day. Generally I allow unlimited time but no books or notes.

Test Makeup: Exams cannot be made up except when I approve it in unusual circumstances. This is *very* rare.

Grading Scale: I use the following grading scale, both for individual assignments and for the course as a whole.

93%+	A	90–92.9%	A-	87–89.9%	B+
83–86.9%	B	80–82.9%	B-	77–79.9%	C+
73–76.9%	C	70–72.9%	C-	67–69.9%	D+
63–66.9%	D	60–62.9%	D-	0–59.9%	F

I reserve the right to up-scale the scores on any assignment, exam, or whatever, if I feel the absolute numeric grading is too harsh. The transformation may even be non-linear, but in any case, absolute rank order will be preserved and no score will go down.

I welcome the opportunity to discuss specific instances of grading with you, and to hear your requests for different grades than were initially assigned. In fact, I encourage it. Some very good learning occurs in these settings (for you **and** for me). About half the time I end up agreeing with the request. The best time to do this is during my office hours. As an alternative, you can submit your argument in writing, together with the original graded work.

Final grades are generally issued by email soon after the final exam, or in-person if I feel that some discussion might be beneficial. Students are invited to visit my office to claim any exams or homework that I am still holding, and to discuss their academic progress. Interim progress reports are issued to the students several times during the course, generally after the midterm exams and before the final exam.

8 Office Hours

Office hours are posted outside my office door. Currently they are MWF 9–10. Office hours are subject to change, as I might discover the need to attend some meeting somewhere, or visit the men's room, or talk to someone in the computer lab.

Students for whom the posted hours are not convenient, or who just want a guaranteed appointment, can come by whenever my door is open (which is most of the time) or contact me by email to make an appointment.

My "open-door policy" is posted on my office door as follows: "If my door is open (even just a bit) feel free to knock and come in. – Bro. Colton"

9 Communication by Email

When I want to say something, or when you want to say something, if we are not in the same room, my first choice is to do it by email. I far prefer it to telephone calls, for instance. When there is an announcement, I will generally tell you in class or send it to you by email. Such announcements might include clarifications on the homework assignments. You will need to maintain an email account and to provide me with a valid email address.

10 Computer Accounts

You should have a computer account in the Computer Science lab (GCB 101). This account gives you access to UNIX systems, software (including compilers and assemblers), email, some storage, and some paper printing (currently 100 pages per CS class). There are also a few modems for dial-in access. You will use your CS account to do the lab work in this class. See me or a lab worker (GCB 101) to get set up.

11 Subject to Change

It is very unlikely that I will make any major changes, but aside from course number, title, and description, I reserve the right to change anything in this syllabus including the grading policies and the course calendar. Important changes are generally communicated in class and by email to those affected. If my changes are unfair to you, let me know. I will try to fix it.

CS 143 Course Calendar — Winter 1999

Probably reliable but subject to change.

mtg	day	date	time	read	Topic	due (pts)
1	Wed	Jan 6	11am		introduction, syllabus	
2	Fri	Jan 8	11am	A.1	Assembler: chart of major instruction groupings	read (10)
3	Mon	Jan 11	11am	A.2	Assembler: registers, memory, immediate	read (10)
4	Wed	Jan 13	11am		Assembler: arithmetic and logic	
5	Fri	Jan 15	11am		Assembler: jump and branch, absolute vs relative	
	Mon	Jan 18			**Holiday: Human Rights Day**	
6	Wed	Jan 20	11am	1.2	Switches: and, or, xor	read (10)
7	Fri	Jan 22	11am	4.1-2	Coding: binary, integer, ascii (p142)	read (10), **Hello (25)**
8	Mon	Jan 25	11am		State (Monopoly), Clock (Dominoes)	
9	Wed	Jan 27	11am		Switches in CMOS	
10	Fri	Jan 29	11am	B.2-3	Karnaugh maps, implicants, and logic	read (15), **Hello Name (25)**
11	Mon	Feb 1	11am		Any function can be done with and, or, not	
12	Wed	Feb 3	11am	B.5	S/R latch	read (15)
13	Fri	Feb 5	11am		Registers and memory	**Starline (30)**
14	Mon	Feb 8	11am	B.4,6	CPU Layout (p358), Clock	read (10)
15	Wed	Feb 10	11am	3.1-3	Machine Language: specifying opcodes and operands	read (10)
16	Fri	Feb 12	11am	3.4	Assembler to Machine: making life easier	read (10), **Starbox (30)**
	Mon	Feb 15			**Holiday: Presidents Day**	
17	Wed	Feb 17	11am		Exam: meanings, fields, s/r, kmap, binary conversion	Exam (100)
18	Fri	Feb 19	11am		Compilation: assignment statements	**5x+2y-1 (35)**
19	Mon	Feb 22	11am		Compilation: arithmetic operations	
20	Wed	Feb 24	11am	3.5	Compilation: selection / decision making (if)	read (10)
21	Fri	Feb 26	11am		Compilation: iteration (while)	**(3x+7y)/2 (35)**
22	Mon	Mar 1	11am		Compilation: iteration (for)	
23	Wed	Mar 3	11am	3.6	Compilation: function calls (jal/jr)	read (10)
24	Fri	Mar 5	11am		Exam: compilation	Exam (100)
25	Mon	Mar 8	11am	3.9,A.3	Libraries, Linkers	read (10)
26	Wed	Mar 10	11am	A.4	Operating System: boot, load	read (5)
27	Fri	Mar 12	11am	A.7	Operating System: interrupts	read (5), **Stack (40)**
28	Mon	Mar 15	11am	3.11	Compilation: arrays	read (5)
29	Wed	Mar 17	11am		Functions: local variables	
30	Fri	Mar 19	11am	A.5-6	Memory management: global, stack, heap	read (15)
31	Mon	Mar 22	11am		Functions: caller save, callee save (3.6, p140)	
32	Wed	Mar 24	11am	4.3-4	Binary arithmetic: add, subtract, and, or	read (10)
	Fri	Mar 26			**Holiday: Kuhio Day**	**Sub (40)**
33	Mon	Mar 29	11am	4.6	Binary arithmetic: multiply, Booth's algorithm	read (15)
34	Wed	Mar 31	11am	4.7	Binary arithmetic: division	read (10)
35	Fri	Apr 2	11am	4.8	Floating point, big numbers	read (25)
36	Mon	Apr 5	11am		Floating point, small numbers	
37	Wed	Apr 7	11am		Exam: binary arithmetic	Exam (100)
38	Fri	Apr 9	11am	5.1-3	CPU Layout, Datapath, State	read (35), **Fibonacci (50)**
39	Mon	Apr 12	11am		Locality of reference, caching	
40	Wed	Apr 14	11am		Glue between languages	
41	Fri	Apr 16	11am		Review for Final	
	Fri	Apr 23	11-2		**Final Exam, 3 hours, in class**	Final (200)
	Mon	Apr 26	noon		Final Grades by email (probably)	