



Instructor with Comments Report

2018-04-04 - 2018-04-18 Report ID: MSR04734

Instructor: Hamilton,Nicole

EECS 398 002

	Responses from your Students**							Other Users of This Item*			School/College		
	5	4	3	2	1	Your	University Wide			75%	50%	25%	
	SA	A	N	D	SD	NA	Median	75% Above	50% Above	25% Above	75% Above	50% Above	25% Above
4 I had a strong desire to take this course.	12	7	0	0	1	0	4.67	3.67	4.17	4.67	3.64	4.12	4.55
891 As compared with other courses of equal credit, the workload for this course was (SA=Much Lighter, A=Lighter, N=Typical, D=Heavier, SD=Much Heavier).	0	4	7	8	1	0	2.64	2.83	3.12	3.46			
1631 This course advanced my understanding of the subject matter.	4	9	3	3	1	0	3.83	4.12	4.50	4.75			
1632 My interest in the subject has increased because of this course.	7	5	4	1	3	0	3.90	3.79	4.25	4.67			
1633 I knew what was expected of me in this course. (SA=Almost Always, A=Frequently, N=Sometimes, D=Occasionally, SD=Hardly Ever).	0	2	6	4	8	0	2.00	4.02	4.40	4.71			
230 The instructor seemed well prepared for class meetings. (SA=Almost Always, A=Frequently, N=Sometimes, D=Occasionally, SD=Hardly Ever)	0	3	6	5	5	1	2.40	4.52	4.81	4.93			
199 The instructor explained material clearly. (SA=Almost Always, A=Frequently, N=Sometimes, D=Occasionally, SD=Hardly Ever)	1	12	2	2	2	1	3.79	4.30	4.70	4.88			
217 The instructor treated students with respect.	7	9	2	0	1	1	4.22	4.68	4.87	4.95			
1 Overall, this was an excellent course.	2	4	5	5	4	0	2.70	3.90	4.33	4.75	3.59	4.14	4.58
2 Overall, the instructor was an excellent teacher.	5	6	2	3	3	1	3.75	4.38	4.75	4.92	4.13	4.58	4.80
3 I learned a great deal from this course.	5	4	4	4	2	0	3.38	4.00	4.40	4.75	3.83	4.25	4.59
61 Prerequisites provided adequate preparation for this course.	1	7	6	3	3	0	3.17	4.04	4.29	4.55			
140 I deepened my interest in the subject matter of this course.	6	7	3	1	3	0	3.93	3.99	4.31	4.67			
201 The instructor gave clear explanations.	2	10	3	1	3	1	3.75	4.25	4.67	4.86			
203 The instructor stressed important points in lectures/discussions.	5	6	2	4	2	1	3.75	4.42	4.74	4.88			
207 The instructor appeared to have a thorough knowledge of the subject.	10	5	2	0	2	1	4.55	4.69	4.87	4.95			
216 The instructor acknowledged all questions insofar as possible.	7	10	2	0	0	1	4.25	4.50	4.75	4.88			
218 The instructor encouraged constructive criticism.	9	6	2	0	1	1	4.50	4.38	4.67	4.84			
228 The instructor followed an outline closely.	1	3	3	5	6	2	2.10	4.13	4.57	4.79			
229 The instructor used class time well.	2	3	4	5	5	1	2.40	4.29	4.69	4.88			
232 Work requirements and grading system were clear from the beginning.	0	2	4	4	10	0	1.50	4.00	4.40	4.67			
239 The amount of work required was appropriate for the credit received.	3	8	6	2	1	0	3.63	3.90	4.17	4.52			
240 The amount of material covered in the course was reasonable.	2	7	4	5	2	0	3.25	4.00	4.33	4.61			
318 Writing assignments seemed carefully chosen.	0	3	4	6	3	4	2.33	3.83	4.17	4.56			
340 The textbook made a valuable contribution to the course.	1	6	2	6	3	2	2.50	3.28	4.00	4.50			
356 Examinations covered the important aspects of the course.	1	4	8	4	3	0	2.88	4.10	4.36	4.69			
365 Grades were assigned fairly and impartially.	1	4	8	2	4	1	2.94	4.00	4.33	4.70			
366 The grading system was clearly explained.	0	1	5	7	6	0	2.00	4.00	4.42	4.70			

Written Comments

900 Comment on the quality of instruction in this course.

Student 1

Date Printed:4/30/2018 7:47:13 AM



Instructor with Comments Report

2018-04-04 - 2018-04-18 Report ID: MSR04734

Instructor: Hamilton, Nicole

EECS 398 002

Given that this is the first offering of this course, it was very well done. When we think about there wasn't any previous material for the professor to utilize it was normal for her to struggle when creating content. That being said, it definitely has room for improvement. The topics were instructed very skillfully and comprehensively. I got most of the topics in class, however I have to add showing code in class briefly without going over the code in extensive detail was really hard to understand at the moment. Most of the time I got the general idea of what the code must do, but I had no idea how it did it, I needed to trace the code line by line over the next week to get how it does what it does.

Student 2

NA

Student 3

Hamilton is an engaging teacher and it is evident that she cares very much about the course. Her issue is that she was overwhelmed preparing an original class by herself. There were two categories of lectures - lectures on the components of the search engine and lectures on making system calls in c++. The lectures on the search engine were clear enough for a first go and advanced my understanding. The lectures on system calls were based entirely on code and were not effective in my opinion. In general code has to be understood by reading it on your own with the man pages, or it needed to really be broken up line by line and stepped through.

Student 4

Quality was fine.

Student 5

The instructor made great efforts to meet the course needs, despite the lack of ready course materials. Overall, the course allowed us to get a lot of practice with applying very crucial programming concepts. For comparison, most other courses cover those concepts largely in theory or in implementation, not letting students practice the application of the concepts.

Student 6

NA

Student 7

Prof. Hamilton was good at explaining topics, and was really helpful answering questions in office hours. However, this course was very disorganized and not planned very well. I anticipated a little bit of fluidity in a new course, however, I felt the lack of organization was more than what was tolerable.

Overall her quality of teaching was good, but quality of organizing lesson plans, curriculums, lecture slides, and rubrics was sub-par.

Student 8

Grading was extremely vague. Rubric's for assignments and the midterm seemed arbitrarily chosen on the spot. Instructor was (self-admittedly) unprepared for lecture frequently. Having slides to just read over linux man pages or go over code copy-pasted from online documentation was not a very helpful use of lecture time.

Student 9

NA

Student 10

Overall, Nicole is a great instructor! While a lot of her slides were heavily based on code she had written and explaining her choices/how the code works, she was able to convey a lot of quality information that I doubt I would have learned on my own. I know another person in the lecture was pushing for more graphics and such to explain things conceptually, and then perhaps a few code snippets to demonstrate how it looks in practice, and I agree a lot with that stance, just so those who find it hard to learn from code find lecture useful. That said, I recognize that the slides were not like that the first time around because the slides had to be generated, and generating the code for the concept was easier and clearer than trying to generate graphics for, say, deadlocking of threads.

Student 11

Nicole came to the class with a good sense of energy and was willing to always explain the material. Understandably, she didn't have any content at the beginning but didn't really take feedback on what students wanted to learn + what would be helpful rather than just seeing code.

Student 12

This class was not well taught at all. The professor came to almost every class unprepared and admitted as much herself. The lectures usually did not fill up the entire 2 hours. Several classes were cancelled because she could not come up with enough content.

Date Printed: 4/30/2018 7:47:13 AM



Instructor with Comments Report

2018-04-04 - 2018-04-18 Report ID: MSR04734

Instructor: Hamilton, Nicole

EECS 398 002

The lectures themselves were very poorly run. The same topic was often covered for several lectures in a row, with the same slides being copy pasted into each presentation. Usually, more than 50% (sometime nearly 90%) of the slides were just code on the screen which would be walked through line by line. This is incredibly boring and hard to pay attention to. There were very few diagrams pictures, etc. Just slide after slide of code. It is extremely difficult to gain a conceptual understanding of a system by just looking at code.

We were not given a date for the midterm until just a couple weeks before it. We were given very little information about what would be on the midterm (so we did not know how to study for it). The reason for this became clear when she told us she finished writing the midterm just one hour before we took it.

The actual grading of the search engine project was never made clear and it still, to this day, unclear. Grades for assignments took more than a month to come out. Overall the professor did not seem to put much effort into this course.

Student 13

Very high quality instruction. Despite the lack of structure and pre-existing materials, instructor did an excellent job of thoroughly covering all topics. Instructor could and was willing to answer any questions about anything and respond to student feedback about what we didn't understand or wanted to cover more.

I did feel that at times the material was over most peoples' heads and that when the instructor asked if we had questions we were mostly silent because we had not even begun to parse the information and it was taken for granted that we were familiar with some topics that we had never seen. Especially regarding Operating Systems topics. I really enjoyed the class I think almost because of the extension of the ideas I had learned in OS, but I felt that probably this made the class unduly challenging to those without OS experience. The technical part of the exam even seemed to just be an extended OS question that mimicked pretty closely an OS exam. Was refreshing for me but I can't speak for my classmates.

Student 14

I would have liked to see more C++ design patterns explained in class rather than linux vs windows apis. Students usually can learn from reading apis themselves.

Student 15

NA

Student 16

Grading system was unclear and it felt like rubrics were determined on the spot. Instead of awarding points for correct answers it seemed like the professor subtracted points for design choices she did not agree with. Grading was rather subjective overall. Lecture time was sometimes used very poorly. Lectures consisted of going over linux code in which I felt was not very helpful. There seemed to be a lack of clear direction within this course and it showed within lecture. The professor occasionally did not seem prepared for lecture as we would sometimes talk about content that did not feel related to the class.

Student 17

NA

Student 18

Professor Hamilton is an incredibly smart and energized professor. She does a great job of explaining concepts and treated everyone with respect. She was always open for questions and made herself very available during office hours. I hope that she continues to teach here at U of M and think that she could benefit from an IA if this class is taught in the future. I also find it very refreshing to have an EECS professor who has worked in industry and not research, as U of M tends to push the research route more than industry in the CS department.

Student 19

The quality of instruction in the course was by far the worst out of any upper level EECS course that I have taken. Instead of teaching any conceptual material, most lectures consisted of the instructor more or less reading source code a program, written both as a Windows and UNIX version, which did not advance my understanding of the material at all.

Student 20

Extreme lack of preparation

Written Comments

Date Printed: 4/30/2018 7:47:13 AM

Page 3 of 9



Instructor with Comments Report

2018-04-04 - 2018-04-18 Report ID: MSR04734

Instructor: Hamilton, Nicole
EECS 398 002

911 Please comment on the quality of the course as a whole.

Student 1

The course was unique and a great experience for me. The fact that we were responsible for a single project throughout the whole semester and we had a lot of freedom on what we could do was an exceptionally freeing experience compared to other classes. It thought me a lot. However, I have to say the name of the course, 'System Design in C++' is a little misleading in this case. The content of the course was much closer to 'How to build a search engine using C++'. We used 95% of our time worrying about how to do a specific task that a search engine is responsible for rather than the system design aspect. We also got little to no feedback on how we should be designing the system or anything specific related to that. I took EECS 381 couple semesters ago, and possibly going over couple design models like MVC, singleton objects etc etc could be a good addition for this course and teaching people about how they should think when they are designing a system from scratch.

Student 2

NA

Student 3

So overall the lectures were not worthwhile, however, the project for the course, building the search engine through out the semester was one of the most enjoyable projects I've done at U of M. We were given the freedom and responsibility of designing and implementing the search engine on our own schedule, with two midpoint check ins with Professor Hamilton.

Student 4

Course needs to be planned out and flushed out a little more.

Student 5

I think this was an excellent course; I was able to practice the way the low-level C++ concepts taught in 280, 281, and 482 are applied in the real world, on large projects. I don't think any other course at the University offers that kind of exposure. In addition, I think that 482 is the first course where I had to deal with designing complicated systems. It makes me really think critically about the problem, and that's why it helped me become a better programmer. This course, I believe, picked up right where 482 left off and took my analytical skills another step further, making me consider a lot more factors when designing a solution. In general, I think my programming skills improved a lot as a result of this course. I am very glad I took it.

Student 6

NA

Student 7

I think the best way to explain the quality is to just list some of my grievances:

-I thought the homework, although good coding exercises, were not even remotely relevant to what we were learning in class. And the goal of homework is NOT to check a box, but to help students learn and understand course material.

- There were only 2 homeworks for our entire contribution, so there was no opportunity to make up for a bad score

- Some of the grading on the homework seems like it didn't make sense. For example, in homework 1 we had to find the greatest sum within some sub array of ints. If there was a tie, you got points for picking the larger one. But we weren't told, this is completely random. Some cases might need shorter or large intervals, or more to the right or left, but without a rubric or context it makes no sense to grade on this.

- The exam had one question that was 80% of the exam, and key piece to the question is multi-threading a program. we did not LEARN how to write a multithreaded program. We were SHOWN how to spawn a thread to print "hello" and that was it. No practice, no review, maybe 4 slides on a question that was 80% of our exam. This gave anyone who took 482 a MAJOR advantage (a class that is not a prereq). If this had been a homework assignment, I'd think it was fair, but if you hadn't taken 482 or worked on the crawler you had no chance. Also, I was marked down more points for ATTEMPTING to thread my program, than my friends who ignored the threading portion of the question. I had a friend who didn't thread, and didn't perform word-count (the two tasks of the question) and got less points taken off for those two topics. Absolutely ridiculous.

-the project outline wasn't clear, but gave us room to be flexible. I think giving points for making something interesting would have been way for useful. E.G. allow us to make a really good search engine for a website ESPN, or something, and then get evaluated on that.



Instructor with Comments Report

2018-04-04 - 2018-04-18 Report ID: MSR04734

Instructor: Hamilton, Nicole

EECS 398 002

- I think the LOC evaluation metric made people EXTREMELY GREEDY, writing useless code and trying hard to write useless functions just to scrape out more code.

-Although we haven't gotten scores back for projects or the final, I have no idea how I will be graded, or what would make a good project. Anywhere from an A+ to a C- wouldn't surprise to be honest.

Student 8

This course did not teach reusable concepts about system design that could be applied to engineering work/projects outside the realm of search engines. In place of learning about actual design patterns, the class focused more on how to implement a search engine and read/write an index.

Student 9

NA

Student 10

The course was great in that it changed my perception of larger, more complex coding projects- they aren't necessarily harder than smaller ones, they just take more time and have more pieces. That said, I did not enjoy how much uncertainty that was a part of this course. Not knowing what is important on the midterm was very stressful, as I was not sure if I should have read more of the textbook/be more familiar with different ways that search engines are implemented or if I should be more familiar with multithreading, or how the index works, or really anything from the wide array of topics we had touched on. In addition, having the semester long project being competitively graded is an interesting choice. While I understand it for the first iteration of the course as no students have built any before, I hope that future iterations of this course abandon that model in favor of a set #of points per feature. This way, if one or two groups have people who are part time students or just with easier semesters and thus can implement more features/make a more polished experience, other teams are not penalized for that.

Student 11

Interesting idea of a class, alot of room for an amazing, real world experience. However, lacked full realization. Could work eventually but would be hard to scale without TA's or additionally help.

Student 12

I had high hopes that this course would teach me about how to design large systems and advance my knowledge of C++. I expected it to be like EECS 381, but with less of a focus on the C++ language itself and more of a focus on design. However, I learned nothing about system design and even less about C++.

The entire class was focused on building a search engine. We were not taught about how to go about coming up with a design at a high level, how different parts should work together, what parts of a module to expose, etc. Instead we were just taught what the different parts of a search engine are. It was not taught very clearly either. We ended up just reading the textbook which was much clearer than anything taught in class. The class should not be called "System Design in C++", it should be called "Build a search engine".

My knowledge of C++ did not improve due to this course. We were discouraged from using the STL or any libraries at all (although the code examples from the lectures always used the STL). If you aren't using the STL or libraries, why are you even using C++? Might as well teach the course in C. This outdated way of thinking did nothing to help me prepare for my job in industry where I know from my internship experience that I will be using the STL and libraries every day.

Overall, this class was a massive waste of my time and I regret signing up for it.

Student 13

As a whole I thought the course was really fun and a really great idea for a course. This is by far one of the most useful courses I have taken in terms of the experience it provides and the amount it has improved my knowledge of programming concepts and proficiency in C++. And learning about building a search engine has taught me so many things that have carried over to job interviews and the understanding of how many common fields/software work.

Really enjoyed learning details about the Software industry world. Think learning about Development process in this course was a really strong plus. Its rare to have an instructor with so much insight in this respect.

Student 14

I thought this course would teach me c++ language fundamentals as well as search engine aspects. I did not learn as much c++ as I would have liked to.

Student 15

NA



Instructor with Comments Report

2018-04-04 - 2018-04-18 Report ID: MSR04734

Instructor: Hamilton, Nicole

EECS 398 002

Student 16

This course is called system design in c++. This was not a course in system design in c++. It was more of doing a big project in c++ than learning anything about c++ or system design. I felt that in terms of search engine knowledge there might have been something unique, but a class dedicated to information retrieval such as EECS 486 would be better for breadth and depth. I believe that the course did not teach any reusable concepts in system design and it is very hard to take this experience and apply it in industry.

Student 17

For a first run of the course, I was quite impressed with the content. Grading seemed very subjective, but also fair.

Student 18

I learned a lot in this course. I think everyone should take a course like this who is considering going into the industry as a software developer. I think that more structured guidelines of what was expected, more valuable homework, and more in class presentations (mid term presentations etc) would have drastically improved my experience. There were many times when I was unaware of expectations and what the timeline was of the course. Overall I feel a great sense of pride and accomplishment finishing the course and feel more confident in myself as a developer.

Student 19

The quality of the course was subpar compared to other upper level EECS courses both due to the instructor's lack of preparedness and the lack of planning with respect to assignments. I learned nothing new about either system design or C++ in this class that I did not learn in other classes.

Student 20

Very dissapointing

Written Comments

931 Please give any other comments on this course as a whole.

Student 1

I strongly believe this course has a great future. It can be refined and structured as time goes by. With its current form its hard for it to be liked by everyone, but with a little bit of a structure without taking away too much from the freedom given to students with the nature of the project it has a great potential.

Student 2

NA

Student 3

It's not quite clear to me how the final grades for the class will be generated and whether they will be fair. The midterm for the class had two components, a series of short answer questions and a long coding question. The short answer section was remarkably refreshing for a computer science exam in that it actually tested general comprehension of course material. The long answer section was perhaps a good test of course material as well, except that it was about 2-3 times as long as could be finished in the time given. A test that is too long, cannot be graded fairly since your score depends on what you prioritized attempting and how fast you can write code. However, I don't think the midterm will be weighted too heavily, so this isn't a huge issue. The final is an open ended essay. I don't know how papers will be objectively graded or how much it will be worth. However, my understanding is that the largest portion of the grade will be from effort put into the search engine. If this is the case, that would be the fairest way to determine grades. An additional comment I have to make is there were two coding assignments, a short problem at the very start of the semester and hashtable that would later be incorporated into the project. The rubric was somewhat arbitrary and too heavily dependent on style versus functionality. In conclusion, I did get something from the course, mostly from the project. I recognize how difficult it is to put a course together from scratch. The course was very roughly put together and I don't know how much can be attributed to it being the first offering of the course.

Student 4

NA

Student 5

As great as the coding samples are, for future terms the course is offered, it would be very helpful to have more graphical examples of the concepts covered in class. The search engine examples were extremely helpful for understanding the material. It would be great to have those also for the OS concepts covered in class. Of course, another solution to that would also be to make 482 a pre-requisite and then give more homeworks related to using the OS concepts.



Instructor with Comments Report

2018-04-04 - 2018-04-18 Report ID: MSR04734

Instructor: Hamilton, Nicole

EECS 398 002

Student 6

NA

Student 7

I thought this was a good attempt, but honestly just really disorganized. And I think a lot of things we went over were just overcomplicated things that are learned in 486. I think if we focused more on system architecture and design, and how to build a big project it wouldve been better.

I came into this class hoping to learn about system design, structuring big complex projects, the software design cycle, good testing, and good object oriented design practiced. But I really didnt learn any of this.

I learned how a search engine works (really cool), and how to access low OS stuff. thats pretty much it.

I really hope Prof Hamilton gets another shot at this class. Structure the class before hand, make rubrics and guidelines. Be flexible and add a TA, and this is a Umich caliber class.

Student 8

NA

Student 9

NA

Student 10

I appreciate how candid Nicole was with all of us and how she just didn't lie about what was going on. I also really enjoyed all of the content and I really hope this course is offered again, as I believe it is a great addition to anyone's computer science education. This course is a unique blend of other courses with a lot of original material and twists on topics that provide for a classroom experience I doubt I can get elsewhere. The only other comment I have comes with regard to the use/lack thereof of the standard library. I really do not understand why one would go about re-implementing a stack, especially if it doesn't do anything special. I would have loved to hear an explanation for that other than "I just don't like the STL" as I am sure there is a good and valid reason that explains why the STL is thoroughly discouraged from our projects.

Student 11

Each major component should be paired with a major coding topic. Crawler-> multi thread etc. Think it is useful to learn how to work on something ambitious and unknown like this. Most students won't ever come out of school with this type of experience.

Student 12

This class should be shelved until it can be revamped to actually teach about system design and C++. Otherwise it should be treated as a failed experiment and be binned.

Student 13

NA

Student 14

NA

Student 15

NA

Student 16

NA

Student 17

I really enjoyed this course and hope to see it again, even if I can't take it.



Instructor with Comments Report

2018-04-04 - 2018-04-18 Report ID: MSR04734

Instructor: Hamilton, Nicole

EECS 398 002

Student 18

As this was an experimental course being taught for the first time, I expected a certain amount of unstructureness. I think that lesson plans could have been outlined better, with a better flow of high-to-low level discussion of topics. Lecture slides with only code as references doesn't help me personally understand topics well. I also felt that the midterm exam preparation could have been better -- I had no idea what the exam would consist of and nothing to practice besides lecture slides.

Student 19

NA

Student 20

I think that this course was a waste of time and money, and the lack of preparation by the instructor was incredibly disrespectful to the students

Written Comments

1098 Among the courses you have already taken, which proved the most (or least) effective in preparing you for this course, and why?

Student 1

EECS 381, the most effective

Student 2

NA

Student 3

The project relies very heavily on eeecs 280 topics. Eeecs 370 helped a little, but not totally necessary.

Student 4

EECS 281, EECS 482

Student 5

From the courses I've taken, 482 provided both the most relevant theoretical and practical background.

Student 6

NA

Student 7

482 482 482 482 482 482 482 482 482 (havent taken)

Student 8

EECS 482 was helpful because the instructor seemed to require knowledge/the ability to use threading after a single lecture on threads/processes despite the class not being a prerequisite.

Student 9

NA

Student 10

281 as it gave me all the data structure knowledge I needed

Student 11

281->algorithmic thinking

MDE/research project -> how to think about scale, OO, modularity, extensibility

Date Printed:4/30/2018 7:47:13 AM



Instructor with Comments Report

2018-04-04 - 2018-04-18 Report ID: MSR04734

Instructor: Hamilton, Nicole

EECS 398 002

481 ->testing , documentation

Student 12

482 taught me about multithreading which was useful.

Student 13

Operating systems seemed to be super helpful. As I mentioned, the exam seemed eerily similar to an OS exam in the technical portion.

Other than that, of course EECS281 - Algorithms and Data Structures was the most important. That class is what taught me how to design data structures and efficiently do stuff like retrieval and sorting based on the different datastructures. Was super helpful in designing the structures and implementing necessary functionality.

EECS484 - databases was kind of useful too. You learn about datastructures on disk and indexing in that class and I applied some of what I learned in the design and implementation of the search engine.

Student 14

281 most effective

Student 15

NA

Student 16

NA

Student 17

NA

Student 18

This class should require 482 and if not go into more depth and explanation on threading concepts. Possibly include homework assignments where everyone can learn threading. I felt that the people on my team who took 482 were the only ones who fully understood that part of the course. On another note, I took 486 with concurrently with this class and didn't experience much overlap.

Student 19

Most effective - EECS 482, EECS 489

Student 20

482, and if this course continues to be taught this way then it should be required

* The quartiles are calculated from Winter 2018 data. The university-wide quartiles are based on all UM classes in which an item was used. The school/college quartiles in this report are based on upper division classes with an enrollment of 16 to 74 students in College of Engineering.

** SA - Strongly Agree, A - Agree, N - Neutral, D - Disagree, SD - Strongly Disagree, NA - Not Applicable.