University of Michigan Fall 2019 Instructor Report With Comments EECS 398-001: Special Topics Nicole Hamilton

15 out of 56 students responded to this evaluation.

Responses to the University-wide questions about the course:

	SA	A	N	D	SD	N/A	Your Median	University- Wide Median	School/College Median
This course advanced my understanding of the subject matter. (Q1631)	8	7	0	0	0	0	4.6	4.5	4.4
My interest in the subject has increased because of this course. (Q1632)	8	5	2	0	0	0	4.6	4.2	4.1
I knew what was expected of me in this course.(Q1633)	2	3	7	2	0	0	3.2	4.4	4.3
Overall, this was an excellent course.(Q1)	3	9	2	1	0	0	4.0	4.2	4.2
I had a strong desire to take this course.(Q4)	8	6	1	0	0	0	4.6	4.0	4.1
As compared with other courses of equal credit, the workload for this course was(SA=Much Lighter to SD=Much Heavier) (Q891)	0	1	4	6	4	0	2.1	3.0	2.8

Responses to University-wide questions about the instructor:

	SA	A	N	D	SD	N/A	Your Median	University-Wide Median	School/College Median
Overall, Nicole Hamilton was an excellent teacher.(Q2)	6	4	4	1	0	0	4.1	4.6	4.5
Nicole Hamilton seemed well prepared for class meetings.(Q230)	2	6	5	2	0	0	3.6	4.8	4.7
Nicole Hamilton explained material clearly.(Q199)	4	6	3	2	0	0	3.9	4.6	4.6
Nicole Hamilton treated students with respect.(Q217)	7	8	0	0	0	0	4.4	4.8	4.8

Responses to additional questions about the course:

	SA	A	N	D	SD	N/A	Your Median	University-Wide Median
Prerequisites provided adequate preparation for this course. (Q61)	5	5	5	0	0	0	4.0	4.3
The textbook made a valuable contribution to the course. (Q64)	1	2	6	4	2	0	2.8	3.6
I developed confidence in my abilities as an engineer. (Q1769)	8	5	2	0	0	0	4.6	4.1
I developed the ability to solve real world engineering problems. (Q1770)	9	5	1	0	0	0	4.7	4.1

The medians are calculated from Fall 2019 data. University-wide medians are based on all UM classes in which an item was used. The school/college medians in this report are based on classes that are upper division with enrollment of 16 to 74 in College of Engineering.



Comment on the quality of instruction in this course. (Q900)

Comments

You had a tragedy that happened this semester that threw you off for a little while, but overall I think you did an excellent job of taking students feedback from previous semester to improve the course and the lectures. Keep doing that every semester and this course will become an excellent MDE course.

Extremely knowledgeable, with relevant real-world experience. Open and willing to help at all times.

Professor Hamilton is extremely knowledgeable, but I struggled to absorb a fair amount of information presented in the course's lectures.

Some of the lectures are definitely still a work in progress, but as a while, the instruction in the course was generally good, especially considering a lot of necessary topics were covered that wouldn't have been covered in 281. The instruction was fine for me because I came in with a lot of background knowledge from other courses, but I feel like some of the material may have been difficult for me personally to understand quick enough to effectively complete the project (still barely seem to be close to finishing).

The content is great but the instruction could be much more effective. Much of lecture time is spent repeating old content. Many lectures are cancelled.

No good evaluation of how we are doing relative to the rest of the class. Could have more checkins and deadlines for major components.

I think Nicole did a wonderful job of giving us a great understanding of the overall project structure and parts. The IA's were able to fill in any of the smaller implementation details that were needed.

Nicole is a unique teacher from my experience, and it took me a few weeks to appreciate her charm. She is very smart and clearly passionate about what she is teaching. She is wonderfully patient and helpful in office hours. However, I found many of the lectures unhelpful both in terms of specifically helping us complete the search engine but also just in general for learning. It was hard to digest the slides that had a lot of code on them.

Overall, the course was amazing. I've learned so much from this class. Lectures were canceled a few times because of a lack of material. Lectures sometimes overlapped and had repeated material, but seeing the same thing more than once made it easier to understand.

I hope the organization of the course can be better next time, especially the frequency of the labs and homework assignment. I hope most of them can be done earlier in the semester so that we have the enough knowledge and time to handle our search engine in the second half of the semester.

The instructor is obviously very knowledgeable. Sometimes that translates to a lot of lectures that go way over my head because we are talking about something that is implemented in bing but will never be implemented in my own search engine.

What were the strengths of the course ? (Q953)

Comments

This course provides an experience I think is lacking amoung MDEs for CS students. A non–niche (Game Dev) systems develop group project that you could

Trial-by-fire learning, exposure to low-level interfaces, high-quality, experienced instructor, interesting project.

This course forces students to strengthen their design skills and programming abilities. The lack of structure in the course proved to be a positive at times, as you are forced to learn on your own.

What I love about this course is the ability to work on a large design project with a lot of flexibility, but with still some restrictions that could come in a real–world corporate environment. It was definitely great practice for work that could take place in a start–up or new–team environment, which I will find very helpful as I enter industry.

the content

The main strength of this course was making the project have the workflow of something you could expect in industry. There were plenty of constraints that we would expect to face in the workforce and having to work around those while still delivering a suitable project was a extremely valuable experience.

Freedom to create something large with multiple teammates. Felt different than a regular class, as a capstone should. Ordered t– shirts, food during end of term presentations, Nicole Hamilton's experience. Guest lectures are cool! Nicole's passion for the subject. The three IAs were all great and definitely added to the course. It was important they were good, and they were.

Nicole has a lot of real world experience and also learns from the students. We've taken a lot of ideas from previous semesters and the IA's have been a great help.

What suggestions would you make for improving the course ? (Q955)

Comments

Keep making improvements to the slides and figuring out when you should introduce what material when in order to maximize the teams productivity.

I think you should make more an emphasis on getting past the surface area as quickly as possible. I really like the idea of setting a deadline of crawling with a passable frontier by mid way through the semester, this way teams have plenty of data to play with in their sandboxes.

Less individual homework and more group work. I feel this class should be about learning to work on a software development team and using homework to emphasize pair program would be cool. Students should be allowed to pair with others on their team in groups of two to three, and you should emphasize that these homeworks are designed to help you work better as a team. I think this could help tremendously with teams that struggle to work together early on, which continues into later in the semester.

I think the lectures where you go over the idea of the code and show us the basic structure of the C library data structures we need to interact with, then provide "starter code" is the optimal way to go. It would be cool if you provided the "starter code" (code like you had for index or query parser where you provided the implementation details but with only stub functions) as files.

More focus on the practical problems of large software design not covered in most courses. Modern exception handling practices, project management techniques (Agile, etc), logging, dependencies, test–driven development, serialization/persistence.

The course material really needs to be presented in a different order with a different time table in mind. 1) Lectures 1-3 = overview of a search engine and explanation of index build components, String + Vector HWK assigned, due in 2 weeks.

2) Lecture 4 = LinuxGet lab

3) Lecture 5-7 = OS concepts, Threads (mutexes, CVs, semaphores), memory mapped & sparse files

4) Lecture 8 = Threading lab

5) Lecture 9–11 = Frontier, Index, Hashing lab or HWK

6) Lectures 12+ ...

I feel that lectures tried to go too in depth and be a jack-of-all-trades with respect to concepts from other courses. I would have greatly appreciated a strong, forced head start on the index build side of this project.

Data structure homeworks should directly follow the interface of the STL as well. This is nitpicky but it was frustrating to work on String & Vector only to have the inferface be different from our team's.

Homework and labs should be optional or for extra credit. While they do definitely help in getting started, they can definitely hinder progress should should they be delayed or contain a lot of restrictive requirements. It's very pointless and counterproductive when they are released after we already implemented them. Personally, I didn't really enjoy the exam for this class, as it seemed highly based on memorization ability that has very little importance to the main concepts learned in the course.

better team formation in the beginning of the class

more deadlines

more checkins

Getting the labs our ASAP would allow for the workflow to be a bit smoother. Towards the end of the project we were finally able to split the workload evenly and it didn't feel like 6 people were working on something only 2 people needed to care about. I would also make this a winter only course as recruiting season and summer slog make it hard to fully put all the necessary time into the product.

I would make a cutoff for homework assignments after Thanksgiving break. By that point, I believe people need to be well on their way to completely all parts of the search engine.

Release the http and hash assignments much earlier in the semester so we can get started faster. Make a deadline to get crawling by. Encourage more people to come to lectures by making them more dynamic/engaging. Learn people's names. Make lectures more useful for the project. I would say bring all of the homeworks and labs closer to the beginning of the semester.

Make homeworks line up more consistently with where we ought to be in the project. I'd also recommend making them optional. For my group we were always ahead of the homeworks so they just slowed us down.

I think making the http/https downloader, hashblob, and multithreading labs due earlier would have helped us plan things better. We ran into a lot of memory issues with our crawler and knowing how to serialize data structures would probably have been useful. I would also change the style guide. It isn't helpful and IDE's have code formatters that are good enough.

I think you should release all the homework assignments and labs together so that the partitioning of work between team members can happen right away.

Make hard deadlines for milestones such as 100,000 sites crawled by October 15. That way crunch time happens a little sooner

Comments

and there is less stress happening later on.

Do away with the assignments that don't help so much for the search engine such as the first homework, the group photo, and the gantt chart.

Among the courses you have already taken, which proved the most (or least) effective in preparing you for this course, and why? (Q1098)

Comments

281, 484 (I first heard about R/W accesses and locking here)

EECS 370 for general low-level understanding.

EECS 485 for understanding web systems and the basics of search engines

EECS 484 for understanding on-disk data structures

EECS 280 and EECS 281 were the most helpful. EECS 280's explanations of memory and pointers prepared me for memory mapped files well. EECS 281 prepared me for reading documentation and using that documentation to write code (cplusplus.com/ for STL features in 281 transitioned to linux.die.net/man/ for system calls in 398).

485 web systems provided a lot of background information that really helped me understand nearly all the concepts covered. 482 Operating Systems was the next most effective as it provided a good background needed for dealing with threads and distributed systems. 281 was obviously useful.

I feel like EECS281 was the most effective in preparing me for this course. Most of the material can be understood even if someone has taken only 281 in their EECS career, and the higher level topics are very digestible.

EECS 485, 370

I think 280 and 281 should have been more helpful, but I didn't learn this material well the first time around, so I was still very unsure of myself when it came to things such as pointers, which was definitely a big part of the project. Esp. with the hash homework.

I've only taken one upper level before taking this class. I think 281 and 370 were good preparation for this class. Some of my team members haven't taken an upper level at all and we were able to produce a functioning search engine. Having taken 482 probably would have helped a lot, but it is not necessary at all. Most of the students in the class are seniors but the class is doable as a sophomore.

The most were EECS 281 and EECS 370. This is because I needed a good understanding of data structures such as hashtables and sets as well as an understanding of how storage and memory work on the computer.