University of Michigan
Winter 2021 Midterm Instructor Report With Comments
EECS 440-001: SysDes Search Engine
Nicole Hamilton

23 out of 93 students responded to this midterm evaluation.

Responses to questions related to the course:

	SA	A	N	D	SD	N/A	Median
I am learning a great deal in this course. (Q966)	12	10	0	1	0	0	4.5

Responses to questions related to the instructor:

	SA	A	N	D	SD	N/A	Median
Nicole Hamilton is enthusiastic. (Q114)	18	5	0	0	0	0	4.9
Overall, this is an excellent course. (Q964)	10	11	1	0	1	0	4.4
Overall, Nicole Hamilton is an excellent teacher. (Q965)	10	9	2	0	0	0	4.4
Nicole Hamilton acknowledges all questions insofar as possible. (Q968)	18	3	0	0	1	0	4.9
Nicole Hamilton uses techniques to foster class participation. (Q972)	14	5	2	1	0	0	4.7
Nicole Hamilton is willing to meet and help students outside class. (Q975)	17	5	0	0	0	0	4.9
Nicole Hamilton keeps students informed of their progress. (Q977)	15	6	0	0	1	0	4.8

Written Comments

What are the major strengths of this class? What is helping you to learn? (Q979)

Comments
Professor Hamilton's flexibility and understanding has been very helpful. The responsiveness to altering homework assignments and focus on main end objective has increased motivation and productivity for the search engine.
Team Colloboration
Cool project. Good guidance with homeworks and lecture.
Help students for their future career. The professor and IAs are extremely helpful.
Professor Hamilton's depth of knowledge, and her understanding of questions that may arise.
I think Nicole is very knowledgeable about the subject material and she's done well to foster a good learning environment for each team. The class has been mostly well–organized so far.
getting some hand–on experience on a realworld–ish project in team environment
It feels like the project and assignments are most helpful. Building a project as big as this is more or less the reason took the class in the first place and it's definitely helping me learn.
Overall, this is a unique class in that there is a lot more freedom for design choice, and it offers a good experience that feels more similar to industry.
Prof. Hamilton is the major strength of this class.
Breadth of content covered in the class (from 482/485/388–style content) and the interesting new topics.

How can Nicole Hamilton improve this class? If possible, give specific examples. (Q980)

Comments
I think assigning a reading schedule (even if optional) would help me keep on top of the book. I know it isn't required for success, but the fact that 482 assigned me readings even though they were optional is what forced me to really try and learn.
She's doing great! (: Love the pseudocode she provides within lectures to help us build our engines!
Meet teams regularly to discuss about the project.
Not much. She's doing the best job I've ever seen at UoM.
I'm not sure how constructive it is to have us implement our own STL functionality, at least not to this scale. I don't think it's practical for teams to spend hours doing these things when better, faster versions of them exist in the STL already. Everyone who has taken and passed 281 is (or should be) fully capable of implementing a vector or string class on their own and I don't think it's necessary to allocate time towards doing something so trivial. I, and many others like me, would strongly prefer dedication to search engine specific content. I feel like discussion and lecture material concerning building the parser (course staff did cover this, so props to you), crawler, index, and other search engine components should be covered early on in order to get students crawling ASAP. Building a string or vector class is not hard, nor is it particularly interesting, so I suggest we spend our time on something more relevant. On a related note, I also think building the front–end server in C++ is completely useless in terms of applicability to our careers. If I'm to build front–end functionality (which is boring in the first place), at least allow us to use industry standard tools like Python/Flask, Django, etc that we may actually eventually work with after graduation. I'm aware that you've said it's easy to do, but this is an capstone course meaning the primary goal is to prepare us for post–academia working environments.
publishing assignments on schedule
I think the shuffling around of assignments (both in due date and details) was understandable given the circumstances but also confusing at times. Otherwise, the class definitely feels like a new class which has its downsides but I think it will grow out of it as it gets taught more.
It might be worthwhile to make the prequesites a little bit stricter. While I understand that the 281 requirement is done in order to make the class more inclusive, this is an MDE class where most of the students are juniors or seniors and have taken multiple upper levels. Having a 482 prereq (for multithreading) or 388/485 prereq (for basic networking) would help streamline the course content – as of right now, the large majority of the lectures (for me) has just been review from previous classes.
Please read the chat more often! I think there are interesting questions raised there but they might not get answered because you don't notice them.

Please enter any additional comments you have for Nicole Hamilton. (Q981)

Comments
– Rather than enforcing your own style guide, maybe it would be better to insist on consistency within the codebase. I think it would be more important that a team be forced to agree on conventions, and then find a way to uphold them. It would be interesting to force students to use tooling like linters and formatters as part of the assignment. – Another idea is to force teams to write and submit comprehensive style guides documenting their internal conventions and how they agreed to it. I think it would reduce some of the frustration students have with your style, and would force students to think about why style matters in larger codebases. – I think it would be cool to do a more in depth case study or two on particular search engines Regarding developing our own versions of STL conveniences: I completely see the purpose of forcing students to understand STL internals. I actually enjoy this part of the project—I like understanding what library vendors do and how they approach optimizations. However, I think there is some redundancy between this class and EECS280/EECS281. In EECS280, we have already implemented our own resizable container, tree structures, and sets. In EECS281, we have implemented our own string with move semantics, an efficient pairing heap, our own unordered_map, etc. I think it makes sense to allow using features that were covered in gory detail in previous courses, and force us to write our own STL features for the more obscure and modern parts of the STL. In particular, I think it would be awesome to primarily focus our re–implementation on smart pointers and threading wrappers. Understanding the internals of these things will greatly help students as they go into industry where modern C++ is expected, and in other languages understanding how features like reference counting and memory safety are truly implemented. By doing this, it will encourage students to focus on understanding not only how to effectively use modern C++ features, but how they actually work under the hood. I've spent a lot of time reading about unique_ptrs... it never really clicked entirely though until I reimplemented it on my own.
Would find it helpful to make it more specific what we need to know —- lots of tangents atm which are interesting but confuse me sometimes when I have less brainpower.
Enthusiastic and strong. Admirable the effort she puts in to this course, despite personal circumstances which may make her feel tired. Thanks Professor Hamilton!
Thank you professor!
Love the class!
Despite my complaints above, I'm loving the class and I think you're a fabulous instructor in spite of your quirks (*cough* *cough* C++ front–end *cough* *cough*). I appreciate your dedication to your job, the enthusiasm you bring to the course, and the positive learning environment that you foster. Thank you!
she is eager to help students.