CS 143 – Introduction to Computer Organization
Course Syllabus and Calendar – Spring 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 1:00–2:50 PM
- **Classroom:** GCB 140
- **Instructor (me):** Don Colton
- **My email:** don@cs.byuh.edu
- **My Office:** GCB 104, Phone: 293-3478
- **My Office Hours:** Spring 1999: MWF 9–10, (subject to change) or drop-in or by arrangement (send email or call for appointment).
- **Teaching Assistant:** Brad Richards
- **TA Hours:** Daily 8pm-10pm, MWF 9am-11am
- **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).

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.

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.)

We generally start out by talking about the upcoming labs or tests. If there is more time, we talk about the reading from the book.

4 Grading

Grades will be computed on the basis of points earned as follows:

500	programming
300	exams and quizzes
200	final exam
1000	total

Grading Scale: I use the following grading scale for this class.

930+	A	900–929	A-	870–899	B+
830–869	B	800–829	B-	770–799	C+
730–769	C	700–729	C-	670–699	D+
630–669	D	600–629	D-	0–599	F

Programming 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.

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.

Discussing/Haggling: 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: 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.

5 GradeBot

GradeBot is my robotic program grader. It (he?) is available 24 hours a day, seven days a week, to grade and return your lab assignments.

GradeBot is correct and authoritative. There is always a particular correct behavior that it wants. You must make your program behave in exactly the way that GradeBot is requiring. Likely this will involve changing the wording of your prompts and/or the spacing and wording of your output. It will not significantly alter the difficulty of the problem.

It is possible but unlikely that GradeBot will make some major crazy mistake. If you find an example of this, bring it to me. I will generally reward you with some extra credit. Otherwise, you must assume GradeBot is right.

6 Lab Submission Rules

There are several rules that I use in this class. **These rules apply to the programs you submit to GradeBot.** They are designed to allow you to learn, but to prevent you from doing things that might let you pass the class without learning. Violation of any of these rules is a violation of the honor code. You will receive a score of zero for any such assignment, and it cannot be made up. Repeated violations may lead to failing the class. Please be careful what you submit.

6.1 The Keystroke Rule

Every keystroke in every lab you submit must come from **your own fingertips.** (If you are handicapped in some way that makes typing difficult or impossible for you, check with me. We can make a special exception for you if necessary.) You can re-use code that you wrote in a prior assignment (or in a prior class or in a prior job).

You are forbidden to submit any code that was not typed by you yourself.

6.2 The Open-Neighbor Rule

All labs are "open-neighbor" in the sense that you can **confer** with other students and lab assistants. You can read their code (if they let you). You can share your code with them. You can talk about your code, your approach, your difficulties, and your ideas. You can draw pictures and make analogies and ask the TA or me (even me) questions. You can use their ideas. However, you cannot submit their code to GradeBot, even if you first modify it.

6.3 The Looking Rule

Except for looking at the textbook, you are not allowed to look at your own code that you will be submitting and somebody else's code at the same time.

6.4 The Challenge Rule

If I think that you may have violated these rules on some particular assignment, I will ask you (by email or in person) to state that you followed these rules. If I don't hear back from you, I will assume that you cheated.

7 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"

8 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.

9 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.

10 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 — Spring 1999

Probably reliable but subject to change.

mtg	day	date	time	read	Topic	due (pts)
1	Wed	Apr 28	1pm		introduction, syllabus	
2	Wed	Apr 28	2pm	A.1	Assembler: chart of major instruction groupings	
3	Fri	Apr 30	1pm	A.2	Assembler: registers, memory, immediate	
4	Fri	Apr 30	2pm		Compilation: assignment statements	
5	Mon	May 3	1pm		Assembler: arithmetic and logic	
6	Mon	May 3	2pm		Compilation: arithmetic operations	**Hello (30) 5/4**
7	Wed	May 5	1pm		Assembler: jump and branch, absolute vs relative	
8	Wed	May 5	2pm	3.5	Compilation: selection / decision making (if)	**Name (35) 5/6**
9	Fri	May 7	1pm		Compilation: iteration (while)	
10	Fri	May 7	2pm		Compilation: iteration (do)	
11	Mon	May 10	1pm		Compilation: iteration (for)	
12	Mon	May 10	2pm	3.6	Compilation: function calls (jal/jr)	**YesNo (35) 5/11**
13	Wed	May 12	1pm		Functions: local variables	
14	Wed	May 12	2pm	3.11	Compilation: arrays	**5x+2y-7 (35) 5/13**
15	Fri	May 14	1pm	4.3-4	Binary arithmetic: add, subtract, and, or	
16	Fri	May 14	2pm	4.6	Binary arithmetic: multiply, Booth's algorithm	
17	Mon	May 17	1pm	4.7	Binary arithmetic: division	
18	Mon	May 17	2pm	4.8	Floating point, big numbers	**(4x+7y)/3 (40) 5/18**
19	Wed	May 19	1pm		Floating point, small numbers	
20	Wed	May 19	2pm	1.2	Switches: and, or, xor	**Starline (45) 5/20**
21	Fri	May 21	1pm	4.1-2	Coding: binary, integer, ascii (p142)	
22	Fri	May 21	2pm		State (Monopoly), Clock (Dominoes)	
23	Mon	May 24	1pm		Switches in CMOS	
24	Mon	May 24	2pm	B.2-3	Karnaugh maps, implicants, and logic	**Starbox (45) 5/25**
25	Wed	May 26	1pm		Any function can be done with and, or, not	
26	Wed	May 26	2pm	B.5	S/R latch	**Ttable (45) 5/27**
27	Fri	May 28	1pm		Registers and memory	
28	Fri	May 28	2pm	B.4,6	CPU Layout (p358), Clock	
	Mon	May 31			**Holiday: Memorial Day**	
29	Wed	Jun 2	1pm	3.1-3	Machine Language: specifying opcodes and operands	
30	Wed	Jun 2	2pm	3.4	Assembler to Machine: making life easier	**Stack (60) 6/3**
31	Fri	Jun 4	1pm	3.9,A.3	Libraries, Linkers	
32	Fri	Jun 4	2pm	A.4	Operating System: boot, load	
33	Mon	Jun 7	1pm	A.5-6	Memory management: global, stack, heap	
34	Mon	Jun 7	2pm	A.7	Operating System: interrupts	**Sub (60) 6/8**
35	Wed	Jun 9	1pm		Functions: caller save, callee save (3.6, p140)	
36	Wed	Jun 9	2pm	5.1-3	CPU Layout, Datapath, State	**Fib (70) 6/10**
37	Fri	Jun 11	1pm		Locality of reference, caching	
38	Fri	Jun 11	2pm		Glue between languages	
39	Mon	Jun 14	1pm		Review for Final	
40	Mon	Jun 14	2pm	tba	tba/40	
41	Wed	Jun 16	1pm	tba	tba/41	
42	Wed	Jun 16	2pm	tba	tba/42	
	Fri	Jun 18	1pm		**Final Exam**	Final Exam (200)
	Mon	Jun 21	noon		Final Grades by email (probably)	