University of Michigan
Winter 2025 Midterm Instructor Report
EECS 498-012: Special Topics
Nicole Hamilton

20 out of 24 students responded to this midterm evaluation.

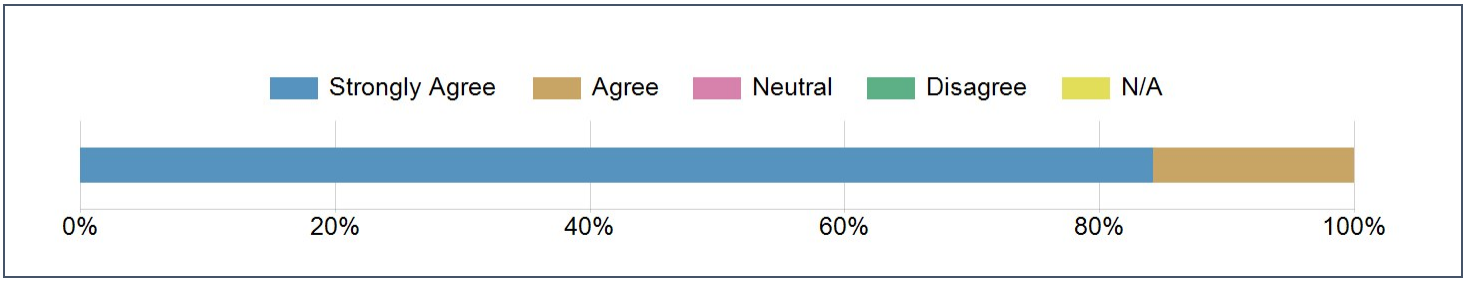
Responses to questions related to the course:

	SA	A	N	D	SD	N/A	Median
I am learning a great deal in this course. (Q966)	15	3	2	0	0	0	4.83

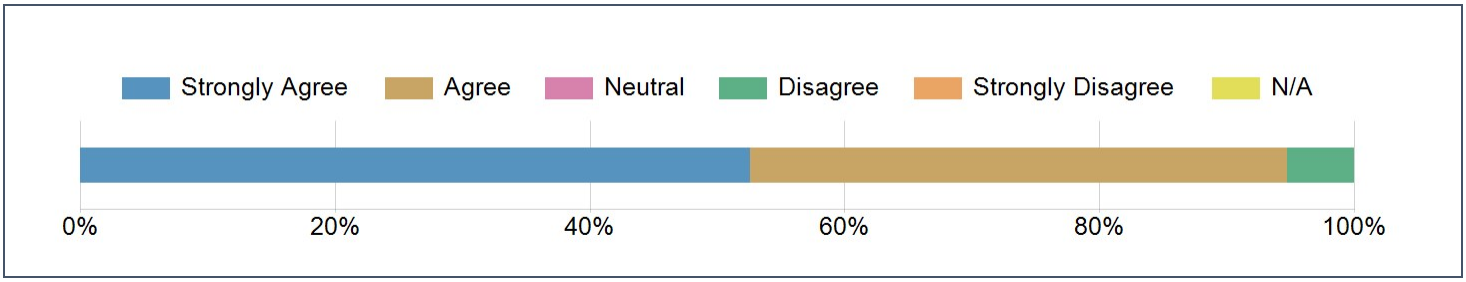
Responses to questions related to the instructor:

	SA	A	N	D	SD	N/A	Median
Nicole Hamilton is enthusiastic. (Q114)	17	3	0	0	0	0	4.91
Overall, this is an excellent course. (Q964)	16	3	1	0	0	0	4.88
Overall, Nicole Hamilton is an excellent teacher. (Q965)	16	2	2	0	0	0	4.88
Nicole Hamilton acknowledges all questions insofar as possible. (Q968)	17	2	1	0	0	0	4.91
Nicole Hamilton uses techniques to foster class participation. (Q972)	14	3	3	0	0	0	4.79
Nicole Hamilton is willing to meet and help students outside class. (Q975)	15	2	1	0	0	2	4.90
Nicole Hamilton keeps students informed of their progress. (Q977)	12	7	1	0	0	0	4.67

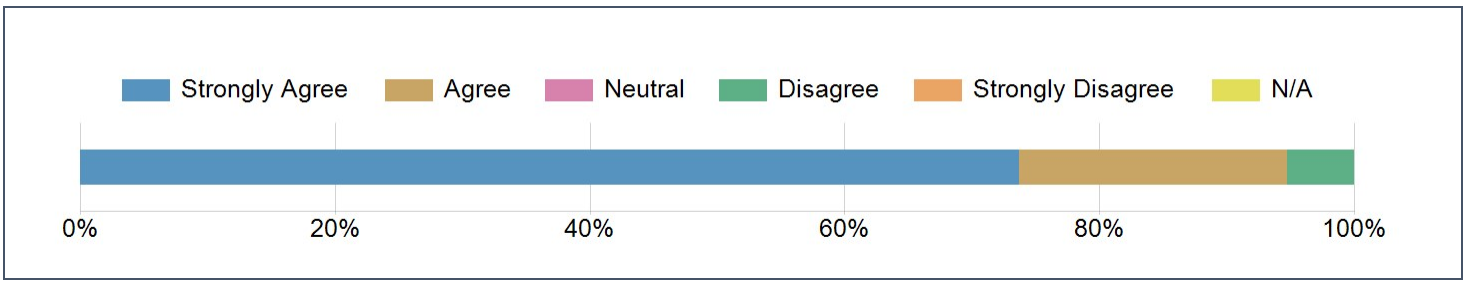
I think this class may help me get a job.



Splitting things up as a team, I don't code everything myself but I'm still learning how everything works.



Working on a team of 6 or 7 is really cool and not like any other class.



Written Comments

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

Comments
Independence and encouragement to build and optimize. We aren't just given an interface and a particular implementation to do.
I feel that one of the major strengths of this class is the professor’s engaging teaching style. They are enthusiastic about the subject, which makes the material more interesting and easier to absorb. Additionally, they take the time to pause for questions, ensuring that students have a chance to clarify concepts. The use of concrete examples to reinforce key ideas is also extremely helpful, as it bridges the gap between theory and application.
I like how hands–on this course is. Lectures are based on real search engines, and we were able to reach Prof. Hamilton's own research paper. The assignments are even cooler, and they make me feel like I'm a software engineer rather than a student.
The lectures are always informative and detailed.
I feel the whole class is teaching me a lot because of its format of being a project based class where we learn by doing.
Large degree of independence
Very close relationship with the professor. I like it very much. I can ask questions anytime I was confused.
I learned more about low level programming in this class than any other course I've taken here. Lecture material is useful.
The mostly unguided projects have helped me learn a lot through the need to browse documentation and consult resources.
The class combines concepts from a wide range of cs topics, like operating systems, networks, and web development, to build a real–world project. The lectures actually go over how the code works which is super helpful instead of just talking about the concepts. The autograder assignments are perfect for making sure our search engine components are working properly. I like that we are learning from a professor who has really built a search engine and has extensive knowledge of the topic.

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

Comments
To be frank, a lot of the "good code" examples, while still relevant in a general sense (naming, verbosity, etc.), are pretty dated. Since this class is billed as a C++ project class, then I think there should be some focus on modern C++ development which does not really exist in most courses here at Michigan. Eg how to set up and use GoogleTest, misc C++ language features, etc.
I believe that a student with only EECS 281 experience might still face some challenges. While the class provides a solid foundation for most topics, it occasionally covers certain concepts too quickly or at a surface level, which could be difficult for a sophomore who hasn't encountered them before. Additionally, though not specific to this class, I feel the exams can be somewhat challenging in terms of knowing exactly what is expected in responses. I think even students with a solid understanding of the material may still struggle to score well due to the open–ended nature of short–answer questions, which allow for a wide range of possible answers. Maybe ~~ providing a rubric or answering guide in advance could help clarify expectations?
Add checkpoints for the actual search engine as an assignment. Maybe we need to submit a short demo as a midterm alternative.
There could be a little more standardization to help teams stay on track. I feel it is very easy to get distracted with other course work and find yourself behind on the search engine.
N/A
I feel like the process of forming groups could be made smoother. For example, maybe during the first/second week of class, a Google Form survey could be sent out to the class to gauge which students – a) know absolutely nobody and are looking for a group – b) know a couple of other students and are looking for other students to fill in the rest of the empty slots of the group or join another group – c) already have a predetermined group of friends that they're working with and are not taking anyone else – d) are tentatively considering dropping the course and still want to shift their schedule around And the results of this survey could be shared with the whole class. This way, students can see where everybody's at and not waste time/effort in meeting people of category c/d. And this survey can have extra information like classes previously taken, the current schedule/workload of the student, availabilities, etc.
Maybe include reading material for every week so that we can learn a bit before lecture.
I feel like Professor Hamilton can maybe try to make the lectures a bit more engaging given the small class size, as sometimes it feels like trying to figure out a wall of text (code).
I disagree with the requirement to rewrite the C++ library data structures like vector and string. I understand it's useful to our understanding, but if the class is supposed to mimic a real–world design project nobody in the real–world is rewriting the C++ library. In my opinion it would be better to let students focus on the search engine system components with the additional time.

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

Comments
None
Love this class, and I'm excited to finish building an actual search engine.
Overrall probably one of my favorite professors I have had during my time here at UM. Probably the most engaging professor I have ever had. Would 100% take another class with her.
Great class!
I really like professor Nicole!
One of the best professors I have had at UM
N/A

What's been the most interesting part of the class so far?

Comments
I actually really enjoy some of the "history of the web" anecdotes that are put in throughout lectures sometimes; it helps for lectures not to be too technical and understand why things came to be the way they were.
Reading Prof. Hamilton's research paper was super cool. Learning about the index structure was really interesting as well.
Putting os principles to work to do real tasks. (As opposed to 482 which was incredibly interesting but the projects felt somewhat artificial)
The complexity of the search engine. I think it is so cool to be able to build such a large scale project which has so many moving parts I feel like I am actually applying my knowledge from other classes towards a real life project.
Writing code to serialize information to disk and interact with it through memory mapping
Group photo!
Learning low–level c things like opening files and creating sockets
Learning about search engine theory.
Learning about the structure of the inverted index and the strategies that are used to match pages and rank them have been the most interesting.

Which of the autograded assignments have been most helpful?

Comments
HashTable and Multithreaded file operations
Hash Table & Hash Blob were most helpful because now I understand how to optimize/compress key data structures.
Hash blob/ hash file
The HTML parser assignment.
Hashblob/hashtable
All of them were great. I think the selection is fantastic and allows one to actually implement some of the rudimentary versions of data structures/algos that are applicable to the final project, but also serve as great CS general knowledge.
HTML parser? I guess so.
HTML Parser
Hashtable and hashblob.
Out of the ones I've worked on, the HTML parser and the multithreading operations were very helpful.