CS 252 – Introduction to Computational Theory
Course Syllabus and Calendar – Winter 2002

Professor Don Colton

Brigham Young University Hawai'i

1 Brief Overview

The "science" of Computer Science is rooted in the limits of what machines can do. This course analyzes those limits in three stages: regular expressions, context-free languages, and Turing machines. Using principles of induction, we prove what limits apply to machines and in a small way confront the principles of intelligence that separate humans from machines.

1.1 The Course

- **Course Number:** CS 252
- **Title:** Introduction to Computational Theory
- **Course Description:** Finite state automata and regular expressions, context-free grammars and push-down automata, Turing machines, computability and undecidability. (Prerequisite: CS 236.)
- **Textbook:** *Introduction to Computer Theory (2e)*, by: Daniel I. A. Cohen ISBN: 0-471-13772-3
- **Class Time:** TTh 11:00–12:20 PM
- **Classroom:** GCB 153
- **Final Exam:** Tue 23 Apr, 7:00–10:00 AM

1.2 The Instructor

- **Instructor (me):** Don Colton
- **My email:** don@cs.byuh.edu
- **My Office:** GCB 130 B
- **Office Hours:** MWF 7–10 AM, 2–4 PM
- **Teaching Assistant:** Andrew Thompson
- **T.A. Hours:** Mon–Thu, 7–11 PM
- **T.A. Location:** GCB 103 (CS Research Lab)

1.3 Office Hours

I am usually in my office with the door open. At least it seems that way to me. I have an open-door policy, 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"

Formally, my office hours for Winter 2002 are MWF 7 to 10 AM and 2 to 4 PM. Any changes will be posted outside my office door. Students for whom the posted hours are not convenient can contact me by email to make an appointment.

1.4 Special Needs

Brigham Young University–Hawai'i is committed to providing a working and learning atmosphere, which reasonably accommodates qualified persons with disabilities. If you have any disability that may impair your ability to complete this course successfully, please contact the students with Special Need Coordinator, Leilani A'una at 293-3518. Reasonable academic accommodations are reviewed for all students who have qualified documented disabilities. If you need assistance or if you feel you have been unlawfully discriminated against on the basis of disability, you may seek resolution through established grievance policy and procedures. You should contact the Human Resource Services at 780-8875.

1.5 Preventing Sexual Harassment

Title IX of the education amendments of 1972 prohibits sex discrimination against any participant in an educational program or activity that receives federal funds, including Federal loans and grants. Title IX also covers student-to-student sexual harassment. If you encounter unlawful sexual harassment or gender-based discrimination, please contact the Human Resource Services at 780-8875 (24 hours).

1.6 Subject to Change

It is possible that I will revise some aspects of the course as we go along. Any changes I make are likely to be to your advantage. If any of my changes seems unfair to you, let me know. I will try to correct it.

2 About the Course

The current generation of physically real machines, as well as all foreseeable generations of machines, are modeled by even more powerful theoretical machines about which we can make hypotheses and prove theorems. The match between model and reality seems to be quite good, and the things we learn by proving theorems about these theoretical machines carry over into the realm of the real.

At the simplest level we have automata that move from internal state to state based on external stimuli. Consider a mosquito that is drawn to warm bodies and clouds of CO_2. Does it really think about its behavior, or is it a captive of a hard-wired brain that must follow its animal instincts? With an automaton, we have a state, a sensory input, and a transition to a new state. Once the transition has occurred, there is no memory of things past. This model fits a large class of computational artifacts and languages, including regular expressions.

Moving to a more complex level, we add a stack for memory. We can "push" to keep track of things we want to get back to, as the person who is interrupted from lunch by a knock at the door, after servicing that interrupt, can remember to return to his lunch. This class of machine is called the push-down automaton, and it corresponds to the class of languages defined by context-free grammars. This model fits a larger class of computational artifacts and languages, including modern programming languages.

Reaching the pinnacle of computing ability, we have the two-stack automaton, which is equivalent to a Turing machine. All modern computers can be simulated on a Turing machine (although inefficiently). With its precise theoretical definition we can answer questions about the limits of what machines can do.

Concluding the semester, we will look at NP-completeness, computability, decidability, and related issues.

3 Attendance

I take roll in this class. Attendance counts for 10% of your final grade. Typically attendance is worth 3 points per day. I take 3-point roll at the start of class. I take 2-point roll about 10 minutes into class. If you come later than that, you can get one point by making sure I notice you in class (maybe right after class). Missing and unnoticed persons get zeros. At the end of the semester I weight the scores so the total comes out at 10%.

Due to INS (immigration) and VA (veterans) requirements the Vice President for Student Life is supposed to be notified whenever a student misses four consecutive class days. I try to do this.

4 Lecture Policies

I typically use a flexible and open lecture style, rather than a regimented sequence of slides. I try to focus on interesting aspects of the subject matter, instead of simply repeating what you have read in the textbook. My goal is that you develop intuition about the subject matter, and get unstuck if you have become stuck. Accordingly, I devote as much time as necessary to answering 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.

5 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. Some of you are accustomed to working more and others less. It is my goal to keep the work load for an average member of the class at these levels.

Generally reading and homework together will not exceed two hours per class period. If you find the workload to be heavier than that, please let me know.

6 Grading

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	80–82.8%	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

Grades will be computed on the basis of points earned on homework, quizzes, lab work, and tests. The weighting is as follows:

10%	attendance
15%	homework and pop quizzes
25%	five labs (programming assignments)
30%	six chapter tests
20%	final exam
100%	total

The weighting will be adjusted at the end of the semester to keep things in the right perspective. For instance, if I do not give six chapter tests, I may drop the total percentage of grade that comes from chapter tests.

Final Exam: Even though the final exam may count for just a small percentage of your overall grade, you must pass the final exam (60.0% or better) in order to get a C or better in the class.

Homework: The primary purpose of homework is to encourage students to master the course material in a low-stress setting where resources such as the textbook can be consulted in a leisurely way. It is my policy that **regular homework in this course can be done with the aid of other students, and that answers can be compared.** It is not in anyone's best interest if answers are simply copied from person to person without at least some attempt at understanding. Generally homework means answering questions from the end of each section in the book. **Extra-credit (bonus) homework problems may be assigned from time to time. These add to your homework score, but are not required. Unlike regular homework, these must be done without the aid of other people, except that you can consult books or ask me (the instructor) for assistance.**

Late Homework: Homework assignments are due at the start of class, and should be turned in to me (at the front of the classroom) when you arrive. Typically I like to discuss a homework assignment on the day that it is turned in, or on the day that I return the graded assignment to the students of the class. This often involves disclosing the answers and discussing how the answers were derived. No late work is accepted **after the homework is discussed in class,** except when I approve it in unusual circumstances.

Quizzes: The primary purpose of quizzes is to measure student learning on a topic-by-topic basis. It shows me how the students are doing, and it shows you where you might need more attention before the bigger tests.

Quiz Makeup: Because quizzes are a small part of the final grade, they cannot be made up except when I approve it in unusual circumstances.

Programming: There will be several programming assignments in this class. The emphasis is on learning the chapter material by developing your programming skills, including specification reading, debugging, and documenting your work.

You may also be asked to write programs for homework, quizzes, or tests, but these will be read by a human rather than being compiled and executed by a machine. The emphasis will be demonstrating your understanding of the algorithms and problems discussed in class and in the textbook, rather than on your skill at debugging and testing your work. Accordingly, well-commented pseudo-code is expected.

Tests: The primary purpose of tests (examinations) is to gauge student learning by measuring performance in a (possibly timed) supervised situation. 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.

Test Makeup: Tests cannot be made up except when I approve it in unusual circumstances.

Other notes: 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.

Whenever you think grading may be unfair or incorrect, I encourage you to discuss specific instances of grading with the me, and to argue for a different grade than I initially assigned. Some very good learning occurs in these settings (for you **and** for me). The best time to do this is during my office hours, or immediately before or after class if the issue is brief. 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, about once a month.

Incomplete and UW: If you quit working in the class before achieving a passing grade, I will probably give you a "UW" grade instead of an "F."

I do not give "I" grades (incompletes) except in unusual circumstances. In my experience only a small fraction of incompletes are ever completed. I will consider giving you an incomplete if you request it, seem to have a good reason, have a pretty solid time line for completion, and you get the necessary paperwork filled out.

7 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 are requested to maintain an email account and to provide me with a valid email address.

8 Computer Accounts

As a member of this class, or as a CS major, you are entitled to a computer account in the CS lab. This account gives you access to UNIX systems, software (including compilers and assemblers), email, web surfing, some storage (currently 10 megabytes to start with), and some paper printing (currently 100 pages per CS class). There are also a few modems for dial-in access. If you had a CS account recently, it is probably still active. If not, see me or a lab person (GCB 101) to get set up.

9 Subject to Change

I like to avoid mutual unhappiness, so I avoid changes as much as I can. The course number, title, and description will not change, but I do reserve the right to change anything else in this syllabus at any time for any reason. This includes the grading policies and the course calendar. If you think my changes are unfair, you have the right to complain. As I said, I like to avoid mutual unhappiness, so I avoid changes as much as I can. Any important change will be communicated in class and by email to those affected.

10 Course Calendar

Here is my best guess at the course calendar.

ISECON: I will be off-island for ISECON'2001, the Information Systems Educators Conference, in Cincinnati, Ohio. I am a member of the board of directors for EDSIG, the sponsoring organization, and am also the Proceedings editor. I plan to fly out Tuesday, Oct 30 and return Sunday, Nov 4. The conference is Thursday through Sunday. **There will be no class on Wednesday or Friday.**

ACM: I will be off-island, taking several programming students to compete in the regional ACM programming contest in Riverside, California. We will fly out Thursday, Nov 8 and return Sunday, Nov 11. The contest itself is on Saturday. **There will be no class on Friday.**

Chapter Tests: To make up for lost time, chapter tests will generally be held in the testing center. The final exam will be held in the regular classroom.

CS 252 Tentative Course Calendar Winter 2002

mtg	day	date	time	read	Topic	Comment
1	Wed	Aug 29	noon		Introduction, Syllabus, Grading form, pretest	
2	Fri	Aug 31	noon	1-7	Review of CS235	
	Mon	Sep 3			Labor Day Holiday	**no class**
3	Wed	Sep 5	noon	8.1-2	Intro, Relations	
4	Fri	Sep 7	noon	8.3-5	Keys, Storage, Indexing	
5	Mon	Sep 10	noon	8.6-7	Navigation, Implementation	
6	Wed	Sep 12	noon	8.8+	Algebra, Summary	chapter test
7	Fri	Sep 14	noon	9.1-2	Introduction	
8	Mon	Sep 17	noon	9.3	Implementing Graphs	
9	Wed	Sep 19	noon	9.4	Connected Components	
10	Fri	Sep 21	noon	9.5	Minimal Spanning Trees	
11	Mon	Sep 24	noon	9.6	Depth-First Search	
12	Wed	Sep 26	noon	9.7	Uses of Depth-First Search	
13	Fri	Sep 28	noon	9.8	Dijkstra's One Shortest-Path Algorithm	
14	Mon	Oct 1	noon	9.9	Floyd's All Shortest-Paths Algorithm	
15	Wed	Oct 3	noon	9.10+	Graph Theory, Summary	chapter test
16	Fri	Oct 5	noon	10.1-2	Intro, State Machines, Automata	
17	Mon	Oct 8	noon	10.3	Determinism in Automata	
18	Wed	Oct 10	noon	10.4	Converting Nondeterminism	
19	Fri	Oct 12	noon	10.5-7	Regular Expressions	
20	Mon	Oct 15	noon	10.8	Converting Regular Expressions to Automata	
21	Wed	Oct 17	noon	10.9+	Converting Automata to Regular Expressions	chapter test
22	Fri	Oct 19	noon	11.1-2	Intro, Context-Free Grammars	
23	Mon	Oct 22	noon	11.3-4	Languages, Parse Trees	
24	Wed	Oct 24	noon	11.5	Ambiguity	
25	Fri	Oct 26	noon	11.6	Constructing Parse Trees	
26	Mon	Oct 29	noon	11.7	Table-Driven Parsing	
27	Wed	Oct 31	noon		10/31: Colton at ISECON	**no class**
28	Fri	Nov 2	noon		11/2: Colton at ISECON	**no class**
29	Mon	Nov 5	noon	11.8+	Comparison, Summary	chapter test
30	Wed	Nov 7	noon	12.1-3	Intro, Expressions	
31	Fri	Nov 9	noon		11/9: ACM Programming Competition	**no class**
32	Mon	Nov 12	noon	12.4-5	Truth Tables, Boolean Funcs, Logical Exprs	
33	Wed	Nov 14	noon	12.6	Karnaugh Maps	
34	Fri	Nov 16	noon	12.7-8	Tautologies	
35	Mon	Nov 19	noon	12.9-10	Tautologies, Deduction	
36	Wed	Nov 21	noon	12.11+	Proofs, Summary	chapter test
	Fri	Nov 23			Thanksgiving Holiday	**no class**
37	Mon	Nov 26	noon	14.1-3	Intro, Predicates, Expressions	
38	Wed	Nov 28	noon	14.4-5	Quantifiers, Interpretations	
39	Fri	Nov 30	noon	14.6-7	Tautologies	
40	Mon	Dec 3	noon	14.8-9	Proofs	
41	Wed	Dec 5	noon	14.10+	Truth, Summary	chapter test
42	Fri	Dec 7	noon	12,14	Review for Final Exam	
	Fri	Dec 14	11-2		Final Exam	