Computer Organization
CS 1410 (810:041)
Syllabus

MEET YOUR INSTRUCTOR

Dr. Mark Fienup, Associate Professor of Computer Science

Dr. Fienup is an Associate Professor of Computer Science. He received his B.A. from the University of Northern Iowa in Mathematics and Chemistry. He earned both his M.S. and Ph.D. in Computer Science at Iowa State University. He has been a faculty member of the Computer Science Department at the University of Northern Iowa since 1985.

Dr. Fienup regularly teaches Introduction to Computing, Data Structures, Computer Organization, and Computer Architecture. His current research activity centers on the Bioinformatics topic of automatics, three-dimensional protein structure prediction.

COURSE OVERVIEW

Computer Organization provides an introduction of how a computer operates at the digital-logic level, machine-language level, and the assembly-language level. It can be taken after or concurrently with an introductory programming class such as UNI's Introduction to Computing (810:051) course. This prerequisite is useful since an understanding of a high-level programming language is useful before studying lower-level, assembly-language programming in the second half of the course. The first half of the course is geared toward understanding how individual computer components (e.g., CPU, memory, bus, control unit) operate at the logic gate level, and how they are connected together to perform machine-language instructions.

Course Objectives

After this course, you should understand:

- simple combinational and memory circuits used to build computer components,
- how these circuits are organized to build a computer,
- how data is represented and manipulated on the computer,
- how to program in assembly language,
- how high-level language programming languages are implemented with respect to the run-time stack and built-in data structures such as arrays and records, and
- general concepts of hardware support necessary for an operating system.

Textbook:

Linda Null and Julia Lobur, *The Essentials of Computer Organization and Architecture*, second edition, Jones and Bartlett Publishers, 2006. ISBN-10: 0-7637-3769-0.

Software Tools:

You will need to download and install the free MARIE simulator from the web-site: http://computerscience.jbpub.com/ecoa/2e/student_resources.cfm

Plus, you will need to download and install the free MIPS simulator from the web-site: http://pages.cs.wisc.edu/~larus/spim.html

COURSE ORGANIZATION

Prerequisite or Corequisite: Introduction to Computing (810:051) or equivalent. If you have, or are, taking an introductory programming course in any high-level programming language (C, C++, Java, Python, Ada, Ruby, Scheme, etc.), you satisfy

the necessary prerequisite.

Written Assignments: Assignments will be both "pencil-and-paper" and assembly-language programming exercises. Assignments are organized around the chapter sections of the textbook according to the below schedule. For each assignment, you should follow these steps:

1) Read the corresponding chapter section(s) thoroughly.

2) Check the outline below to see if I have provided additional supplemental material to further explain a textbook topic or provide additional examples.

3) Complete the assignment related to the chapter section(s) and submit your solution to the **Assignment Submission** link. **Need help?** See the eLearning Tutorials for instructions on how to submit an assignment.

Assignments consist of two types:

a. Textbook and instructor supplied exercises which are short-answer or problem based questions. You can use a word processor program (Word, Open-Office, etc.) to save your responses to a file in Rich-text format (.rtf). However, many of the assignments require complex diagrams that are more easily done on paper, so you might want to use a scanner/printer or digital camera to generate a .jpg file. Zip multiple files together and submit a single .zip file for the assignment.

b. Assembly-language programs as text files and their associated output files. Zip your output and assembly-language program files together and submit a single .zip file for the assignment.

Examinations: There will be three proctored exams: two and a comprehensive final exam. Examination request forms are included at the appropriate places in the **Course Content**.

Course Outline:

Chapters		**Supplemental Material**	**Assignment**
Sections	**Topic**		
1.1 - 1.5	Intro. to Computer Organization and Computer History		Written Assignment 1
2.1 - 2.4	Binary and Hexadecimal number systems; Unsigned Integers; Signed Integers: two's complement		Written Assignment 2
2.5 - 2.7	Floating point representation; Character representation; Error correction and detection		Written Assignment 3
3.1 - 3.5	Boolean Algebra; Logical Gates; Common Combinational Circuits		Written Assignment 4
3.6	1-bit latches, Timing Diagrams; Flip-Flops, Registers, Register File	Supplement 5	Written Assignment 5
Exam 1: covering chapters 1- 3			
4.1 - 4.7	CPU, Bus, Clock, I/O, Memory, Interrupts		Written Assignment 6

4.8 - 4.10	Intro. to MARIE assembly instructions, Fetch-decode-execute cycle; simple programs	Supplement 7	Written Assignment 7
4.11 - 4. 12	Assemblers; MARIE subroutine instructions and programming		Written Assignment 8
4.13	MARIE Hardwired and Microprogrammed Control Units	Supplement 9	Written Assignment 9
Exam 2: covering chapter sections 4.1 - 4.13			
4.14	Intel x86 (CISC), MIPS (RISC) architectures, simple MIPS assembly language programming, arrays	Supplement 10	Written Assignment 10
	Run-time stack; MIPS procedure/funciton calling conventions; PCSpim System calls	Supplement 11	Written Assignment 11
	MIPS logical and shifting instructions; bit-string set representation	Supplement 12	Written Assignment 12
4.5 7.1-7.4 8.1-8.4	Hardware Support for OS: CPU timer, privileged instructions, dual-mode CPU operation, memory protection	Supplement 13	Written Assignment 13
Final exam: comprehensive, but focusing on details since Exam 2			

GRADING

Grading policy: Course components are weighted as:

Written Assignments:	30 %
Exam 1:	23 %
Exam 2:	23 %
Final Exam:	24 %

Course grades will be assigned based on the following grading scale:

100-90	A
89 - 80	B
79 - 70	C
69 - 60	D
Below 59	F

Plus and minus grades will be assigned for scores within two percentage points from a grade cutoff (e.g., 91.9 to 90 is an A-, and 89.9 to 88 is a B+).

UNI Guided Independent Study requires that you submit all assignments and complete exams to recieve a grade in the course.