ECE 220 - Computer Systems & Programming

Instructors:

Professor Steve Lumetta is the original instructor for the class and is responsible for much of the course material and homework. Historically, Professors Sayan Mitra (who has previously taught ECE 190), Alexander Schwing, Yih-Chun Hu (another ECE 190 professor), Sanjay Patel (the co-author of the class textbook), and Yuting Chen have taught the course as well. It was most recently taught by Professors Hu, Patel, and Chen. The summer course in the past several years has been taught by Professor Volodymyr Kindratenko.

Prerequisites:

ECE 120 is a prerequisite for the class. ECE 220 is a programming-focused class but it assumes no prior programming experience besides the introduction to LC-3 assembly found in ECE 120. However, any prior programming experience should help students understand some of the concepts taught in the class more easily.

When to Take It:

ECE 220 should be taken the semester immediately after taking ECE 120, as the class is meant to be a continuation of ECE 120. As such, it is not advisable to take ECE 120 and ECE 220 concurrently, though there may be some special cases. In general, it is recommended for students to take this sequence as soon as possible, as these classes are part of the core ECE curriculum and are prerequisites for other important classes such as CS 225 and ECE 391. For freshmen who take ECE 120 during their second semester but want to take ECE 220 as soon as possible, the course is also offered as a more time-intensive 8-week course during the summer.

Class Content:

The main goal of ECE 220 is to provide students with a bottom-up approach to learn a solid programming base to lead into future courses such as CS 225 and ECE 391. The class starts out by refreshing students on the LC-3 ISA and quickly adds new concepts such as I/O, subroutines, TRAP operations, and the stack. Around the first midterm the course shifts into learning the C programming language, often referencing LC-3 and sometimes even breaking down pieces of code in C into LC-3 assembly. After learning some of the basic syntax and operators of C, students learn concepts such as pointers, arrays, I/O, and recursion. The course also has some amount of data structures built into it, starting from structures and enumerations then leading into dynamic memory allocation, linked lists, and basic tree structures. The coverage of data structures is wide and not too deep, aiming to better prepare students for future data structure courses such as CS 225. Several basic sorting algorithms are also covered in this course. Towards the end of the class, the curriculum focuses on a few core differences between C and C++, looking at concepts such as classes and object-oriented design, inheritance, access control, and dynamic memory management.

Due to the new steps taken by the ECE department towards prerequisite changes and overhauls to CS 225, there is the possibility of ECE 220 switching over to C++. The main change this would imply is a more object-oriented design introduction to software, and a simpler transition to legitimate C++ near the end. The changes in CS 225 have also caused a ramped-up intro to C++, with concepts typically introduced in CS 225 (templates, references, etc.) being moved back to the end of ECE 220. The issue with the C++ overload at the end is that it is a massive amount of information for the time it is taught in, but the information was typically re-covered in the first weeks of CS 225, but going forward this may not be true, so it may be worth learning the material in greater depth after the course, or before CS 225. Typically, the concepts taught in this final section were the concepts of encapsulation, abstraction, inheritance, and polymorphism. These concepts are never formally implemented in a thorough MP, for the MP is rather introductory and basic for the concepts taught.

Work:

ECE 220 is a four credit-hour class, but the weekly time commitment can vary greatly based on programming skill. Students are given a machine problem (MP) every week. Usually the MPs task students with writing a certain function (or multiple functions) within a larger program. An outline for each function is always provided in the MP descriptions. The first couple MPs are in LC-3 assembly, with the rest of the MPs being in C. Sometimes there is an MP written in C++. The MPs generally take up to 10-15 hours of work a week, though some students may find them to take much less time. MPs can be done in groups of up to three people.

The class has two evening midterm exams and a cumulative final. At one point, the exams were conducted using a virtual machine. Students were required to bring their own laptop to the exam and it was strongly encouraged that they make sure their VM is working before coming to exams. Due to issues in the past with VMs not being readily available, not functioning properly between exams, and simply becoming burdensome for many students to use, the last few semesters have reverted to using paper for exams.

The majority of the exam is a programming portion, usually multiple (two to three) problems, where students must write some function, similar to the MP's assigned in the class. On occasion the test also includes short-answer questions, ranging from topics discussed explicitly in class to other questions meant to test how students can apply knowledge to different situations. Some short-answer questions may have students analyze code and explain a bug found within the code. Preparation for exams of course depends on the student, but in general it is recommended to have full understanding of the assigned MP's. In the past, there have been practice exams given in the lab section in order to give students an idea of what to expect on the exam. In general, understanding the MP's and going through class notes generally should provide enough preparation. Students should expect to devote at least a few hours before each exam.

A very notable MP for ECE 220 is the sparse matrices MP; this MP has on average taken a vastly greater amount of time when assigned as other MP's. The difficulty of this MP is very much more than all other MPs, so it is worth mentioning that the week of this MP will be dominated by pretty much the MP.
Life After:

This class is likely the first class entirely focused on programming that students will encounter. Many CompE students continue after this class to CS 225, a required CompE course. At this point, students also have the possibility of taking ECE 391, though many tend to wait longer before taking the class. In general, ECE 220 is an important introduction to programming which allows students to advance into other programming-related topics. ECE 385 is the next step in the ECE core curriculum, which applies some programming concepts but does not require extensive use of LC-3 assembly or C.