

CS 210 – Computer Organization

Course Syllabus and Calendar – Fall 2002

Professor Don Colton

Brigham Young University Hawaii

1 Brief Overview

Great mystery surrounds computers. We know, for example, that everything a computer says is true. Always. And we know that programming is just a matter of finding the right way to ask the question, the right magical incantation to make the computer understand.

Well, it sounds ridiculous stated that way, but it is uncomfortably close to a “truth” that many people seem to believe.

In this class, our main goal is to demystify computing. What exactly happens when a computer computes? How are numbers and letters and formulas processed to give us answers? We will go to the lowest levels of computing, functions such as AND and OR, and numbering systems such as BASE 2. We will see how it is done.

This class is the prerequisite to Computer Architecture. In that class we examine speed and efficiency, topics near and dear to many of us. But more than that, this class is a prerequisite to really understand what is going on under the hood on your computer, to really understand what your program needs to do, and how it does it.

1.1 The Course

- **Course Number:** CS 210
- **Title:** Computer Organization
- **Course Description:** Digital logic, digital systems, machine-level representation of data, assembly-level machine organization. (Prerequisite: CS 101 or above, Math 110 or above.)
- **Textbook:** *Computer Organization and Design, The Hardware / Software Interface, second edition*, by: David A. Patterson and John L. Hennessy, ISBN: 1-55860-428-6.
- **Class Time:** MW 11:00–11:50 AM
- **Final Exam:** Fri 13 Dec, 11:00–2:00 PM
- **Classroom:** GCB 140

1.2 The Instructor

- **Instructor (me):** Don Colton
- **My email:** don@colton.byuh.edu
- **My Office:** GCB 130 B

1.3 The Tutors

- **T.A. Hours:** Mon–Thu, 3 PM to 1 AM
- **T.A. Hours:** Fri, 3 PM to 8 PM
- **T.A. Hours:** Sat, 3 PM to 6 PM
- **T.A. Location:** GCB 101 (CS Lab)

1.4 Office Hours

My office hours for Fall 2002 are TTh 9–3. Updated office hours (when necessary) are posted outside my office door. Students for whom the posted hours are not convenient can contact me by email to make an appointment.

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”

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 CS 210 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.

| mtg | day | date | time | read | Topic | due (pts) |
|-----|-----|--------|------|-------|---|------------------|
| 1 | Wed | Aug 28 | 11am | B.1-3 | intro, syl, gates | |
| | Mon | Sep 2 | | | Holiday: Labor Day | |
| 2 | Wed | Sep 4 | 11am | B.4-5 | Karnaugh maps, memory | |
| 3 | Mon | Sep 9 | 11am | 1.all | overview and history | |
| 4 | Wed | Sep 11 | 11am | A.1 | Assembler: chart of major instruction groupings | |
| 5 | Mon | Sep 16 | 11am | A.2 | Assembler: registers, memory, immediate | |
| 6 | Wed | Sep 18 | 11am | | Compilation: assignment statements | |
| 7 | Mon | Sep 23 | 11am | | Assembler: arithmetic and logic | |
| 8 | Wed | Sep 25 | 11am | | Compilation: arithmetic operations | Hello (30) |
| 9 | Mon | Sep 30 | 11am | | Assembler: jump and branch, absolute vs relative | |
| 10 | Wed | Oct 2 | 11am | 3.1-3 | Machine Language: specifying opcodes and operands | |
| 11 | Mon | Oct 7 | 11am | 3.4 | Assembler to Machine: making life easier | Stack (60) |
| 12 | Wed | Oct 9 | 11am | 3.5 | Compilation: if/else | |
| 13 | Mon | Oct 14 | 11am | | Compilation: loops | |
| 14 | Wed | Oct 16 | 11am | 3.6 | Compilation: function calls (jal/jr) | YesNo (35) |
| 15 | Mon | Oct 21 | 11am | 3.6 | Functions: caller save, callee save (3.6, p140) | |
| 16 | Wed | Oct 23 | 11am | | Functions: local variables | |
| 17 | Mon | Oct 28 | 11am | 3.11 | Compilation: arrays | 5x+2y-7 (35) |
| 18 | Wed | Oct 30 | 11am | 4.1-2 | Coding: binary, integer, signed | |
| 19 | Mon | Nov 4 | 11am | 4.3-4 | Binary arithmetic: add, subtract, and, or | |
| 20 | Wed | Nov 6 | 11am | 4.8 | Floating point, extreme numbers | (4x+7y)/3 (40) |
| 21 | Mon | Nov 11 | 11am | 3.7 | Ascii, Unicode (p142) | |
| 22 | Wed | Nov 13 | 11am | | Records and arrays | |
| 23 | Mon | Nov 18 | 11am | A.5-6 | Memory management: global, stack, heap | |
| 24 | Wed | Nov 20 | 11am | A.7 | Operating System: interrupts | Sub (60) |
| 25 | Mon | Nov 25 | 11am | B.4,6 | CPU Layout (p358), Clock | |
| 26 | Wed | Nov 27 | 11am | 5.1-3 | CPU Layout, Datapath, State | Fib (70) |
| 27 | Mon | Dec 2 | 11am | | Gate delays, fan-in, fan-out | |
| 28 | Wed | Dec 4 | 11am | | Review for Final Exam | |
| | Fri | Dec 13 | 11am | | Final Exam | Final Exam (200) |

Table 1: CS 210 Course Calendar (best guess) — Fall 2002

AR1. Digital logic and digital systems

- Overview and history of computer architecture
- Fundamental building blocks (logic gates, flip-flops, counters, registers, PLA)
- Logic expressions, minimization, sum of products
- Register transfer notation
- Physical considerations (gate delays, fan-in, fan-out)

AR2. Machine level representation of data

- Bits, bytes, and words
- Numeric data representation and number bases
- Fixed- and floating-point systems
- Signed and twos-complement representations
- Representation of nonnumeric data (character codes, graphical data)
- Representation of records and arrays

AR3. Assembly level machine organization

- Basic organization of the von Neumann machine

- Control unit; instruction fetch, decode, and execution
- Instruction sets and types (data manipulation, control, I/O)
- Assembly/machine language programming
- Instruction formats
- Addressing modes
- Subroutine call and return mechanisms
- I/O and interrupts

4 Grading

The allocation of points for grading has not been decided. This paragraph reflects some past practice, but it is under reconsideration. Take it with a grain of salt.

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 |

To get better than a C, you must also take and pass the final exam. Your final grade will be the **lower** of your total-points grade and your final-exam grade. The final-exam grading will be as follows:

| | | | | | |
|-------|---|-------|----|-------|----|
| 93+ | A | 90-92 | A- | 87-89 | B+ |
| 83-86 | B | 80-82 | B- | 77-79 | C+ |

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 your 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.

5 GradeBot

GradeBot is my robotic program grader. It is available 24 hours a day, seven days a week, to test, 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.

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 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.

7 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).