ECE 199JL: Introduction to Computer Engineering **Fall 2012**

Course Overview

Instructors: Douglas L. Jones (dl-jones@illinois.edu)
113 Coordinated Sciences Laboratory
Office hours: TBD

Steven S. Lumetta (lumetta@illinois.edu)
209 Coordinated Sciences Laboratory
Th 1-3 upstairs at miaZa's
(west of Wright on south side of Green)

Discussion Leader: Geoffrey Herman (glherman@illinois.edu)
Teaching Assistants:
Sai Ma (saima1@illinois.edu)
Seyed Ahmadyan (ahmadya2@illinois.edu)
Office hours for will be announced on the web site.

Together with ECE199KL, this course gives an introduction to the design and programming of computing systems. We start the course by motivating our objectives and connecting them with students future ECE studies and career paths. The philosophy of our approach is quite different than the typical introduction to programming course: after a brief illustration of our goals and objectives with a quick introduction to C, we approach programming from the bottom upwards. In particular, we begin by describing the architecture of a computer, including logic gates, datapaths, registers, and memories. Throughout the course, we will make connections between hardware and software and explore the engineering tradeoffs in using each to develop computing systems.

The second course in the sequence (ECE199KL) will focus on C programming, where each new C concept will be related to the fundamental concepts described in ECE199JL. We will cover basic programming concepts, functions, arrays, pointers, I/O, recursion, simple data structures, and concepts in object-oriented programming. A bottom-up understanding of computing systems has proven more successful in helping students to understand advanced concepts in computing that follow in the ECE curriculum.

This course has no prerequisites.

Objectives of the course:

After completing this course, students should

- understand the role and importance of abstraction in computing systems,
- recognize and be able to make use of standard digital components in designing simple combinational logic,
- be able to design and implement a simple finite state machine (FSM) as a clock-synchronous sequential circuit using digital logic,
- be able to design and implement a simple FSM using assembly language, and
- be able to write simple programs in both C and assembly language.

Since this is a freshman course, we also have explicit objectives for providing students with an understanding of the profession. In particular, students who have completed this course should

- understand the expectations of the engineering discipline in terms of effort, quality, and objectivity,
- recognize that self-motivation and lifelong learning are necessary to success in engineering,
- be able to articulate the importance of understanding tradeoffs, and
- be able to recognize and identify basic design tradeoffs.

Textbook:

Yale N. Patt and Sanjay J. Patel, *Introduction to Computing Systems: from bits and gates to C and beyond*, 2nd Edition, McGraw-Hill, 2003.

This course is mostly based on the textbook listed above, but we will provide different examples, additional reading materials, and a somewhat different viewpoint in the lectures.

What you should expect:

There will be **weekly homework assignments**, most of which will include some computer work. They will be posted on the webpage for the course and due in class on Fridays.

There will be **five laboratory assignments**. The first will introduce you to the EWS laboratory environment and the use of command-line interfaces that you will use for your homework and programming assignments. The next two will give you hands-on experience with digital logic implementation. The last two will be programming assignments and will involve assembly-level programming using the LC-3 simulator. The second assignment will be fairly substantial. Some of the lab assignments will have intermediate checkpoints that you will turn in.

There will be **three midterm exams** and **one final exam**. The midterm exams will be from 8 to 10 p.m. on the evenings of Tuesday 18 September, Tuesday 16 October, and Tuesday 13 November. The final exam is on Tuesday 18 December from 1:30 to 4:30 p.m. Any conflict that you have with any of the exams **must be reported** to either Professor Jones or Professor Lumetta **at least one week before the exam**, but please report such conflicts as early as possible. Conflict exams for the midterms will be held just before the regular time, from 6 to 8 p.m. on the same day.

Grading mechanics:

Homeworks:	15%
Lab Assignments:	15%
Exams:	15% for each of three midterms; 25% for the final

Website and web board:

The website (http://courses.engr.illinois.edu/ece199/) will contain important announcements, lecture notes, handouts, and other material helpful for succeeding in this course.

The ECE199JL "web board" is available through the web boards link at http://my.ece.illinois.edu. Note that this is NOT the same as the campus' web board project. The web board serves as a forum for students to post and answer questions, discuss issues, warn of pitfalls, etc. You should read the board at least once a day. The TAs and I will read and post to the web board to focus discussions and to provide more definitive answers to posted questions.

Final thoughts:

We want to meet with each of you early in the semester, so we have integrated a short survey and meeting as part of your first laboratory assignment.

Challenge assumptions: Computer Science and Engineering deals with man-made artifacts, and you may be able to invent better ways to make them. Innovation requires that someone challenge the current way of doing things.

You are encouraged to study in groups, and to come to office hours in groups. Studying in groups usually will result in all of you understanding the material better. You, working with other members of your study group, can often unravel concepts to the benefit of all members of the group much better than one can person can, working alone.

Although we encourage you to study together, all work products of this course (homeworks, programming assignments, examinations) must be your own individual work. Do not, for example, exchange code with others. We will use code comparison tools to identify violations. If you cheat, you violate the soul of the University, which we take very seriously, and will not compromise. First offense will, in the least, result in a 0 on the assignment or exam. The policy for the course is based on Article 1.4 of the *Student Code* (available at http://www.admin.uiuc.edu/policy/code/).

ECE 199JL: Introduction to Computer Engineering **Fall 2012**

Douglas L. Jones and Steven S. Lumetta, Instructors
Geoffrey Herman, Discussion Leader
Sai Ma and Seyed Ahmadyan, TAs

Course Syllabus

Lecture	Date	Topics	Book Sections
1	8/27	Overview and orientation; computer systems organized as a systematic set of transformations	Ch. 1
2	8/29	Representation using bits: unsigned and signed integers	§ 2.1—2.4
3	8/31	Operations on bits: arithmetic	§ 2.5
4	9/5	Operations on bits: logical	§ 2.6
5	9/7	Other representations: ASCII, floating point; hexadecimal notation	§ 2.7
6	9/10	C programming: basic data types, program structure	Ch. 11
7	9/12	C programming: basic I/O, operators, control structures	§ 11.5, 12.1—12.3, 13.1—13.3
8	9/14	C programming: examples, problem solving	§ 12.4, 13.4
	9/17	**OPTIONAL REVIEW SESSION FOR MIDTERM #1**	
9	9/19	Logic gates	§ 3.1, 3.2
10	9/21	Boolean algebra: truth tables, Boolean expressions	
11	9/24	Boolean algebra: logical completeness and two-level design	
12	9/26	Boolean algebra: canonical forms, minimization	
13	9/28	Boolean algebra properties	
14	10/1	Gates and combinational logic design examples	
15	10/3	Combinational logic design examples	
16	10/5	Sequential logic components: latches, flip-flops	§ 3.4
17	10/8	Sequential logic components: registers	
18	10/10	Finite state machines: abstraction	
19	10/12	Clock synchronous design	
	10/15	**OPTIONAL REVIEW SESSION FOR MIDTERM #2**	
20	10/17	FSM example mapping to digital logic	§ 3.6
21	10/19	FSM example mapping to digital logic	
22	10/22	Combinational logic structures: decoders, muxes	§ 3.3
23	10/24	Memory	§ 3.5
24	10/26	FSM design example	
25	10/29	FSM design example	

26	10/31	FSM design example	
27	11/2	The von Neumann Model: basic components	§ 4.1, 4.2
28	11/5	The von Neumann Model: instruction processing, sequencing	§ 4.3—4.5
29	11/7	The LC-3 Instruction Set Architecture: operates, data movement, control	§ 5.1—5.4
30	11/9	Example program in LC-3 machine language	§ 5.5, 5.6
	11/12	**OPTIONAL REVIEW SESSION FOR MIDTERM #3**	
31	11/14	Problem solving using systematic decomposition	§ 6.1
32	11/16	More programming examples, debugging	§ 6.2
33	11/26	Assembly language: introduction	§ 7.1, 7.2
34	11/28	Assembly language: examples	
35	11/30	Assembly language: how assemblers work	§ 7.3
36	12/3	Design of the LC-3 datapath; optimization of elements such as ALU	
37	12/5	Control unit design as FSM (microcode)	App. C
38	12/7	Implementation of the LC-3	App. C
39	12/10	I/O in hardware and software (assembly)	Ch 8
40	12/12	Wrap-up and advice	

Supplemental notes will also be provided, particularly for the second and third parts of the class.

ECE 199JL: Introduction to Computer Engineering **Fall 2012**

Douglas L. Jones and Steven S. Lumetta, Instructors
Geoffrey Herman, Discussion Leader
Sai Ma and Seyed Ahmadyan, TAs

Course Timeline

Due Date	Homework/Project/Exam
Fri. 8/31	Homework 1: representations and bits
Tues. 9/4	Lab 0: introduction to the lab environment
Fri. 9/7	Homework 2: operations on bits
Fri. 9/14	Homework 3: basic C programs
Tues. 9/18	**Midterm Exam 1: 8:00 – 10:00 p.m.**
Fri. 9/21	Homework 4: more representations
Fri. 9/28	Homework 5: Boolean algebra
Fri. 10/5	Homework 6: logic elements
Fri. 10/12	Homework 7: basic sequential logic
Tues. 10/16	**Midterm Exam 2: 8:00 – 10:00 p.m.**
Fri. 10/19	Homework 8: finite state machines
Tues. 10/23	Lab 1: combinational logic
Fri. 10/26	Homework 9: logic components
Tues. 10/30	Lab 2.1: sequential logic (checkpoint)
Fri. 11/2	Homework 10: logic design with components
Tues. 11/6	Lab 2.2: sequential logic
Fri. 11/9	Homework 11: LC-3 machine language
Tues. 11/13	**Midterm Exam 3: 8:00 – 10:00 p.m.**
Fri. 11/16	Homework 12: LC-3 flow chart (for Lab 3)
Tues. 11/27	Lab 3: LC-3 binary programming
Fri. 11/30	Homework 13: LC-3 encoding and ISAs
Tues. 12/4	Lab 4.1: LC-3 assembly programming
Fri. 12/7	Homework 14: the LC-3 datapath
Tues. 12/11	Lab 4.2: LC-3 assembly programming
Tues. 12/18	**Final Exam: 1:30 – 4:30 p.m. (location TBD)**

(The back side of this page provides more details on due date timing and policy.)

Homeworks are due in class on the dates indicated above (all Fridays). They are due at the ***beginning*** of lecture. We will hand out solutions sometime in the middle of class, after which homeworks will not be accepted.

Projects are due on the dates indicated above at times indicated on the project descriptions (usually at 10 p.m. on Wednesdays). We will use automated hand-in script for submitting your project. More details later.

Examination conflicts (if any) must be reported no less than one week before the exam in order to be accommodated. Please let one of the professors know as soon as you know about them, however.

Late policy: We will not accept late homeworks. Late projects lose two points (of 100 total points) per hour late or fraction thereof.