CS 271 – Computer Architecture and Assembly Language

Catalog Description: Introduction to functional organization and operation of digital computers. Coverage of assembly language; addressing, stacks, argument passing, arithmetic operations, decisions, macros, modularization, linkers and debuggers.

Credits: 4   Terms Offered: On-campus: Winter, Summer
              Ecampus: All

Structure:
On-campus: Two 80-minute lectures per week.
Note: Due to extensive programming assignments, this course has an implied, non-scheduled lab. The lab takes place in an EECS computer lab at various times, and is not part of the official course schedule. TAs are available to help the students with programming assignments at times announced in the syllabus.

Ecampus: Term totals: This course combines approximately 120 hours of instruction, online activities, and assignments for 4 credits (30 hours of online instruction, 10 hours of online participation, 2 hours of online quizzes, 30 hours of offline reading/study, 15 hours of offline homework/lab assignments, 30 hours of offline programming projects, and 3 hours of proctored exams).

Prerequisites: CS 151 or ECE 151 or CS 161 or CS 165

Courses that require this as a prerequisite: CS 311, CS/ECE 372, CS 411

Instructors: TBD

Course Content:
- Hardware, architectures
- Internal representation of data, instructions, and addresses
- Boolean Algebra
- Elementary circuits
- Instruction set architecture, micro-programs
- Assembly language
- Debuggers

Learning Resources:
Measurable Student Learning Outcomes:
At the completion of the course, students will be able to...

1. **Identify** the major components of a computer architecture, and **explain** their purposes and interactions.
2. **Simulate** the internal representation of data, and **show** how data is stored and accessed in memory.
3. **Explain** the relationships between a hardware architecture and its instruction set, and **simulate** micro-programs.
4. **Create** and **simplify** circuits that produce specified output for given inputs (e.g., adders, multiplexers, etc.).
5. **Explain** the Instruction Execution Cycle.
6. **Explain** the differences and relationships among high-level, assembly, and machine languages.
7. **Write** well-modularized computer programs in an assembly language, implementing decision, repetition, and procedure structures.
8. **Use** a debugger, and **explain** register contents.
9. **Simulate** the system stack as it is used for procedure calls and parameter passing.
10. **Explain** how editors, assemblers, linkers, and operating systems enable computer programming.
11. **Explain** various mechanisms for implementing parallelism in hardware/software.

Evaluation of Student Learning:
- Exercises (approx. 5%)
- Homework sets (approx. 10%)
- Programming projects (approx. 30%)
- Quizzes (approx. 20%)
- Midterm exam (15%)
- Final exam (20%)

Students with Disabilities:
Accommodations are collaborative efforts between students, faculty and Disability Access Services (DAS). Students with accommodations approved through DAS are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through DAS should contact DAS immediately at 737-4098.

Link to Statement of Expectations for Student Conduct, i.e., cheating policies [http://oregonstate.edu/admin/stucon/achon.htm](http://oregonstate.edu/admin/stucon/achon.htm)

Revised: 05/20/2012