CS 320 – Introduction to Computational Theory
Course Syllabus and Calendar – Winter 2003

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 320
- **Title:** Introduction to Computational Theory
- **Course Description:** Finite state automata and regular expressions, context-free grammars and pushdown automata, Turing machines, computability and undecidability. (Prerequisite: CS 201, Math 201.)
- **Textbook:** *Introduction to Computer Theory (2e)*, by: Daniel I. A. Cohen ISBN: 0-471-13772-3.
- **Class Time:** MW 3:00–3:50 PM
- **Classroom:** GCB 140
- **Final Exam:** Mon 21 Apr, 3:00–6:00 PM

1.2 The Instructor

- **Instructor (me):** Don Colton
- **My email:** don@colton.byuh.edu
- **My Office:** GCB 130 B
- **Office Hours:** MWF 9 AM to 11 AM

1.3 Office Hours

My office hours are shown above. You can contact me by email to make an appointment at another time. I also 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"

1.4 Subject to Change

I may make a few changes in course calendar as we go along, but typically I do not make major changes once the course has started. If necessary I reserve the right to change any aspect of the course (except number, title, and description). Important changes are generally negotiated and voted on in class. If any changes seem unfair to you, let me know. I will try to fix it.

2 Course Content

The CS 320 course covers the following CC2001 Knowledge Units. These are defined in Computing Curricula 2001, a joint project of IEEE-CS and ACM. The IEEE Computer Society and the Association for Computing Machinery are the two major professional societies in computer science.

AL5. Basic computability

- Finite-state machines
- Context-free grammars
- Tractable and intractable problems
- Uncomputable functions
- The halting problem
- Implications of uncomputability

AL7. Automata theory

- Deterministic finite automata (DFAs)
- Nondeterministic finite automata (NFAs)
- Equivalence of DFAs and NFAs
- Regular expressions
- The pumping lemma for regular expressions
- Push-down automata (PDAs)
- Relationship of PDAs and context-free grammars
- Properties of context-free grammars
- Turing machines
- Nondeterministic Turing machines
- Sets and languages
- Chomsky hierarchy
- The Church-Turing thesis

3 About the Course

The current generation of physically real machines, as well as all foreseeable generations of machines, are mod-

eled 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.

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 attendance, reading, quizzes, homework, lab work (programming), and tests. The weighting is as follows, however you must pass the final exam (60.0% or better) in order to get a C or better in the class.

10%	attendance
20%	reading
10%	quizzes and homework
10%	one programming lab
30%	three midterm exams
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 three midterm exams, I may drop the total percentage of grade that comes from exams.

7 Attendance

I take roll in this class. Attendance counts for 10% of your final grade. Typically attendance is worth 4 points per day. I take 4-point roll at the start of class. I take 3-point roll about 10 minutes into class. If you come later than that, you can get two points 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.

7.1 Reading

The book is written for a typical CS student with only a little programming background. It will be too easy for some of you, and too difficult for some of you.

To get reading credit, you must let your sight rest on each of the words in the assignment, and you must try to understand what is being said. If you can speed-read some or all of it with reasonable comprehension, that is acceptable too.

After completing the reading, you must submit to GradeBot (see below) a program whose output makes the claim that you did the required reading. The exact wording can be had from GradeBot.

7.2 Programming Labs

There is one programming assignment in this class. The emphasis is on learning the chapter material by developing your programming skills, including specification reading, debugging, and documenting your work.

7.3 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. Quizzes are generally not announced in advance, and can take place at any time.

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.

7.4 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.

7.5 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.

7.6 Final Exam

The final exam is given in a timed and controlled setting. It will review the tests given previously and will include some programming problems to verify your skills demonstrated on the labs.

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.

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.

8 Assignment Calendar

The dates on this list are not guaranteed. They are approximately correct. You should run a GradeBot status report to find the authoritative, correct due dates for you.

```
 2: c01       thru Jan 08 (Tue)   2 pts
 4: c02       thru Jan 10 (Thu)   4 pts
 6: c03       thru Jan 15 (Tue)   3 pts
 8: c04       thru Jan 17 (Thu)   6 pts
10: c05       thru Jan 22 (Tue)   6 pts
12: c06       thru Jan 24 (Thu)   4 pts
15: c07       thru Jan 31 (Thu)  16 pts
17: c08       thru Feb 05 (Tue)   5 pts
19: c09       thru Feb 07 (Thu)   6 pts
21: c10       thru Feb 12 (Tue)   6 pts
23: c11       thru Feb 14 (Thu)   3 pts
25: c12       thru Feb 19 (Tue)  10 pts
27: c13       thru Feb 21 (Thu)   9 pts
29: c14       thru Feb 26 (Tue)   8 pts
31: c15       thru Feb 28 (Thu)  10 pts
33: c16       thru Mar 05 (Tue)   7 pts
35: c17       thru Mar 07 (Thu)   7 pts
37: c18       thru Mar 12 (Tue)   9 pts
39: c19       thru Mar 14 (Thu)  10 pts
41: c20       thru Mar 19 (Tue)  10 pts
43: c21       thru Mar 21 (Thu)   6 pts
46: c22       thru Apr 02 (Tue)  18 pts
48: c23       thru Apr 04 (Thu)  12 pts
50: c24       thru Apr 09 (Tue)  12 pts
52: c25       thru Apr 11 (Thu)  11 pts
54: final     thru Apr 23 (Tue) 100 pts
```

9 Special Needs

Brigham Young University Hawaii 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.

10 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).