UNIVERSITY OF MICHIGAN
College of Engineering
Curriculum Committee Meeting
Tuesday, January 28 – Email Vote
Email Vote

Attending:

Call to Order:

Adjourned:

AGENDA

1. 1.14.20 Meeting Minutes: APPROVED

CARF SUMMARIES

PAGE	SUBJECT	COURSE #	ACTION	SUMMARY	EFFECTIVE TERM	MIN. GRADE REQ. FOR ENF. PREPREQ	APPROVED	NOTES & REVISIONS	TABLED
3	EECS	440	NEW	Modification of course description	FA 2020	C	X	No suggestions made	
25	EECS	494	MOD	Change to enforced prerequisite	FA 2020	C	X	No suggestions made	

UNIVERSITY OF MICHIGAN
College of Engineering
Curriculum Committee Meeting
Tuesday, January 28 – Email Vote
Email Vote

Attending:

Call to Order:

Adjourned:

AGENDA

1. 1.14.20 Meeting Minutes: APPROVED

CARF SUMMARIES

PAGE	SUBJECT	COURSE #	ACTION	SUMMARY	EFFECTIVE TERM	MIN. GRADE REQ. FOR ENF. PREPREQ	APPROVED	NOTES & REVISIONS	TABLED
3	EECS	440	NEW	Modification of course description	FA 2020	C			
25	EECS	494	MOD	Change to enforced prerequisite	FA 2020	C			

UNIVERSITY OF MICHIGAN
College of Engineering
Curriculum Committee Meeting
Tuesday, January 14 – Email Vote
Email Vote

Attending:

Call to Order:

Adjourned:

AGENDA

1. 11.19.19 Meeting Minutes: APPROVED

CARF SUMMARIES

PAGE	SUBJECT	COURSE #	ACTION	SUMMARY	EFFECTIVE TERM	MIN. GRADE REQ. FOR ENF. PREPREQ	APPROVED	NOTES & REVISIONS	TABLED
5	ROB	502	NEW	change to enforced prerequisites	FA 2020		X	Suggested edit to allow course to count for non-Rackham graduate credit	
17	ROB	511	NEW	change to enforced prerequisites	FA 2020	-	X	Suggested edit to allow course to count for non-Rackham graduate credit	
104	MECHENG	395	MOD	change to enforced prerequisites	FA 2020	C-	X	Change pre-req listing from “382” to “MECHENG 382”	
107	MECHENG	450	MOD	change to enforced prerequisites	FA 2020	C	X		

Course Approval Request Form

Office of the Registrar, University of Michigan

1210 LSA Building
500 S. State Street
Ann Arbor, MI 48109-1382
Phone: 734.763.2113
Fax: 734.936.3148
ro.curriculum@umich.edu
ro.umich.edu

☑ **CHECK APPROPRIATE BOXES FOR ALL CHANGES**

Action Requested
- ☑ New Course
- ☐ Modification of Existing Course
- ☐ Deletion of Existing Course

Date of Submission: 2019-11-25
Effective Term: Fall 2020

☑	Course Offered ☑ Indefinitely ☐ One term only	**RO USE ONLY** Date Received: Date Completed: Completed By:

	CURRENT LISTING	**REQUESTED LISTING**
☑	Dept (Home): Subject: Catalog:	Dept (Home): Elec Engin & Computer Sci Subject: EECS Catalog: 440
☐	☐ Course is Cross-Listed with Other Departments	☐ Course is Cross-Listed with Other Departments
	Department / Subject / Catalog Number	Department / Subject / Catalog Number
☐		
☑	Course Title (full title)	Course Title (full title) System Design of a Search Engine
☑	Abbreviated Title (20 char)	Abbreviated Title (20 char) SysDes Search Engine
☑	Course Description (Please limit to 50 words and attach separate sheet if necessary) Design and development of a search engine. Topics include search engine internals, OS facilities, file systems, sockets, and threads. Students work in teams to write a complete multithreaded internet search engine from scratch.	
☑	Full Term Credit Hours Undergraduate Min: 4 Graduate Min: Undergraduate Max: 4 Graduate Max:	Half Term Credit Hours Undergraduate Min: Graduate Min: Undergraduate Max: Graduate Max:
☑	Course Credit Type Undergraduate Student	
☐	Repeatability ☐ Course is Repeatable for Credit Maximum number of repeatable credits:	☐ Course is Y graded ☐ Can be taken more than once in the same term

Subject:	Catalog:

☑	Grading Basis ☑ Graded (A – E) ☐ Credit/No Credit ☐ Satisfactory/Unsatisfactory ☐ Pass/Fail ☐ Business Administration Grading ☐ Not for Credit ☐ Not for Degree Credit ☐ Degree Credit Only	Add Consent ☐ Department Consent ☐ Instructor Consent ☑ No Consent	Drop Consent ☐ Department Consent ☐ Instructor Consent ☑ No Consent

	CURRENT LISTING	**REQUESTED LISTING**
☐	Advisory Prerequisite (254 char)	Advisory Prerequisite (254 char)
☑	Enforced Prerequisite (254 char) Minimum grade requirement:	Enforced Prerequisite (254 char) EECS 281 Minimum grade requirement: C
☐	Credit Exclusions	Credit Exclusions

☑	Course Components ☑ Lecture ☐ Seminar ☐ Recitation ☑ Lab ☐ Discussion ☐ Independent Study	Graded Component ☑ ☐ ☐ ☐ ☐ ☐	Terms Typically Offered ☑ Fall ☑ Winter ☐ Spring ☐ Summer ☐ Spring/Summer

Cognizant Faculty Member Name: Nicole Hamilton	Cognizant Faculty Member Title:

SIGNATURES ARE REQUIRED FROM ALL DEPARTMENTS INVOLVED (Please Print AND Sign Name)

Contact Person: Punam Vyas Email: vyas@umich.edu Phone: 647-1754

Curriculum Committee Member:		Print: Edmund H. Durfee	Date: 1/6/20
Curriculum Committee Chair:		Print:	Date:
Home Department Chair:		Print:	Date:
Cross-Listed Department Chair:		Print:	Date:
Cross-Listed Department Chair:		Print:	Date:
Cross-Listed Department Chair:		Print:	Date:

DEPARTMENTAL/COLLEGE USE ONLY

Current:	Requested:
Course Description	Course Description Design and development of a search engine. Topics include search engine internals, OS facilities, file systems, sockets, and threads. Students work in teams to write a complete multithreaded internet search engine from scratch.
Class Length	Class Length Full term
Contact hours (lecture):	Contact hours (lecture): 4
Contact hours (recitation)	Contact hours (recitation)
Contact hours (lab)	Contact hours (lab) 1

Additional Info:

Submitted by:
Home dept

Describe how this course fits with the degree requirements:
The course satisfies the major design experience in the CS-Eng major and the capstone requirement for students in the CS-LSA major.

ABET departmental program outcomes for undergraduate courses:
1,2,3,5,6

Special resources of facilities required for this course:

Supporting statement:
The proposed course addresses the currently unfilled need for a course that replicates the experience of working on a startup team of software developers (“devs”) on a complex system design project. Over the semester, each team of six students must plan and then build a working internet search engine, completely from scratch in C++. The project is both “close to the metal” and one that integrates a multiplicity of topics in computer science, including operating systems, networking, compilers, data structures, algorithms and information retrieval. It exposes students to the practical applications of multithreading and locks, sockets, mapped files and data structures on disk, and top-down recursive descent compiling.

See accompanying documentation for a more complete supporting statement, including a summary of the syllabus, of enrollments, and of student evaluations.

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.

EECS 398 System Design of a Search Engine
Fall 2019 Syllabus

Instructor Nicole Hamilton
nham@umich.edu
https://web.eecs.umich.edu/~nham/
C: 425-765-9574

Lectures EECS 1200
Tuesdays and Thursdays, 3:00 pm to 4:50 pm
Lectures will be recorded but attendance is strongly advised.

Office hours Beyster 2649
Mondays and Wednesdays, 1:30 pm to 2:30 pm
Tuesdays and Thursdays, 5:00 pm to 6:00 pm

Prerequisites EECS 280 and 281

Classification Depending on your major:

CE:	MDE or EECS elective
CSE:	MDE or Flex Tech elective
CS-LSA:	Capstone or Flex Tech elective
EE and DS:	Flex Tech elective

Overview

This is a course in how to tackle a large system programming project in C++. You'll work on a team of 6 to write a complete multithreaded internet search engine from scratch.

My objective is to offer students an opportunity to work on a significant *relatable* design project (something you can explain to your family and to recruiters) early in your undergraduate careers. I want the experience to be like working on the startup dev team for a new product (like the one I had on the startup team for what became the Bing engine.)

Sixty percent of your grade will be on the project and will be competitive, based mostly on your team's performance compared to the others and on your individual contribution to your team. Thirty percent of your grade will be on the exams. Ten percent is homework.

This will be the third time for this course. The engines submitted in the past have ranged in size, averaging around 6 KLOC. (Yes, a little smaller than you might have guessed.) Sounds easy (maybe) but you will deal with a lot more ambiguity than in most classes, in part because the course is still so raw and under construction.

Learning objectives

Here are main learning objectives:

1. The experience of working on the startup dev team for a significant project in C++, building a complete working internet search engine from scratch.
2. The knowledge, skills and ability to work with others to research, define, estimate, plan and carry out the design of such a large, complex software project, especially, one you don't already know how to do.
3. An understanding of how a large system is decomposed into separable parts with interfaces exposing a limited surface area and how they talk to each other.
4. An understanding of and ability to use the low-level operating system facilities for files, process and thread creation, synchronization, IPC and sockets.
5. An appreciation of software as art and of the software design process as a fundamentally creative activity where you make choices.
6. The ability to communicate your results by demonstrating your product, producing a final report and presenting your work to the class and invited guests in a professional manner.

System design

System design projects always seem to have some defining characteristics.

1. There's an important domain-specific part that asks you to learn something new about an interesting problem you've never seen before, in this case, how a search engine works.
2. There's a need to invent a solution, an architecture, breaking the problem down into lots of moving parts, with lots of data structures and algorithms.
3. It's usually "close to the metal" with lots of low-level OS calls everywhere and the need to define file and interchange formats, perform handshakes, share resources using locks, deal seriously with error recovery, etc.
4. They're usually team efforts because they're too big to do any other way.
5. You build the whole thing, often from scratch, and at the end, you get to see it work and it feels good.

Every new design project always has a new domain-specific part, which keeps a career in system design interesting, but the rest of the skills are the same and they're things most people learn by doing.

What I like about a search engine as a case study in system design is that it's a really compact fun project that hits on every bit of what, to me, system design is all about.

The project

You will self-select into teams of 6 to design and build a complete working internet search engine, assigning your own roles. Think about who you'd like to work with and see if they'd like to go in with you. Teams that form early have advantage. If you've taken 280 and 281, I will try to cover the rest of what you need to know to build any part of the engine.

These are the basic pieces you will need to build.

1. HTML parser.
2. Crawler.
3. Index.
4. Constraint solver.
5. Query language.
6. Ranker.
7. Front end.

In lectures, we'll walk through the problem posed by each piece and how it might be solved. You will also find the text helpful in explaining how a search engine works. A lot of the pieces need to talk to the operating system, so we'll go through a lot of toy examples in C++ on Linux in lecture, lab and homework showing, e.g., how to memory-map a file, create a thread or open a socket.

But you will have to make many of your own decisions as a group about what your architecture should look like, what's minimally needed and what's nice to have, how each piece will work, how the pieces will talk and how you'll get it all done on time.

All your work will be in C++ 11.0 and all of it must be yours. All your code must adhere to the style sheet I'll provide. I'm still struggling with how to set guidance but I prefer you not use a lot of STL outside of test cases. I'd like you do your own problem-solving, inventing your own data structures and templates. I think it will raise your awareness of lower-level design and performance issues.

Your grade on the project will depend in roughly equal measure on your individual and team performance. Your team performance will be determined on a competitive basis by ranking your engine's overall performance, features, index size and quality against the engines created by the other teams. I will also consider your team's ability to communicate your results through your presentation, demonstration and final report.

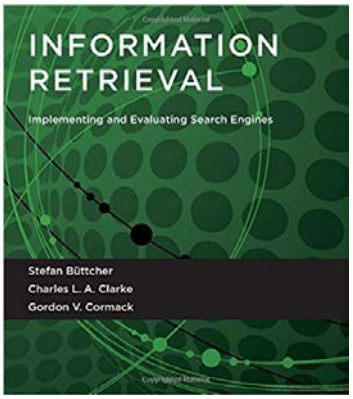
Your individual performance on the project will be based on your ability to work with others and contribute to your team's success. I'll primarily consider the number of lines of code you write, the complexity of the tasks it performs, its performance, the creativity it displays, and the overall elegance. I will also consider how you've contributed to the atmosphere, enthusiasm, creativity, progress, inclusiveness, camaraderie, health and overall success of your team.

Homework

I anticipate perhaps a half-dozen homework or lab assignments at two or three-week intervals.

Texts

There will be only one required text. What I like about *Information Retrieval* is that it describes a lot of the pieces (except for ranking) in the same way we thought about it at Microsoft.



Information Retrieval: Implementing and Evaluating Search Engines
Reprint edition (February 12, 2016)
Stefan Büttcher, Charles L.A. Clarke, Gordon V. Cormack
The MIT Press
ISBN 978-0262528870

Grading

Here's a rough idea of the weighting I expect to apply.

Group project and other activities	
Group performance	30%
Individual contribution	30%
Homework	10%
Midterm	15%
Optional final exam	15%

If you choose to skip the final, I will use your midterm score. I grade on the curve, scaling the results by deciding what I think was an A and what I think was a C and then interpolating in between. I expect but do not promise that most students will fall between 2.8 and 4.0 with median around 3.2 or a little higher.

Late policy

Late submissions will not be accepted.

Late withdrawals

I intend to deny virtually all requests for withdrawals once the teams are formed and underway.

Course revisions

This is only the third time for this course so of course you should expect I'm still learning how to make it better and that I may make reasonable revisions to content, assignments, exams or grading as may be appropriate.

All the work must be your own

You may certainly compare notes with other students and other teams and of course I understand that you may do research using Google. But absolutely everything you turn in to me must be your own work. On your exams, you will be required to copy the Honor Pledge in your own handwriting and sign your name to it.

> *I have neither given nor received unauthorized aid on this examination, nor have I concealed any violations of the Honor Code.*

I report everything.

Disability

Access and Accommodations: Your experience in this class is important to me and to the University of Michigan, where it is our policy and practice to create inclusive and accessible learning environments consistent with federal and state law.

If you experience barriers based on a temporary or permanent disability (including, but not limited to mental health, attention-related, learning, vision, hearing, physical or health impacts), please seek a meeting with the Services for Students with Disabilities (SSD) office at 734-763-3000 or https://ssd.umich.edu to help us determine appropriate academic accommodations. SSD typically recommends accommodations through a Verified Individualized Services and Accommodations (VISA) form. Any information you provide is private and confidential and will be treated as such.

If you have already established accommodations with SSD, please tell me what they are so I can be sure to provide them.

List of topics

Here is a rough outline of the topics we'll discuss, interwoven over the semester. We obviously won't dive into everything to the same depth nor necessarily in this order.

1. Introduction to the course. Expectations, grading, coding style requirements. Brief overview of the project and of how a search engine works.
2. Attributes and organization of a successful software development team. How to and how not.
3. The role of ideas and creativity in software design. Software as art. Beautiful code, efficiency, elegance. The style guide we'll follow.
4. The software design process.
    a. The initial vision. Something has become possible.
    b. Research, reverse engineering, brain-storming.
    c. Problem-choosing, deciding what's in and what's not.
    d. Decomposition into an architecture of components with interfaces.
    e. The product spec.
    f. Gantt charts and project planning.
    g. KLOCs, index size and other metrics.
5. Search engine basics.
    a. Brief history of information retrieval and web search.
    b. The concept of relevance.
    c. Dynamic and static rank.
    d. Text formats and code sets, Unicode and UTF-8.
    e. Simple HTML, title, heading, body text, links and anchor text.
    f. Tokenization, stop words, stemming.
    g. Crawling and the frontier.
    h. Politeness in crawling.
    i. Duplicates and shingling.
    j. The inverted word index.
    k. The constraint solver.
    l. Query compiling.
    m. Simple dynamic ranking.
6. OS facilities.
    a. The file system.
    b. Memory-mapped files.
    c. Sockets and SSL.
    d. Processes and threads.
    e. Servers.
    f. Producer-consumer relationships.
    g. Locks, critical sections, race conditions.

7. Ethics. What does it mean to be ethical? What does it mean to be an ethical engineer? Are we responsible for our products? What if someone could use our products to harm people? What if we know there's a probability they may fail and kill people?

University of Michigan
Office of the Registrar - Evaluations
ro.umich.edu/evals/

Winter 2018 Final
20 students responded out of the total enrolled 26

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*					
									University Wide			School/College		
		5 SA	4 A	3 N	2 D	1 SD	NA	Your 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

Winter 2019 Instructor Report With Comments of EECS 398-001: Special Topics for Nicole Hamilton

Project Title: **Winter 2019 Teaching Evaluation**

Course Audience: **61**
Responses Received: **15**
Response Ratio: **24.6%**

Report Comments

This report is a summary that tabulates all quantitative ratings on a single page. Results from the open-ended questions appear at the end of this report. Ratings are from the Winter 2019 teaching evaluations of EECS 398-001: Special Topics.

Prepared by: **Office of the Registrar**
Creation Date: **Monday, May 6, 2019**

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)	10	5	0	0	0	0	4.8	4.5	4.4
My interest in the subject has increased because of this course. (Q1632)	10	2	1	2	0	0	4.8	4.2	4.0
I knew what was expected of me in this course.(Q1633)	4	4	2	3	2	0	3.6	4.5	4.2
Overall, this was an excellent course.(Q1)	5	7	1	2	0	0	4.1	4.2	4.0
I had a strong desire to take this course.(Q4)	7	7	1	0	0	0	4.4	4.0	3.9
As compared with other courses of equal credit, the workload for this course was... (SA=Much Lighter to SD=Much Heavier)	0	0	6	7	2	0	2.3	3.0	2.9

Responses to the 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.	6	7	1	1	0	0	4.3	4.5	4.4
Nicole Hamilton seemed well prepared for class meetings.	3	8	1	2	1	0	3.9	4.8	4.6
Nicole Hamilton explained material clearly.	4	8	3	0	0	0	4.1	4.6	4.5
Nicole Hamilton treated students with respect.	10	2	2	0	0	0	4.8	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	3	2	4	1	0	3.7	4.3
The textbook made a valuable contribution to the course. (Q340)	2	1	4	4	1	3	2.8	3.6
I developed confidence in my abilities as an engineer.	5	8	0	2	0	0	4.2	4.0
I developed the ability to solve real world engineering problems.	9	4	1	1	0	0	4.7	4.0

EECS 440 System Design of a Search Engine
Supporting Statement

The proposed course will address the currently unfilled need for a course which replicates the experience of working on the startup team of software developers ("devs") on a complex system design project. Over the semester, each team of six students must plan and then build a working internet search engine, completely from scratch in C++.

The project is both "close to the metal" and one that integrates a multiplicity of topics in computer science, including operating systems, networking, compilers, data structures, algorithms and information retrieval. It exposes students to the practical applications of multithreading and locks, sockets, mapped files and data structures on disk, and top-down recursive descent compiling, making this an ideal student MDE.

As on a real team, students are forced to negotiate and cooperate to make decisions on product goals, architecture, and assignments and responsibilities, often under uncertainty.

Roughly 30% the student's final grade is based on their ability to work with others and contribute to their team's success, e.g., as defined in ABET Outcome 5, and is measured both objectively, e.g., by lines of code and other contributions compared to others on the team and in the class, and subjectively, e.g., by survey of the student's teammates.

Past offerings

The course has been taught three times as an experimental EECS 398 course, in W18 as System Design in C++ and in W19 and F19 as System Design of a Search Engine, better reflecting what the course is really about.

	W18	*W19*	*F19*	*Total*
Enrollment	26	61	58	145
Student teams	5	10	10	25

Student evals have shown both interest and improvement as I've added content, better organized the material and responded to suggestions, e.g., to drop the discussion of the Windows API to focus on Linux. Students are clearly satisfied with the outcomes, especially, their understanding of and interest in the subject matter, and their confidence and ability to solve real world engineering problems.

Student evals	W18 Final	W19 Midterm	W19 Final	F19 Midterm	F19 Final
Q1. Overall, this is an excellent course	2.70	3.6	4.2	4.6	4.0
Q2. Overall, Nicole Hamilton is an excellent teacher.	3.75	3.4	4.3	4.6	4.1
Q3. I learned / am learning a great deal from this course.	3.38	3.9	--	4.7	--
Q4. I had a strong desire to take this course.	4.67	4.7	4.4	4.8	4.6
Q61. Prerequisites provided adequate preparation for the course.	3.17	--	3.7	--	4.0
Q1631. This course advanced my understanding of the subject matter.	--	4.4	4.8	4.6	4.6
Q1632. My interest in the subject has increased because of this course.	--	4.8	4.8	4.6	4.6
Q1769. I developed confidence in my abilities as an engineer.	--	--	4.2	--	4.6
Q1770. I developed the ability to solve real world engineering problems.	--	--	4.7	--	4.7

The falloff in Q1 and Q2 scores between the midterm and final evals in F19 is probably deserved. I need to cover most of the essential topics beyond 281 needed to build a whole search engine by about midway through the course if the students are to have enough time left to actually do it. I'm doing well on that. But then I need to fill my remaining lectures with content that's relevant and interesting but not essential to completing the project. That still needs work.

Course development

In W18, I started with little more than a syllabus and struggled to create two hours of original lecture material every two days. And the evals reflect that. The big risk was that the task was impossible for a small team of students in one semester. But while some teams did better than others, every team succeeded in delivering a working engine.

In W19, I renamed the course to communicate more clearly what it was about and focused on organizing the material to get teams crawling the web sooner, added lecture content and, with the help of an IA, created some autograded homework and labs.

In F19, again with the help of my IAs, I've continued to add labs, homework and lecture content and to improve the sequence of topics. Responding to student feedback, I dropped the Windows discussion. I also arranged for each student to receive a free $100 student Amazon Web Service (AWS) account so they can build their engines on the cloud.

I also introduced a simple Google survey in which students were asked to rate their teammates' contributions and effectiveness as members or leaders of a team, in support the ABET objective. Actual example individual and team reports, but with fictious names, are attached.

Student engines

As the course has improved, the students' indexes have steadily grown. Their engines are typically crawling on multiple machines on the cloud with gigabit internet connections and anywhere from 1,000 to over 14,000 threads. (In W19, as I reorganized the material to get teams crawling faster, they got better faster than I expected; one team accidentally DOS'ed the Duke University Law School Registrar's site during reg week. We now spend much more time discussing politeness before unleashing such awesome power.)

Index sizes in documents	*W18*	*W19*	*F19*
High	13.4M	150M	586M
Median	8,000	4.65M	32.12M
Low	4,816	1.9M	1.2M

Engines in lines of code	*W18*	*W19*	*F19*
High	14,271	26,887	17,079
Median	5,300	13,000	11,973
Low	4,170	9,414	4,366

Individual lines of code	*W18*	*W19*	*F19*
High	4,096	8,646	7,826
Median	1,006	1,750	1,571
Low	0	444	120

Reflections on F19

I've now run 25 teams through this class. I know the class is getting better, but a surprising result in F19 was that for the first time, I had 3 teams decide to take incompletes and finish in January because they weren't ready to demo.

In my final meetings with each team, students frequently cited fall semester distractions like recruiting and football, but it was also apparent that many seemed to have falsely regarded the Thanksgiving break as the halfway point in the semester, which it's not, especially this year, when it came late. (Thanksgiving recess was 85 days into F19; winter vacation is only 52 days into W20.) It's possible the course is better scheduled for the winter.

But it's also possible that I contributed to the problem by making the course "better". As I added content and structure, especially, labs and homework, I intended for these to be helpful with important

concepts and students consistently remarked that they were. But the result may have been to distract energy and attention from the project.

In my final reviews with the individual teams, then later in their evals, many students agreed with a change I propose to make next time, which is to announce the homework and lab assignments all at once, early in the semester, and make them all group assignments. I can do that now that they all exist.

In previous semesters, students suggested I prereq EECS 482 Operating Systems. But I wanted the course to be accessible to sophomores, so I resolved to improve my treatment of these topics. I was gratified that the course evals this time indicated agreement that the course is fully accessible to students with only EECS 281.

The peer evaluation surveys worked well. I got a high response rate and the results were helpful, especially in the insights it gave me into the team dynamics. For example, on one team, one student contributed a huge share of the total lines of code in their project which looked pretty good on the numbers alone. But his teammates rated him poorly on quality, reliability and cooperativeness; this led to discovery in discussion with the group that he had habit of dumping big lumps of buggy code on them and then disappearing from contact.

I also had one team that tried to game the system by giving each other 5 out 5 on every attribute. This was one of the teams that took an incomplete, so they were definitely not all 5 out 5 on everything. Going forward, I intend to instruct students that they are on their honor not to collude or discuss their peer evals with each other and that if I detect gaming, I may ignore their surveys in determining their grades.

Several students remarked in the evals and in discussions that if they didn't come in with friends in the course, that it was difficult to find a group. Not wanting to be rude by walking away to look for a better match, they teamed with the first people they talked to. This sometimes led to dysfunctional teams, the most common problem being that they couldn't get everyone in one room at the same time to get stuff done. At the same time, students were overwhelmingly in favor of being able to choose their own teams; they did not want to be assigned.

I hope to somewhat ameliorate this problem of finding a compatible group next time by creating a shared Google spreadsheet where students may fill in contact info, best times and places (e.g., North vs. Central) for meetings, and any other relevant info they care. I may also introduce some "speed dating" exercises into the first couple of lectures.

Looking ahead

My hopes for this class are:

1. To continue developing it toward what I hope will be a complete and largely turnkey course with content and staff that other instructors might also enjoy teaching or working on. I think I'm getting close.
2. For the students, to deliver an experience of success and satisfaction working on a small team building a complex system that they can explain to family and friends and to recruiters and hiring managers.

Example individual report.

Hello, Kara Danvers (kara).

Below is an individual report of your EECS 398 peer evaluations.
Results range from 1 (Strongly Disagree) to 5 (Strongly Agree).

	Your Rating	Team Average	Class Average
Sets High Standards And Delivers High Quality.	2.6	4.2	4.2
Treats Others With Respect And Respects Their Boundaries.	4.6	4.4	4.6
Listens To And Encourages Others' Ideas.	4.2	4.3	4.5
Is Cooperative.	4.2	4.4	4.5
Is Reliable, Shows Up, Answers Emails, Delivers On Time.	1.6	4.1	4.2
Contributed Lots Of Helpful Ideas.	3.4	4.2	4.2
Responds Constructively To Suggestions Or Criticism.	4.4	4.3	4.4
Really Stepped Up To Do Their Share Of The Work.	2.8	4.3	4.2
Is An Excellent Teammate.	2.8	4.3	4.3

Example team report.

SUMMARY FOR SUPERHEROS

This team member sets high standards and delivers high quality.

Name	Uniqname	Rating	Team Average	Class Average
Clark Kent	clark	4.6	3.8	4.2
Kara Danvers	kara	2.6	3.8	4.2
Bruce Wayne	bruce	4.4	3.8	4.2
Hal Jordan	hal	4.6	3.8	4.2
Barry Allen	barry	4.8	3.8	4.2
Arthur Curry	arthur	4.2	3.8	4.2

This team member treats others with respect and respects their boundaries.

Name	Uniqname	Rating	Team Average	Class Average
Clark Kent	clark	4.8	4.6	4.6
Kara Danvers	kara	4.6	4.6	4.6
Bruce Wayne	bruce	3.2	4.6	4.6
Hal Jordan	hal	4.6	4.6	4.6
Barry Allen	barry	4.6	4.6	4.6
Arthur Curry	arthur	4.6	4.6	4.6

This team member listens to and encourages othersΓÇÖ ideas.

Name	Uniqname	Rating	Team Average	Class Average
Clark Kent	clark	4.8	4	4.5
Kara Danvers	kara	4.2	4	4.5
Bruce Wayne	bruce	3	4	4.5
Hal Jordan	hal	4.4	4	4.5
Barry Allen	barry	4.6	4	4.5
Arthur Curry	arthur	4.6	4	4.5

This team member is cooperative.

Name	Uniqname	Rating	Team Average	Class Average
Clark Kent	clark	4.6	4.1	4.5
Kara Danvers	kara	4.2	4.1	4.5
Bruce Wayne	bruce	3.6	4.1	4.5
Hal Jordan	hal	4.4	4.1	4.5
Barry Allen	barry	4.8	4.1	4.5
Arthur Curry	arthur	4.6	4.1	4.5

This team member is reliable, shows up, answers emails, delivers on time.

Name	Uniqname	Rating	Team Average	Class Average
Clark Kent	clark	4.6	4.1	4.2
Kara Danvers	kara	1.6	4.1	4.2
Bruce Wayne	bruce	4.8	4.1	4.2
Hal Jordan	hal	4.2	4.1	4.2
Barry Allen	barry	4.6	4.1	4.2
Arthur Curry	arthur	4.8	4.1	4.2

This team member contributed lots of helpful ideas.

Name	Uniqname	Rating	Team Average	Class Average
Clark Kent	clark	4.2	3.8	4.2
Kara Danvers	kara	3.4	3.8	4.2
Bruce Wayne	bruce	4.4	3.8	4.2
Hal Jordan	hal	4.4	3.8	4.2
Barry Allen	barry	4.6	3.8	4.2
Arthur Curry	arthur	4.2	3.8	4.2

This team member responds constructively to suggestions or criticism.

Name	Uniqname	Rating	Team Average	Class Average
Clark Kent	clark	4.4	4.2	4.4
Kara Danvers	kara	4.4	4.2	4.4
Bruce Wayne	bruce	4	4.2	4.4
Hal Jordan	hal	4.2	4.2	4.4
Barry Allen	barry	4.6	4.2	4.4
Arthur Curry	arthur	4.4	4.2	4.4

This team member really stepped up to do their share of the work.

Name	Uniqname	Rating	Team Average	Class Average
Abraham A Alawy	aalawy	4.6	3.7	4.2
Kara Danvers	kara	2.8	3.7	4.2
Bruce Wayne	bruce	4.6	3.7	4.2
Hal Jordan	hal	4.6	3.7	4.2
David Leroy Smith	dleroys	4.8	3.7	4.2
Arthur Curry	arthur	4.6	3.7	4.2

Overall, this team member is an excellent teammate.

Name	Uniqname	Rating	Team Average	Class Average
Abraham A Alawy	aalawy	5	3.9	4.3
Kara Danvers	kara	2.8	3.9	4.3
Bruce Wayne	bruce	4	3.9	4.3
Hal Jordan	hal	4.6	3.9	4.3
David Leroy Smith	dleroys	4.8	3.9	4.3
Arthur Curry	arthur	4.4	3.9	4.3

Course Approval Request Form

Office of the Registrar, University of Michigan

1210 LSA Building
500 S. State Street
Ann Arbor, MI 48109-1382
Phone: 734.763.2113
Fax: 734.936.3148
ro.curriculum@umich.edu
ro.umich.edu

☑ **CHECK APPROPRIATE BOXES FOR ALL CHANGES**

Action Requested
- ☐ New Course
- ☑ Modification of Existing Course
- ☐ Deletion of Existing Course

Date of Submission: 2019-11-25
Effective Term: Fall 2020

☑	Course Offered ☑ Indefinitely ☐ One term only	**RO USE ONLY** Date Received: Date Completed: Completed By:

	CURRENT LISTING	**REQUESTED LISTING**
☐	Dept (Home): Elec Engin & Computer Sci Subject: EECS Catalog: 494	Dept (Home): Elec Engin & Computer Sci Subject: EECS Catalog: 494
☐	☐ Course is Cross-Listed with Other Departments Department / Subject / Catalog Number	☐ Course is Cross-Listed with Other Departments Department / Subject / Catalog Number
☐	Course Title (full title) Computer Game Design and Development	Course Title (full title) Computer Game Design and Development
☐	Abbreviated Title (20 char) Comp Game	Abbreviated Title (20 char) Comp Game
☑	Course Description (Please limit to 50 words and attach separate sheet if necessary) Design, development, and application of digital games. Topics include: game engines, design patterns, shaders and graphics programming, agile development methods, iterative game / experience design, project management and resource allocation, virtual reality, classic games, multidisciplinary relationships, product exhibition, and portfolio construction. Students work in teams to develop and exhibit new games.	
☑	Full Term Credit Hours Undergraduate Min: 4 Graduate Min: 4 Undergraduate Max: 4 Graduate Max: 4	Half Term Credit Hours Undergraduate Min: Graduate Min: Undergraduate Max: Graduate Max:
☑	Course Credit Type Undergraduate Student, Rackham Graduate Student	
☐	Repeatability ☐ Course is Repeatable for Credit Maximum number of repeatable credits:	☐ Course is Y graded ☐ Can be taken more than once in the same term

Subject: Elec Engin & Computer Sci Catalog: 494

☑

Grading Basis
- ☑ Graded (A – E)
- ☐ Credit/No Credit
- ☐ Satisfactory/Unsatisfactory
- ☐ Pass/Fail
- ☐ Business Administration

Grading
- ☐ Not for Credit
- ☐ Not for Degree Credit
- ☐ Degree Credit Only

Add Consent
- ☐ Department Consent
- ☐ Instructor Consent
- ☑ No Consent

Drop Consent
- ☐ Department Consent
- ☐ Instructor Consent
- ☑ No Consent

	CURRENT LISTING	**REQUESTED LISTING**
☐	Advisory Prerequisite (254 char)	Advisory Prerequisite (254 char)
☑	Enforced Prerequisite (254 char) (EECS 281 and (C or better)) or graduate standing in CSE Minimum grade requirement:	Enforced Prerequisite (254 char) EECS 281(minimum grade C) or Grad Standing in CSE Minimum grade requirement: C
☐	Credit Exclusions	Credit Exclusions

☑

Course Components
- ☑ Lecture
- ☐ Seminar
- ☐ Recitation
- ☐ Lab
- ☐ Discussion
- ☐ Independent Study

Graded Component
- ☑
- ☐
- ☐
- ☐
- ☐
- ☐

Terms Typically Offered
- ☑ Fall
- ☑ Winter
- ☐ Spring
- ☐ Summer
- ☐ Spring/Summer

Cognizant Faculty Member Name: Austin Yarger Cognizant Faculty Member Title:

SIGNATURES ARE REQUIRED FROM ALL DEPARTMENTS INVOLVED (Please Print AND Sign Name)

Contact Person: Email: Phone:

Curriculum Committee Member:  Print: Edmund H. Durfee Date: 1/6/20

Curriculum Committee Chair: Print: Date:

Home Department Chair: Print: Date:

Cross-Listed Department Chair: Print: Date:

Cross-Listed Department Chair: Print: Date:

Cross-Listed Department Chair: Print: Date:

DEPARTMENTAL/COLLEGE USE ONLY

Current:	Requested:
Course Description Concepts and methods for the design and development of computer games. Topics include: history of games, 2D graphics and animation, sprites, 3D animation, binary space partition trees, software engineering, game design, interactive fiction, user interfaces, artificial intelligence, game SDK's, networking, multi-player games, game development environments, commercialization of software.	Course Description Design, development, and application of digital games. Topics include: game engines, design patterns, shaders and graphics programming, agile development methods, iterative game / experience design, project management and resource allocation, virtual reality, classic games, multidisciplinary relationships, product exhibition, and portfolio construction. Students work in teams to develop and exhibit new games.
Class Length Full term	Class Length Full term
Contact hours (lecture):	Contact hours (lecture): 4
Contact hours (recitation)	Contact hours (recitation)
Contact hours (lab)	Contact hours (lab)

Additional Info:

Submitted by:
Home dept

Describe how this course fits with the degree requirements:

ABET departmental program outcomes for undergraduate courses:
1,2,3,4,5

Special resources of facilities required for this course:

Supporting statement:
The only modification to this course is its description. In the time since the existing description came into use, EECS 494 has seen three new primary instructors take the reins, in addition to substantial industry / technology advancement, causing a gradual shift in course content. While the current iteration of EECS 494 enjoys quality course output and high student ratings, its current description misses a number of topics that are now addressed (shaders, project management, etc), and includes some that are no longer a course focus (binary space partition trees, interactive fiction, etc). The modified description thus better represents the evolved emphases of the course.