ECE 198KL: Introduction to Computer Engineering II **Spring 2013**

Course Overview

Instructor: 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)

Together with ECE198JL (ECE199 in Fall 2012), which you should have already taken, this course gives an introduction to the design and programming of computing systems.

This course (ECE198KL), the second in our new sequence, will focus on C programming, where each new C concept will be related to the fundamental concepts described in ECE198JL. We will start by finishing our coverage of low-level concepts such as I/O, subroutines, and stacks in LC-3 assembly language, then move on to C. 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.

Again, this course requires that you take ECE198JL first. Concurrent enrollment is not acceptable. If you have not met this requirement but think that you should still enroll, speak with the instructor.

Objectives of the course:

After completing this course, students should

- be familiar with basic data organizations such as arrays, structures, lists, trees, and jump tables, and how they are laid out in memory,
- be able to write assembly language programs that make use of these data structures to accomplish simple tasks,
- understand the transformation between programming constructs in languages such as C and their implementation on a modern microprocessor,
- be able to write C programs to accomplish simple tasks, such as functional simulation of a processor,
- understand the importance of structuring code in a way that it can be tested, be able to write effective tests for code, and be familiar with tools for aiding in this process, and
- be familiar with object-oriented design, information hiding, inheritance, and access control in the context of C++, and be able to write basic programs that make use of these ideas.

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. See the syllabus for more details on content.

What you should expect:

There will be **weekly programming assignments**. They will be distributed on Monday and due (by computer handin) the following Monday. These assignments will use the EWS laboratory environment and Android tablets (Nexus 7). The first three will require you to write LC-3 assembly language, while the remaining 11 will require you to write C code. Some of the assignments may build from week to week.
There will be **weekly programming studios** in which we work together as a class on a piece of code for the week's programming assignment. Attendance is mandatory and will be counted as part of your assignment grade for that week.

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 5 February, Tuesday 5 march, and Tuesday 9 April. The final exam is on Thursday 9 May from 1:30 to 4:30 p.m. Any conflict that you have with any of the exams **must be reported** to Professor Lumetta **at least one week before the exam**, but please report such conflicts as early as possible. Ideally, 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:

Programming Assignments:	25%
Midterms:	15% each
Final Exam:	30%

Website and web board:

The website (http://courses.engr.illinois.edu/ece198kl/) will contain important announcements, lecture notes, handouts, and other material helpful for succeeding in this course.

The ECE199KL "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. I will read and post to the web board to focus discussions and to provide more definitive answers to posted questions.

Final thoughts:

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, with the sole exception of the parts of the programming assignments identified for group solution during the studio hour, all work products of this course (programming assignments and 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 198KL: Introduction to Computer Engineering II **Spring 2013**
Steven S. Lumetta, Instructor

Course Syllabus

Lecture	Date	Topics	Book Sections
1	1/14	Review: problem solving using systematic decomposition, examples	Ch. 6
2	1/16	Review: assembly language programming and process	Ch. 7
3	1/18	I/O abstractions: input from the keyboard, output to the monitor	§ 8.1, 8.2, 8.3, 8.4
4	1/23	Repeated code: subroutines and TRAPs	Ch. 9
5	1/25	Repeated code: interrupts and libraries	Ch. 9
6	1/28	Stacks: abstraction and use in assembly code	Ch. 10
7	1/30	Stacks: conventions and use in building subroutines	Ch. 10
8	2/1	Re-introduction to C programming: data types, operators, scope, and storage class	Ch. 11, Ch. 12
	2/4	**OPTIONAL REVIEW SESSION FOR MIDTERM #1**	
9	2/6	C examples using variables and operators	Ch. 11, Ch. 12
10	2/8	C control and iteration constructs	Ch. 13
11	2/11	Comprehensive C examples	(Ch. 11 – 13)
12	2/13	Problem solving with C	(Ch. 11 – 13)
13	2/15	Functions: introduction, syntax, stack frames/activation records	§ 14.1, 14.2, 14.3
14	2/18	C examples using functions	§ 14.3, 14.4
15	2/20	Pointers and Arrays: introduction	§ 16.1, 16.2
16	2/22	Pointers and Arrays: problem solving and examples	§ 16.3
17	2/25	Testing and Debugging: introduction, error taxonomy	§ 15.1, 15.2, 15.3
18	2/27	Testing and Debugging: strategy and tactics	§ 15.4, 15.5, 15.6
19	3/1	Testing and Debugging: examples	(Ch. 15)
	3/4	**OPTIONAL REVIEW SESSION FOR MIDTERM #2**	
20	3/6	Recursion: introduction and basic examples	§ 17.1, 17.2, 17.3
21	3/8	Recursion: example showing run-time stack	§ 17.4, 17.5, 17.6
22	3/11	Input and Output in C: introduction and underpinnings	Ch 18
23	3/13	C library I/O functions	notes
24	3/15	Extensive I/O example	notes
25	3/25	Data Structures: introduction	§ 19.1, 19.2
26	3/27	Data Structures: defining new types, enumerations	§ 19.3
27	3/29	Dynamic memory allocation and management	§ 19.4

28	4/1	Dynamic arrays and linked lists	notes, § 19.5
29	4/3	Dynamic memory implementation	notes
30	4/5	Comprehensive example	notes
	4/8	**OPTIONAL REVIEW SESSION FOR MIDTERM #3**	
31	4/10	Pointer-based data structures: introduction	notes
32	4/12	Pointer-based data structures: problem solving	notes
33	4/15	Structure hierarchies and subtyping	notes
34	4/17	Inheritance, information hiding, and object-oriented design	notes
35	4/19	C++ concepts: classes, inheritance, and type management	notes
36	4/22	C++ concepts: access control and namespaces	notes
37	4/24	C++ concepts: constructors and destructors	notes
38	4/26	C++ concepts: bug avoidance	notes
39	4/29	C++ concepts: dynamic memory management	notes
40	5/1	Wrap-up and advice	notes

ECE 198KL: Introduction to Computer Engineering II **Spring 2013**
Steven S. Lumetta, Instructor

Course Timeline

Due Date	Homework/Project/Exam
Mon. 1/21	Program 1 (LC-3 assembly)
Mon. 1/28	Program 2 (LC-3 assembly)
Mon. 2/4	Program 3 (LC-3 assembly)
Tues. 2/5	**Midterm Exam 1: 8:00 – 10:00 p.m. in the lab**
Mon. 2/11	Program 4 (C)
Mon. 2/18	Program 5 (C)
Mon. 2/25	Program 6 (C)
Mon. 3/4	Program 7 (C)
Tues. 3/5	**Midterm Exam 2: 8:00 – 10:00 p.m. in the lab**
Mon. 3/11	Program 8 (C)
	(Spring Break week)
Mon. 3/25	Program 9 (C)
Mon. 4/1	Program 10 (C)
Mon. 4/8	Program 11 (C)
Tues. 4/9	**Midterm Exam 3: 8:00 – 10:00 p.m. in the lab**
Mon. 4/15	Program 12 (C)
Mon. 4/22	Program 13 (C)
Mon. 4/29	Program 14 (C)
Thurs. 5/9	**Final Exam: 1:30 – 4:30 p.m. in the lab**

Programs are due on the dates indicated above at times indicated on the project descriptions (10 p.m. on Mondays). You will submit your project using Subversion. Generally, program assignments will go out on Monday in class. You have until Thursday to work on them on your own, if you wish (and you are encouraged to do so). On Thursday, we will develop a part of the program together in programming studio. You must then complete the program by the following Monday night.

Please note: except for the portion of the program written during the studio hour, **all programs must be your own individual work.**

I will try to provide 'extra' portions on each program assignment for those of you who wish to obtain James Scholar honors credit for this course or who just want more of a challenge.

Midterm exams will be held in the computer lab (probably L440 DCL).

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: Late programs lose two points (of 100 total points) per hour late or fraction thereof.