CS 225 - Data Structures

Instructors:

Cinda Heeren taught the course for several years. She was a very capable instructor who is also very friendly to students. She also tried to keep class interesting, sometimes involving the class in activities not normally seen in traditional "lectures." In general, her explanations were clear and thorough. She always uploaded her lecture slides to the course website, and these were generally good enough to make the textbook unnecessary.

Since Cinda's departure to the University of British Columbia, Wade Fagen-Ulmschneider, Thierry Ramais, and most recently Eric Shaffer have been running the course.

Prerequisites:

Officially, one of either ECE 220 or CS 125 and one of either CS 173 or MATH 213 (to satisfy the discrete math requirement). Since ECE 220 teaches C while CS 125 teaches Java, during the early part of the course the ECE students need to pick up the concept of "classes" and objects while the CS students need to learn how to use pointers. It seems like the students coming from ECE 220 have a slightly easier time, but after a couple of weeks everyone is pretty much on the same page. Although officially it says CS 173 or MATH 213 is required, not having taken either of those two courses doesn't put a student at much of a disadvantage, as many students tend to take CS 173/MATH 213 and CS 225 concurrently. However, the first homework assignment is a review of CS 173 and several skills introduced in CS 173 are needed in CS 225 (induction, inductive proofs, big-O notation). This can be learned fairly easily with a little bit of effort. If a student is taking CS 173/MATH 213 at the same time as CS 225, then induction would already have been covered by start time CS 225 may need it. Note that in recent semesters, the requirement for ECE majors to take CS 225 before this course has been strictly enforced.

From spring 2018 semester forward, the CS department no longer allows concurrent registration in CS 225 and CS 173 / MATH 213. This is due to a lot of the mathematics/proof concepts being pushed to the front of the class while scrapping the C++ introduction portion. According to the CS department this is allegedly too difficult for concurrent registration purposes. The first important point to note is that this is changed should you get some type of A in ECE 220, because this would allow you to survive with concurrent registration (allegedly). Should this not be met, a very possible route is proficiency testing out of CS 173; most find this not detrimental to CS 225 (including Theory Exam 0).

When to Take It:

For EE students using this class as one of the 3 out of 5 courses, it doesn't really matter when CS 225 is taken. However, for CompE students, this class is much more important as it is the gateway to many upper level ECE and CS courses. CS 225 should be taken soon after taking ECE 220, and as mentioned earlier, it can be taken after or in conjunction with Discrete Math. Generally, most EE and CompE students will take this class during their second semester of their sophomore year, but many also take it during the first semester. With the emphasis placed on using software for a variety of work in ECE, this is a fundamental course and will benefit EEs seeking to strengthen their programming skills. Another reason to take this class early is so the student will have an advantage if applying for internships related to software development.

Class Content:

For students interested in pursuing any kind of software development in their careers, the material covered in CS 225 will be extremely important and useful. The class introduces students to many types of data structures such as lists, stacks, queues, and trees, as well as searching and sorting techniques. These are all fundamental concepts that interviewers will test time and time again. Students will learn the inner workings of how to implement these data structures in code as well as learn to analyze its performance using principles from discrete math - namely, big-O analysis. In recent semesters, the class has also taught basic concepts of parallel programming such as how to write parallel code and solve race conditions introduced by parallel programming.

Work:

Students who enjoy programming will love CS 225. The majority of the workload for the class is contained in the form of usually 7 MPs. Each MP is broken up into two checkpoints. If students complete the first checkpoint within 1 week of the MP being assigned, they will receive extra credit. Otherwise, MPs are usually due on Fridays, two weeks after they are assigned. Many of the MPs are really cool, such as compressing an image using a quadtree or building a random maze and then solving it. Students who start early and work at a steady pace will find doing the MPs to be pretty fun. There is also a lab section for this course, and in recent semesters the lab assignments have been mandatory. Usually the labs focus on concepts from lecture such as trees, recursion and parallelism. Labs are graded and due on Sundays but solutions are provided too. Overall, this course requires a fairly large weekly time commitment in relation to other courses in the ECE curriculum, but this could depend on how good at programming a student is.

In the past, the class also had 2 midterms and a final - the midterms tended to focus more on understanding and writing code, while the final had a bit less code and focused a bit more on big-O performance of the algorithms you’ve learned in class. In the most recent semester (Spring 2018), due to the overwhelming amount of students signed up for the class, the exam format has changed to 7 midterms (an "Exam 0", three theory-based exams, and three programming-based exams, all taken in the Computer-Based Testing Facility) and a final exam.

Life After:
As mentioned earlier, CS 225 is a prerequisite to many higher level ECE and CS courses, such as CS 418 (Interactive Computer Graphics) and ECE 448 (Artificial Intelligence). In addition to this academic aspect, there’s a real-world benefit as well: students looking towards software development as a possible career will find that the concepts learned from this class will help greatly in competing for internships in this field.