Department of Computer Science

Chair

- Henry Hoffmann

Professors

- Laszlo Babai
- David Cash
- Andrew Chien
- Frederic Chong
- Nick Feamster
- Ian Foster
- Michael Franklin
- Henry Hoffmann
- Stuart A. Kurtz
- Shan Lu
- Ketan Mulmuley
- Alexander Razborov
- John Reppy
- Janos Simon
- Rick L. Stevens
- Rebecca Willett
- Ben Zhao
- Heather Zheng

Professors Emeriti

- John Goldsmith

Associate Professors

- Marshini Chetty
- Ravi Chugh
- Aaron Elmore
- Diana Franklin
- Haryadi Gunawi
- Gordon Kindlmann
- Risi Kondor
- Pedro Lopes
- Michael Maire
- Anne Rogers
- Blase Ur

Assistant Professors

- Raul Castro Fernandez
- Yuxin Chen
- Aloni Cohen
- William Fefferman
- Rana Hanocka
- Grant Ho
- Junchen Jiang
- Alex Kale
- Sanjay Krishnan
- Yanjing Li
- Ken Nakagaki
- Lorenzo Orecchia

- Aaron Potechin
- Robert Rand
- Sarah Sebo
- Chenhao Tan
- Haifeng Xu

Research Assistant Professors

- Kyle Chard

Clinical faculty

- Ishan Agarwal (Assistant Clinical Professor)
- Rafi Almhana (Assistant Clinical Professor)
- Andrew Binkowski (Associate Clinical Professor)
- Gerry Brady (Clinical Professor)
- Amitabh Chaudhary (Clinical Professor)
- Lamont Samuels (Assistant Clinical Professor)
- James Turk (Assistant Clinical Professor)

Adjunct faculty

- Jeffrey Cohen (Adjunct Associate Professor)
- Adam Gerber (Adjunct Associate Professor)
- Jan Hueckelheim (Adjunct Assistant Professor)
- Anthony Nicholson (Adjunct Associate Professor)
- Drew Olson (Adjunct Assistant Professor)
- Shelley Rossell (Adjunct Associate Professor)
- Mark Shacklette (Adjunct Professor)
- Andrew Siegel (Adjunct Professor)
- Michael Spertus (Adjunct Professor)
- Vasilios Vasiliadis (Adjunct Associate Professor)
- Kathryn Wozniak (Adjunct Associate Professor)

The Department of Computer Science is dedicated to advancing and improving the knowledge, understanding, and practice of computer science through basic research and education.

Research

We have a broad view of Computer Science, that includes the whole spectrum of computing, from relevant mathematics and statistics to building hardware devices, networks, data science, machine learning, human computer interaction, and Computer Science education. The list is not exhaustive, but an attempt to convey some of the research interests of current faculty. We investigate computation, information, communication and data as fundamental phenomena to be studied in Computer Science. We also investigate the many interdisciplinary ways the study of computation interacts with other disciplines like the sciences, society, and learning.

There is an ongoing major thrust to expand the role of Computer Science and computation at the University, with considerable expansion of the faculty, and expanded support to explore new research areas. These include the new Data Science Institute (DSI), a joint effort of the Statistics and Computer Science Departments, that is also experiencing rapid growth. Accordingly, the descriptions below, a snapshot of our current active research, are likely to expand.

Current active research areas include computing systems, computer architecture, computer security and privacy, error-tolerant computing and error recovery in computing systems, databases and data intensive computing, theoretical computer science, discrete mathematics, quantum computing, programming languages, machine learning, computational linguistics, computer vision, cloud computing, sustainable computing, scientific computing, data visualization, high performance computing, human-computer interaction, computer science education, robotics, and interdisciplinary research in computing in the physical, biological, and social sciences.

Current major research areas include:

Theoretical Computer Science

Programming Languages

Data Science

Machine Learning and AI

HCI

Scientific and High Performance Computing

Computer Science Education

For more information about current research groups and active research areas, refer to the Research section of the departmental webpage (https://www.cs.uchicago.edu/research/research-areas/).

Our research efforts are enhanced by the interdisciplinary nature of the University, and, specifically, by strong connections to the Data Science Institute (DSI), the Center for Translational Data Science, the James Frank Institute, the Institute for Biophysical Dynamics, with the Pritzker School of Molecular Engineering, and with the Booth School of Business. Our research collaborations involve faculty from many different departments, including Mathematics, Statistics, Physics, Linguistics, Psychology, and Sociology. We have very strong research ties with ANL, the Argonne National Laboratory, operated by the University for the US DOE. We also have almost seamless collaborations with the Toyota Technological Institute Chicago, on campus: many TTIC faculty have part-time appointments in the Department.

Graduate Programs

We offer two graduate curricula in computer science.

1. A graduate professional curriculum leading to the Master of Science (MS) degree, for students who wish to enter or advance themselves in computer science practice.
2. A graduate research curriculum leading to the PhD degree that prepares students to perform advanced basic research in computer science either in industry or academia. Teaching experience is available for students preparing for academic careers.

Acquire further information about our Masters Program in Computer Science (MPCS) through the MPCS website (http://masters.cs.uchicago.edu/), by writing to our MPCS Admissions, Department of Computer Science, University of Chicago, 5730 S Ellis Avenue, IL 60637, or by telephoning 773.834.3388. You may also email any questions to our questions@cs.uchicago.edu email address.

Acquire further information about our PhD program through our PhD admissions website, (https://www.cs.uchicago.edu/graduate/phd-programs/) by writing to Admissions, Department of Computer Science, University of Chicago, 5730 S. Ellis Avenue, Chicago, IL 60637, or by telephoning 773.702.6011.

General information about our department is available from the departmental website (https://computerscience.uchicago.edu/).

The PhD Program

The PhD program is done in three phases:

Qualifying Phase

The objective of the Qualifying Phase is to provide the scientific foundations in their area, a breadth of knowledge within Computer Science, and the ability to write up research results. Students without a previous MS should satisfy course requirements, and write and defend an MS paper.

The course requirement consists of five core courses (two in Theory, two in Systems, and one in ML), three electives, and a seminar. In order to ensure breadth, no more than four of these courses should be in the student's main research area.

After the public defense of the paper, the student may be eligible to receive an MS in Computer Science from the University. Please refer to the departmental webpage (https://www.cs.uchicago.edu/graduate/phd-programs/phase-1-masters-degree-within-the-phd-program/) for details.

Students with a previous MS may receive credit for their previous work: up to waiving the core course requirement. The details of the mechanism to do so can be found on our website (https://www.cs.uchicago.edu/graduate/phd-programs/).

The minimum formal requirements for the core courses (”Ph.D. Pass”) are the following: Students are required to complete the five core courses with a grade point average (GPA) of at least 3.25 in the five core courses. In computing the GPA, A=4, B=3, and a + or a - counts as .3 of a point. Note that for the core courses, students who significantly outperform even the typical ”A” students may receive a grade of ”A+” (recorded internally by the CS Department Student Representative since the University does not officially grant the grade of A+.) So, for instance a student with grades A+, B+, B+, B-, B- in the five core courses has a GPA of 3.26 and thus satisfies the minimum GPA requirement, as does a student with grades of A+, A, B+, B, and C-. In the graduate program grades below C- are not passing grades.

Students must complete their electives with a grade of B or better in each course.

Candidacy Exam

After the Qualifying Phase, students must pass an exam for Admission to Candidacy, that is a plan for their dissertation research.

Dissertation and Dissertation Defense

Finally, they must write and defend their dissertation.

Teaching Opportunities for Students in the PhD Program

The department takes its undergraduate teaching responsibilities very seriously, and offers supervised teaching opportunities, including lecturing, acting as teaching assistants, and working as lab assistants to its best graduate students.

Computing Facilities

In addition to the general University computing facilities including the Research Computing Center (https://rcc.uchicago.edu/resources) and access to high performance computers at ANL, and our Computer Science Instructional Laboratory (which contains about 50 Macintosh computers and 40 desktops running Linux), our department provides the faculty, students, and postdoctoral associates in computer science with computing resources. We have the flexibility to adapt quickly to new research needs.

The resources include: 24 hour 7 day interactive computing on a number of shared computing servers as well as individually assigned desktops. These servers and desktops run the Linux operating system and are interconnected via high speed Ethernet. These systems are supported by substantial amounts of both local and networked disk storage for individual group use and are backed up regularly. Linux servers are available for general instructional and research purposes as well as hardware and virtual machines which are adapted to specialized needs.

Individual research groups operate additional computing facilities, often with considerable computing and storage resources. Some have specialized hardware, and fabrication tools (for example, GPU arrays, 3-D printers, etc.). Information about these resources is available from the research group's webpage.

Joint Math/CS PhD program

The Department of Computer Science and the Department of Mathematics offer a joint PhD program. For more details see https://mathematics.uchicago.edu/graduate/joint-math-cs-phd-program/

Courses

For the list of courses offered and the course descriptions, please consult the courses section of the departmental web page (https://computerscience.uchicago.edu/graduate/courses/).

Computer Science Courses

CMSC 30100. Technical Writing and Presentation. 100 Units.
Clear, logical writing and presentations are foundational skills for computer scientists. This class is meant to introduce computer science students to basic ideas and techniques for effective communication in both writing and presentations. The class will include several complementary components, including critical analysis of technical papers, weekly writing assignments focusing on writing style, clarity, and logical flow, and discussions of style for different research areas and venues. Later weeks will focus on skills for effective technical presentations in different settings, e.g. conference presentations, job talks, and keynotes. The course is primarily targeted towards graduates students, although undergraduates can audit the class (or enroll with permission from the instructor).
Instructor(s): Ben Zhao
Prerequisite(s): None

CMSC 30300. Introduction to Human-Computer Interaction. 100 Units.
An introduction to the field of Human-Computer Interaction (HCI), with an emphasis in understanding, designing and programming user-facing software and hardware systems. This class covers the core concepts of HCI: affordances, mental models, selection techniques (pointing, touch, menus, text entry, widgets, etc), conducting user studies (psychophysics, basic statistics, etc), rapid prototyping (3D printing, etc), and the fundamentals of 3D interfaces (optics for VR, AR, etc). We compliment the lectures with weekly programming assignments and two larger projects, in which we build/program/test user-facing interactive systems.
Prerequisite(s): CMSC 14200 or CMSC 15400 or CMSC 22000
Equivalent Course(s): MADD 25300, CMSC 20300

CMSC 30350. Security, Privacy, and Consumer Protection. 100 Units.
This course will cover the principles and practice of security, privacy, and consumer protection. Topics include: basic cryptography; physical, network, endpoint, and data security; privacy (including user surveillance and

tracking); attacks and defenses; and relevant concepts in usable security. The course will place fundamental security and privacy concepts in the context of past and ongoing legal, regulatory, and policy developments, including: consumer privacy, censorship, platform content moderation, data breaches, net neutrality, government surveillance, election security, vulnerability discovery and disclosure, and the fairness and accountability of automated decision making, including machine learning systems. Students will learn both technical fundamentals and how to apply these concepts to public policy outputs and recommendations.
Instructor(s): Feamster, Nicholas
Prerequisite(s): CMSC 14300 or CMSC 15400 or equivalent, and instructor consent.
Note(s): Prerequisites: CMSC 14300 or CMSC 15400 or equivalent, or graduate student. Instructor consent required.
Equivalent Course(s): CAPP 30350, CMSC 23206

CMSC 30370. Inclusive Technology: Designing for Underserved and Marginalized Populations. 100 Units.
Creating technologies that are inclusive of people in marginalized communities involves more than having technically sophisticated algorithms, systems, and infrastructure. It involves deeply understanding various community needs and using this understanding coupled with our knowledge of how people think and behave to design user-facing interfaces that can enhance and augment human capabilities. When dealing with underserved and marginalized communities, achieving these goals requires us to think through how different constraints such as costs, access to resources, and various cognitive and physical capabilities shape what socio-technical systems can best address a particular issue. This course leverages human-computer interaction and the tools, techniques, and principles that guide research on people to introduce you to the concepts of inclusive technology design. You will learn about different underserved and marginalized communities such as children, the elderly, those needing assistive technology, and users in developing countries, and their particular needs. In addition, you will learn how to be mindful of working with populations that can easily be exploited and how to think creatively of inclusive technology solutions. You will also put your skills into practice in a semester long group project involving the creation of an interactive system for one of the user populations we study.
Prerequisite(s): CMSC 14400 or CMSC 15400 or CMSC 12300 or CMSC 22000 or CMSC 20300
Equivalent Course(s): CMSC 20370, MADD 20370

CMSC 30380. Actuated User Interfaces and Technology. 100 Units.
The recent advancement in interactive technologies allows computer scientists, designers, and researchers to prototype and experiment with future user interfaces that can dynamically move and shape-change. This class offers hands-on experience in learning and employing actuated and shape-changing user interface technologies to build interactive user experiences. The class provides a range of basic engineering techniques to allow students to develop their own actuated user interface systems, including 3D mechanical design, digital fabrication (e.g. 3D Printing), electronics (Arduino microcontroller), and actuator control (utilizing different kinds of motors). Through multiple project-based assignments, students practice the acquired techniques to build interactive tangible experiences of their own.
Prerequisite(s): CMSC 20300
Equivalent Course(s): MADD 20380, CMSC 20380

CMSC 30600. Introduction to Robotics. 100 Units.
The University of Chicago's CMSC 20600 Introduction to Robotics course gives students a hands-on introduction to robot programming covering topics including sensing in real-world environments, sensory-motor control, state estimation, localization, forward/inverse kinematics, vision, and reinforcement learning. This course is centered around 3 mini projects exploring central concepts to robot programming and 1 final project whose topic is chosen by the students. Each of these mini projects will involve students programming real, physical robots interacting with the real world. The use of physical robots and real-world environments is essential in order for students to 1) see the result of their programs 'come to life' in a physical environment and 2) gain experience facing and overcoming the challenges of programming robots (e.g., sensor noise, edge cases due to environment variability, physical constraints of the robot and environment).
Prerequisite(s): CMSC 14200 or CMSC 15400
Equivalent Course(s): CMSC 20600

CMSC 30630. Human-Robot Interaction: Research and Practice. 100 Units.
The field of human-robot interaction (HRI) is a new and growing field of research that explores the interface between people and robots. Applications of HRI research include developing robots to tutor elementary school students, assist human workers in manufacturing contexts, provide museum tours, interact with families within their homes, and help care for the elderly. The field of HRI is highly interdisciplinary, incorporating methods and techniques from human-computer interaction, robotics, psychology, artificial intelligence, and other fields. This course exposes students to a broad range of recent and cutting-edge research in HRI. The topics covered in this course include: nonverbal robot behavior, verbal robot behavior, social dynamics, norms and ethics, collaboration and learning, group interactions, applications, and future challenges of HRI. Course meetings will involve students in the class leading discussions about cutting-edge peer-reviewed research HRI publications. Throughout the quarter, teams of students in the course will also complete an HRI course project of their choosing where they will investigate an HRI research question of interest to them.
Prerequisite(s): CMSC 14200 or CMSC 15400
Equivalent Course(s): CMSC 20630

CMSC 30900. Computers for Learning. 100 Units.
Over time, technology has occupied an increasing role in education, with mixed results. Massive Open Online Courses (MOOCs) were created to bring education to those without access to universities, yet most of the students who succeed in them are those who are already successful in the current educational model. This course focuses on one intersection of technology and learning: computer games. This course covers education theory, psychology (e.g., motivation, engagement), and game design so that students can design and build an educational learning application. Labs focus on developing expertise in technology, and readings supplement lecture discussions on the human components of education.
Prerequisite(s): CMSC 14300 or CMSC 15400 or CMSC 22000
Equivalent Course(s): CMSC 20900, MADD 20900

CMSC 31010. Mathematical Foundations. 100 Units.
This course is an introduction to formal tools and techniques which can be used to better understand linguistic phenomena. A major goal of this course is to enable students to formalize and evaluate theoretical claims.
Equivalent Course(s): CMSC 21010, LING 21010, LING 31010

CMSC 31140. Computational Imaging: Theory and Methods. 100 Units.
Computational imaging refers to the process of forming images from data where computation plays an integral role. This course will cover basic principles of computational imaging, including image denoising, regularization techniques, linear inverse problems and optimization-based solvers, and data acquisition models associated with tomography and interferometry. Specific topics may include patch-based denoising, sparse coding, total variation, dictionary learning, computational photography, compressive imaging, inpainting, and deep learning for image reconstruction.
Instructor(s): R. Willett Terms Offered: To be determined
Equivalent Course(s): STAT 31140, CAAM 31140

CMSC 31150. Mathematical Toolkit. 100 Units.
The course is aimed at first-year graduate students and advanced undergraduates. The goal of the course is to collect and present important mathematical tools used in different areas of computer science. The course will mostly focus on linear algebra and probability. We intend to cover the following topics and examples: Abstract linear algebra: vector spaces, linear transformations, Hilbert spaces, inner product, Gram-Schmidt orthogonalization, Eigenvalues and eigenvectors, SVD, least squares (under/over-constrained) Discrete probability: random variables, Markov, Chebyshev and Chernoff bounds. Gaussian variables, concentration inequalities, dimension reduction Martingales (time permitting) Stochastic Processes (time permitting) Expected outcomes: Ability to write correctly typed rigorous proofs. Understanding of various notions of linear algebra in the context of abstract vector spaces. Ability to understand and analyze stochastic processes. Familiarity with discrete and continuous random variables and various concentration bounds. Prerequisites: None
Instructor(s): Avrim Blum and Madhur Tulsiani Terms Offered: Autumn
Equivalent Course(s): TTIC 31150

CMSC 31170. Planning, Learning, and Estimation for Robotics and Artificial Intelligence. 100 Units.
This course concerned with fundamental techniques in robotics and artificial intelligence (AI), with an emphasis on probabilistic inference, learning, and planning under uncertainty. The course will investigate the theoretical foundations underlying these topics as rigorous mathematical tools that enable solutions to real-world problems drawn broadly from robotics and AI. The course will cover topics that include: Bayesian filtering (Kalman filtering, particle filtering, and dynamic Bayesian networks), simultaneous localization and mapping, planning, Markov decision processes, partially observable Markov decision processes, reinforcement learning, and graphical models. Expected outcomes: •
Instructor(s): Matthew Walter Terms Offered: Spring
Prerequisite(s): Basic familiarity with basic linear algebra; background in probability theory; basic programming experience
Equivalent Course(s): TTIC 31170

CMSC 31230. Fundamentals of Deep Learning. 100 Units.
Introduction to fundamental principles of deep learning. Deep learning systems are evolving rapidly and this course presents up to date material at a conceptual level. The course emphasizes theoretical and intuitive understanding rather than particular programming formalisms. Topics: Information theory as an organizing principle for machine learning and deep learning in particular. Deep learning frameworks. The "educational framework" (EDF) witten in directly in NumPy. Deep networks for computer vision: Convolutional neural networks (CNNs) and Resnet and the general principles behind them. Deep networks for language processing: Recurrent neural networks (RNNs), the Transformer, their applications and the general principles behind them. The theory and practice of stochastic gradient descent. Regularization and Generalization. Generative Adversarial Networks (GANs) Variational Autoencoders (VAEs) Contrastive Predictive Coding (CPC) Energy Based Models Reinforcement learning and AlphaZero Expected outcomes: An understanding of the general issues sufficient to guide architecture design and training. An ability to read and understand the current research literature in deep learning.
Instructor(s): David McAllester Terms Offered: Autumn
Prerequisite(s): Prerequisites: linear algebra, vector calculus, familiarity with multivariate Gaussian probability distributions and Markov processes.

Equivalent Course(s): TTIC 31230

CMSC 31500. Data Interaction. 100 Units.
This course provides core knowledge and technical skills around data interfaces, with an emphasis on visualization and front-end software development. Graduate students in Data Science and Computer Science will engage in project-based learning to become fluent with visualization APIs, computational notebooks, web development, technical writing, and presentation. Topics of interest include data visualization design, spatial and visual reasoning, cartography, interactive articles, data storytelling, data-driven persuasion, uncertainty communication, and model interpretability.
Instructor(s): A. Kale Terms Offered: Autumn
Prerequisite(s): Consent of Instructor unless graduate student in Data Science
Equivalent Course(s): DATA 31500

CMSC 31801. Topics in Data Science. 100 Units.
Graduate study in current topics in data science.

CMSC 32001. Topics in Programming Languages. 100 Units.
This course covers a selection of advanced topics in programming languages.
Prerequisite(s): Consent of department counselor and instructor

CMSC 32011. Topics in Formal Verification. 100 Units.
Graduate study of topics in formal verification.

CMSC 32200. Computer Architecture. 100 Units.
This course is a survey of contemporary computer organization covering CPU design, instruction sets, control, processors, busses, ALU, memory, pipelined computers, multiprocessors, networking, and case studies. We focus on the techniques of quantitative analysis and evaluation of modern computing systems, such as the selection of appropriate benchmarks to reveal and compare the performance of alternative design choices in system design. We emphasize major component subsystems of high-performance computers: pipelining, instruction-level parallelism, memory hierarchies, input/output, and network-oriented interconnections.
Instructor(s): Hoffmann

CMSC 32201. Topics in Computer Architecture. 100 Units.
This course covers a selection of advanced topics in computer architecture.
Prerequisite(s): Consent of department counselor and instructor

CMSC 32250. Intro to Computer Security. 100 Units.
This course introduces the principles and practice of computer security. It aims to teach how to model threats to computer systems and how to think like a potential attacker. It presents standard cryptographic functions and protocols and gives an overview of threats and defenses for software, host systems, networks, and the Web. It also touches on some of the legal, policy, and ethical issues surrounding computer security in areas such as privacy, surveillance, and the disclosure of security vulnerabilities. The goal of this course is to provide a foundation for further study in computer security and to help better understand how to design, build, and use computer systems more securely.

CMSC 32400. Programming Proofs. 100 Units.
In this course, we will explore the use of proof assistants, computer programs that allow us to write, automate, and mechanically check proofs. These tools have two main uses. They allow us to prove properties of our programs, thereby guaranteeing that our code is free of software errors. They also allow us to formalize mathematics, stating and proving mathematical theorems in a manner that leaves no doubt as to their meaning or veracity. At the intersection of these two uses lies mechanized computer science, involving proofs about data structures, algorithms, programming languages and verification itself.
Prerequisite(s): (CMSC 27100 or CMSC 27130 or CMSC 37000 or CMSC 37110), and (CMSC 14100 or CMSC 15100 or CMSC 16100 or CMSC 22100 or CMSC 22300)
Note(s): This course can count toward the Programming Languages & Systems requirement for the CS Major.
Equivalent Course(s): CMSC 22400

CMSC 32450. Foundations of Programming Languages. 100 Units.
This course is an introduction to the theory of programming languages. It develops the mathematical tools for specifying and reasoning about the static and dynamic semantics of programming languages. The course covers the λ-calculus, which underpins the semantics of many real-world languages, as well as various different techniques for specifying language semantics.
Prerequisite(s): (CMSC 27100 or CMSC 27130 or CMSC 37110) or (CMSC 27200 or CMSC 27230 or CMSC 37000).
Equivalent Course(s): CMSC 22450

CMSC 32700. Quantum Programming and Verification. 100 Units.
As quantum computing comes into its own, a growing number of researchers are trying to address two key challenges that will face computer scientists over the coming years: How do we program quantum computers and how do we ensure that the programs we write reflect our intent? In this course, we will survey a range of papers on quantum programming languages, verification tools, and debugging frameworks for quantum programs. We will identify a variety of abstractions used in quantum programming languages and different approaches to guaranteeing program correctness, including hands-on work in the Coq proof assistant. The

course will culminate in a final project that either develops techniques for quantum programming/verification or applies existing ones to new problems.
Instructor(s): Rand

CMSC 32800. Picturing Quantum Processes. 100 Units.
Quantum Picturalism provides a compelling view of quantum processes as string diagrams. Born from categorical quantum mechanics, which studies the underlying structures of quantum computing, these diagrams convey key concepts like entanglement in a way that is far more intuitive than the dominant state vector formalism. In this course, we will study quantum computing from the perspective of Quantum Picturalism, covering key results in quantum algorithms and the underlying quantum mechanics. We will also cover powerful languages built upon the this framework, like the ZX and ZW calculi, and their applications from quantum key distribution to surfaces codes. This course assumes no background in quantum computing or quantum mechanics.

CMSC 32900. Quantum Computer Systems. 100 Units.
This course will explore the design, optimization, and verification of the software and hardware involved in practical quantum computer systems. The course will provide an introduction to quantum computation and quantum technologies, as well as classical and quantum compiler techniques to optimize computations for technologies. Verification techniques to evaluate the correctness of quantum software and hardware will also be explored.
Prerequisite(s): CMSC 14400 or CMSC 15400, and CMSC 22880 or MENG 26400 or CMSC 38410 or MATH 38410 or CMSC 38420 or MATH 38420
Equivalent Course(s): CMSC 22900

CMSC 33000. Operating Systems. 100 Units.

CMSC 33001. Topics in Systems. 100 Units.
Graduate study of current topics in systems.
Prerequisite(s): Consent of department counselor and instructor

CMSC 33100. Advanced Operating Systems. 100 Units.
This course covers advanced topics in operating systems and systems research. Possible topics include, but are not limited to the following: OS philosophies, networked operating systems, distributed file systems, virtual machines, fault-tolerant systems, resource allocation, parallel computing and multiprocessing, cloud computing, and security.
Instructor(s): Lu
Prerequisite(s): Consent of department counselor and instructor

CMSC 33200. Topics: Operating Systems. 100 Units.
In this class we will discuss the literature that addresses performance and correctness issues in real-world large-scale distributed systems. For hands-on projects, students will pick complex, real-world faults in deployment, reproduce them on an academic cloud, debug and diagnose the issues. Students will also use published tools and evaluate whether they can find or diagnose the faults being reproduced.

CMSC 33210. Usable Security and Privacy. 100 Units.
Regardless of how secure a system is in theory, failing to consider how humans actually use the system leads to disaster in practice. This course will examine how to design for security and privacy from a user-centered perspective by combining insights from computer systems, human-computer interaction (HCI), and public policy. We will introduce core security and privacy technologies, as well as HCI techniques for conducting robust user studies. Topics will include usable authentication, user-centered web security, anonymity software, privacy notices, security warnings, and data-driven privacy tools in domains ranging from social media to the Internet of Things. Students will complete weekly problem sets, as well as conduct novel research in a group capstone project. No prior experience in security, privacy, or HCI is required.
Prerequisite(s): CMSC 12300 or CMSC 14200 or CMSC 15400
Equivalent Course(s): CMSC 23210

CMSC 33211. Topics in Privacy. 100 Units.
Graduate study of current topics in privacy.

CMSC 33218. Surveillance Aesthetics: Provocations About Privacy and Security in the Digital Age. 100 Units.
In the modern world, individuals' activities are tracked, surveilled, and computationally modeled to both beneficial and problematic ends. Jointly with the School of the Art Institute of Chicago (SAIC), this course will examine privacy and security issues at the intersection of the physical and digital worlds. Through both computer science and studio art, students will design algorithms, implement systems, and create interactive artworks that communicate, provoke, and reframe pervasive issues in modern privacy and security. The course will unpack and re-entangle computational connections and data-driven interactions between people, built space, sensors, structures, devices, and data. Synthesizing technology and aesthetics, we will communicate our findings to the broader public not only through academic avenues, but also via public art and media. The first phase of the course will involve prompts in which students design and program small-scale artworks in various contexts, including (1) data collected from web browsing; (2) mobility data; (3) data collected about consumers by major companies; and (4) raw sensor data. Students will receive detailed feedback on their work

from computer scientists, artists, and curators at the Museum of Science & Industry (MSI). The course culminates in the production and presentation of a capstone interactive artwork by teams of computer scientists and artists; successful products may be considered for prototyping at the MSI.
Prerequisite(s): One of CMSC 23200, CMSC 23210, CMSC 25900, CMSC 28400, CMSC 33210, CMSC 33250, or CMSC 33251 recommended, but not required.
Note(s): Students interested in this class should complete this form to request permission to enroll: https://uchicago.co1.qualtrics.com/jfe/form/SV_5jPT8gRDXDKQ26a
Equivalent Course(s): CMSC 23218, MADD 23218

CMSC 33220. A Practice in Art and Technology. 100 Units.
This graduate-level class, "A&T" for short, offers project-based learning through 3 phases of interactive art development: 1. Ideate (coming up with unique and original ideas), 2. Build (implementing the idea with technical methods), and 3. Exhibit (facilitating the prototypes to convey experiences to people). The instructor will provide continuous weekly feedback to the students' project to develop technically working, interactively engaging, and conceptually provocative systems that will be showcased in a public exhibition at the end of the quarter.
Instructor(s): Nakagaki

CMSC 33221. Advanced Topics in Law and Computing. 100 Units.
This interdisciplinary seminar will bring together instructors and graduate students from Computer Science / Data Sciences and the Law School. The seminar's focus will be on topics where law and policy intersect with computer science. Such topics may include cryptography and encryption; electronic surveillance and criminal procedure; the Computer Fraud & Abuse Act; the law governing data breaches; redistricting and the US Census; deep fakes; GDRP, Europe's Digital Services Act and the CCPA; and international data transfers. Students will be evaluated on the basis of short bi-weekly reaction papers, class participation based on weekly assigned reading, and team projects that pair law students with computer and data scientists.
Equivalent Course(s): DATA 33221

CMSC 33230. Engineering Interactive Electronics onto Printed Circuit Boards. 100 Units.
In this class we will engineer electronics onto Printed Circuit Boards (PCBs). We will focus on designing and laying out the circuit and PCB for our own custom-made I/O devices, such as wearable or haptic devices. In order for you to be successful in engineering a functional PCB, we will (1) review digital circuits and three microcontrollers (ATMEGA, NRF, SAMD); (2) use KICAD to build circuit schematics; (3) learn how to wire analog/digital sensors or actuators to our microcontroller, including SPI and I2C protocols; (4) use KICAD to build PCB schematics; (5) actually manufacture our designs; (6) receive in our hands our PCBs from factory; (7) finally, learn how to debug our custom-made PCBs.
Prerequisite(s): CMSC 14400 or CMSC 15400, and CMSC 23220 or CMSC 21400 or CMSC 20380 or PHYS 13300 or PHYS 14300 or PHYS 22700.
Equivalent Course(s): CMSC 23230

CMSC 33231. Topics in Human Computer Interaction. 100 Units.
Graduate study in current topics in Human Computer Interaction.
Instructor(s): Pedro Lopes

CMSC 33240. Emergent Interface Technologies. 100 Units.
In this class, we critically examine emergent technologies that might impact the future generations of computing interfaces, these include: physiological I/O (e.g., brain and muscle computer interfaces), tangible computing (giving shape and form to interfaces), wearable computing (I/O devices closer to the user's body), rendering new realities (e.g., virtual and augmented reality), haptics (giving computers the ability to generate touch and forces) and unusual auditory interfaces (e.g., silent speech and microphones as sensors). In this class you will: (1) learn about these new developments during the lectures, (2) read HCI papers and summarize these in short weekly assignments, and lastly, (3) start inventing the future of computing interfaces by proposing a new idea in the form of a paper abstract, which you will present at the end of the semester and have it peer-reviewed in class by your classmates.

CMSC 33250. Introduction to Computer Security. 100 Units.
This course introduces the principles and practice of computer security. It aims to teach how to model threats to computer systems and how to think like a potential attacker. It presents standard cryptographic functions and protocols and gives an overview of threats and defenses for software, host systems, networks, and the Web. It also touches on some of the legal, policy, and ethical issues surrounding computer security in areas such as privacy, surveillance, and the disclosure of security vulnerabilities. The goal of this course is to provide a foundation for further study in computer security and to help better understand how to design, build, and use computer systems more securely.
Prerequisite(s): CMSC 14400 or CMSC 15400
Equivalent Course(s): CMSC 23200

CMSC 33251. Topics in Computer Security. 100 Units.
Seminar on current topics in computer security.

CMSC 33260. Internet Censorship and Online Speech. 100 Units.
Information dissemination and online discourse on the Internet are subject to the algorithms and filters that operate on Internet infrastructure, from network firewalls to search engines. This course will explore the technologies that are used to control access to online speech and information, and cutting-edge technologies that can empower citizens in the face of these information controls. Students will learn about and experiment with technologies to control online discourse, ranging from firewalls that perform network traffic filtering to algorithms for content personalization and content moderation. We will also explore underlying technical trends, such as the increasing consolidation of Internet infrastructure and protocols, and the implications of consolidation for control over online discourse. Each course meeting will include a technical overview, reading discussion, and a hands-on laboratory activity.
Prerequisite(s): None
Equivalent Course(s): MADD 23620, PARR 33260

CMSC 33281. Topics in Human Robot Interaction. 100 Units.
The field of human-robot interaction (HRI) is a new and growing field of research that explores the interface between people and robots. Applications of HRI research include developing robots to tutor elementary students, assist human workers in manufacturing contexts, provide museum tours, interact with families within their homes, and help care for the elderly. The field of HRI is highly interdisciplinary, incorporating methods and techniques from human-computer interaction, robotics, psychology, artificial intelligence, and other fields. The University of Chicago's CMSC 33281 Topics in Human-Robot Interaction course exposes students to a broad range of recent and cutting-edge research in HRI. The topics covered in this course include nonverbal robot behavior, verbal robot behavior, social dynamics, norms & ethics, collaboration & learning, group interactions, applications, and future challenges of HRI. Course meetings will involve students in the class leading discussions about cutting-edge peer-reviewed research HRI publications. Throughout the quarter, teams of students in the course will also complete an HRI course project of their choosing where they will investigate an HRI research question of interest to them.
Equivalent Course(s): MADD 23281

CMSC 33300. Networks and Distributed Systems. 100 Units.
This course will focus on studying the state of the art in networking and networked systems, from a research and design perspective. We will cover a variety of topics from routing protocols to Internet stability, peer-to-peer, social networks and networking for data centers. Coverage of each topic will dive into fundamental design questions of protocols and systems, including updates from results of currently active research. Readings will focus on classic and current research publications, and students are expected to come in with a solid background on networking basics. Students will learn tools, techniques, and concepts while learning to carry out original research in an open-ended course project, with the end goal of producing real, publishable results by the end of the quarter. Students are also expected to gain experience in two skills: quickly reading technical papers (without sacrificing understanding), and giving clear and well-organized presentations.

CMSC 33400. Mobile Computing. 100 Units.
Mobile computing is pervasive and changing nearly every aspect of society. Sensing, actuation, and mediation capabilities of mobile devices are transforming all aspects of computing: uses, networking, interface, form, etc. This course explores new technologies driving mobile computing and their implications for systems and society. Current focus areas include expanded visual experience with computational photography, video and interactive augmented reality, and synchronicity and proximity-detection to enable shared social experiences. Labs expose students to software and hardware capabilities of mobile computing systems, and develop the capability to envision radical new applications for a large-scale course project.
Instructor(s): A. Chien Terms Offered: Not offered 2017-2018.
Prerequisite(s): CMSC 23000 or 23300 or equivalent are required.

CMSC 33501. Topics in Databases. 100 Units.
This course covers a selection of advanced topics in database systems.
Prerequisite(s): Consent of department counselor and instructor

CMSC 33520. Data Intensive Computer Systems. 100 Units.
Cloud computing has become the primary means for large-scale computation and services to be delivered. This class covers a wide range of cloud software elements that both underpin and shape the design of scalable internet data services. The success of cloud computing depends on efficient resource sharing and application isolation -- typically supported by virtual machines (VMs) and containers, controlled by orchestration infrastructures such as VMWare and Kubernetes. We will cover these models, and the oversubscription resource management approaches that sustain them, as well as the sustainability and carbon-emission challenges that limit them. Newer application architectures such as micro-services and function-as-a-service (FaaS or Serverless) enable easy application design evolution. We will cover these models and the challenges they present for implementation. These software structures support "big data" applications, including analytical systems such as map-reduce and Spark. They also include data serving systems, enable big data, and support these systems including parallel filesystems, databases, and noSQL key-value stores such as Cassandra, Memcached, MongoDB, and more. Students will develop a broad familiarity with current challenges, the state of the art, including leading edge research in the area, and hands-on experience with a range of systems which together provide a solid preparation for research in the area.

Prerequisite(s): Consent of department counselor and instructor

CMSC 33550. Introduction to Database Systems. 100 Units.
This course is an introduction to database design and implementation. Topics include DBMS architecture, entity-relationship and relational models, relational algebra, concurrency control, recovery, indexing, physical data organization, and modern database systems. The lab section guides students through the implementation of a relational database management system, allowing students to see topics such as physical data organization and DBMS architecture in practice, and exercise general skills such as software systems development.
Prerequisite(s): CMSC 14400 or CMSC 15400
Equivalent Course(s): CMSC 23500

CMSC 33580. The Value of Data. 100 Units.
While data and artificial intelligence are driving many changes to our economic, social, political, financial, and legal systems, we know little about their foundations and governing dynamics. While the recombination and integration of diverse data creates vast new value, we currently have neither theory for how data can be combined nor an industrial policy for how to protect against the personal exposures and abuses that grow in proportion. Many of these issues call for treating data as a first-class citizen and thinking of it as an asset. What is the value of data, and how do we measure it? This course explores these questions from a multidisciplinary perspective that includes computer science, economy, social science, among others. The sessions are organized around broad themes related to the value of data. A typical session will consist of a short introduction by the instructor, followed by a paper discussion (students are expected to read the assigned papers ahead of the class), and finishing with a discussion about individual projects. The course can be taken pass-fail or for a letter grade. For a letter grade, you need to complete an individual, quarter-long project. You also need to attend classes, participate in discussions, and scribe for two/three classes. For pass-fail, you do not need to complete the individual project but all other requirements apply.

CMSC 33581. Topics in Big Data. 100 Units.
The amount of data generated is growing faster than our computational infrastructure, and ever-smarter algorithms will be needed to structure, analyze, and manage such scales. This 10-week seminar studies key "great ideas" in scalable algorithms for data analysis. Topics include: sampling, sketching, randomized index structures, basic coding theory, and query optimization.

CMSC 33700. Introduction to Computer Graphics. 100 Units.
This course introduces the basic concepts and techniques used in three-dimensional computer graphics. The focus is on real-time rendering techniques, such as those found in computer games. The course places an emphasis on developing a strong foundation for 3D graphics by covering topics such as the graphics pipeline, coordinate systems, coordinate transformations, lighting, texture mapping, and basic geometric algorithms and data structures. These foundations are applied using programmable shaders to implement real-time rendering techniques, such as real-time shadows and other effects. Students are required to complete both written assignments and programming projects using C++ and the Vulkan 3D graphics library.
Prerequisite(s): CMSC 14400 or CMSC 15400
Note(s): Prior experience with basic linear algebra (matrix algebra) is recommended.
Equivalent Course(s): CMSC 23700

CMSC 33710. Scientific Visualization. 100 Units.
Scientific visualization combines computer graphics, numerical methods, and mathematical models of the physical world to create a visual framework for understanding and solving scientific problems. The mathematical and algorithmic foundations of scientific visualization (for example, scalar, vector, and tensor fields) will be explained in the context of real-world data from scientific and biomedical domains. The course is also intended for students outside computer science who are experienced with programming and computing with scientific data. Programming projects will be in C and C++.
Prerequisite(s): CMSC 14300 or CMSC 15400
Equivalent Course(s): CMSC 23710

CMSC 33750. Machine Learning and Cancer. 100 Units.
In this topics course we will investigate the use of machine learning methods in the study of Cancer and the development of precision oncology. Cancer is a complex disease that impacts millions each year. Recently the concept of precision oncology has gained popularity as an approach to customize Cancer treatments based on the genomic profile and history of the patient, the molecular properties of the patient's tumor and the action and mode of treatments that are available. At the center of any precision medicine approach are large-scale datasets from which predictive models can be built, scalable analysis methods for processing and integrating data and machine learning methods for constructing and evaluating predictive models that can be used in diagnosis, treatment planning, and outcome prediction for patient care. In this course we will work through the development of the entire pipeline from raw data to predictive models. We will develop and evaluate predictive models for drug response, tumor typing, image based diagnosis, and treatment outcomes. We will also develop some population based models that include environmental factors. Students will work through key papers, representative datasets and a variety of machine learning methods including some deep learning models under development in the joint DOE/NCI Cancer project. Familiarity with python and machine learning will be helpful. Students will have an opportunity to do significant project work as part of the course.

CMSC 33900. Data Visualization. 100 Units.
Data visualizations provide a visual setting in which to explore, understand, and explain datasets. This class describes mathematical and perceptual principles, methods, and applications of "data visualization" (as it is popularly understood to refer primarily to tabulated data). A range of data types and visual encodings will be presented and evaluated. Visualizations will be primarily web-based, using D3.js, and possibly other higher-level languages and libraries.

CMSC 34702. Topics in Networks: 100 Units.

CMSC 34703. Topics in Distributed Systems. 100 Units.
Topics in Distributed Systems

CMSC 34900. Topics in Scientific Computing. 100 Units.
This course covers a selection of advanced topics in Scientific Computing.
Prerequisite(s): Consent of department counselor and instructor

CMSC 34901. Special Topics in Operations Mgt./Mgt. Science. 100 Units.
Course Search (https://intranet.chicagobooth.edu/pub/coursesearch/coursesearch/)
Equivalent Course(s): BUSN 40901

CMSC 35050. Graduate Computational Linguistics. 100 Units.
This course is a graduate-level introduction to topics at the intersection of computation and language. We will study computational linguistics from both scientific and engineering angles: the use of computational modeling to address scientific questions in linguistics and cognitive science, as well as the design of computational systems to solve engineering problems in natural language processing (NLP). The course will combine analysis and discussion of these approaches with training in the programming and mathematical foundations necessary to put these methods into practice. The course is designed to accommodate students both with and without prior programming experience. Our goal is for all students to leave the course able to engage with and critically evaluate research in cognitive/linguistic modeling and NLP, and to be able to implement intermediate-level computational models for novel computational linguistics research.
Instructor(s): J. Goldsmith Terms Offered: Spring
Prerequisite(s): CMSC 12200, 15200 or 16200, or by consent
Equivalent Course(s): LING 38600

CMSC 35100. Natural Language Processing. 100 Units.
This course introduces the theory and practice of natural language processing, with applications to both text and speech. Topics include regular expressions, finite state automata, morphology, part of speech tagging, context free grammars, parsing, semantics, discourse, and dialogue. Symbolic and probabilistic models are presented. Techniques for automatic acquisition of linguistic knowledge are emphasized.

CMSC 35110. Speech Technologies. 100 Units.
This course will introduce techniques used in speech technologies, mainly focusing on speech recognition. Speech recognition is one of the oldest and most complex structured sequence prediction tasks receiving significant research and commercial attention, and therefore provides a good case study for many of the techniques that are used in other areas of artificial intelligence involving sequence modeling. It is also a good example of the effectiveness of combining statistics and learning with domain knowledge. The course will include practical homework exercises using Matlab and speech toolkits. Expected outcomes: Understand and apply tools for analyzing speech time series such as Fourier analysis and dynamic time warping. Understand and apply hidden Markov models, Gaussian mixtures, and the EM algorithm for speech problems. Understand and apply n-gram language models, smoothing techniques, and their application to speech recognition. Understand generative and discriminative structured prediction approaches for speech problems.
Instructor(s): Karen Livescu Terms Offered: Spring
Prerequisite(s): A good background in basic probability.
Equivalent Course(s): TTIC 31110

CMSC 35200. Deep Learning Systems. 100 Units.
Deep learning is emerging as a major technique for solving problems in a variety of fields, including computer vision, personalized medicine, autonomous vehicles, and natural language processing. Critical to success in these target domains is the development of learning systems: deep learning frameworks that support the tasks of learning complex models and inferencing with those models, and targeting heterogeneous computing devices. This course is aimed as an introduction to this topic. We will cover various aspects of deep learning systems, including: basics of deep learning, programming models for expressing machine learning models, automatic differentiation methods used to compute gradients for training, memory optimization, scheduling, data and model parallel and distributed learning, hardware acceleration, domain specific languages, workflows for large-scale machine learning including hyper parameter optimization and uncertainty quantification, and training data and model serving. The goal is to present a comprehensive picture of how current deep learning systems work, discuss and explore research opportunities, for extending and building on existing frameworks, and deep dive into the accelerators being developed by numerous startups to address the needs of the machine learning community. A typical week will contain one lecture on a specific aspect of deep learning systems and one lab session exploring technologies such as Keras, Tensorflow, CNTK, Mxnet, and PyTorch.

Note(s): This course will provide useful background for students wishing to take our Spring 2019 class on Neuromorphic Computing.

CMSC 35230. Applications of Machine Learning in Large-Scale Computing Systems. 100 Units.
Recent research has demonstrated the feasibility of replacing the traditional heuristics used in computer systems with ones learned from data in areas such as scheduling, data structure design, query optimization, compilers, and control of warehouse scale computing systems. This seminar overviews this recent research trend and studies the characteristics of successful research in this space. The objective is to understand whether this trend will have a long-term impact on the design and implementation of large-scale computing systems. The seminar hopes to highlight the remaining bottlenecks to practical acceptance of machine learning in computing infrastructure and to inspire future systems and learning research.
Prerequisite(s): Good working knowledge of computer systems, especially in the areas of operating systems and databases.

CMSC 35246. Deep Learning. 100 Units.
Deep Neural Networks are remarkably effective in large scale learning problems, especially in speech recognition and computer vision. This course aims to cover the basics of Deep Learning, some of the underlying theory, and specific architectures, including Convolutional Neural Networks, Recurrent Neural Networks and the Long Short Term Memory Networks.

CMSC 35300. Mathematical Foundations of Machine Learning. 100 Units.
This course is an introduction to the mathematical foundations of machine learning that focuses on matrix methods and features real-world applications ranging from classification and clustering to denoising and data analysis. Mathematical topics covered include linear equations, regression, regularization, the singular value decomposition, and iterative algorithms. Machine learning topics include classification and regression, support vector machines, kernel methods, clustering, matrix completion, neural networks, and deep learning. Students are expected to have taken calculus and have exposure to numerical computing (e.g. Matlab, Python, Julia, R).
Prerequisite(s): CMSC 11900 or CMSC 12200 or CMSC 14100 or CMSC 15200 or CMSC 16200
Note(s): Undergraduate students are not allowed to enroll in CMSC 35300.
Equivalent Course(s): CMSC 25300, STAT 27700

CMSC 35350. Neuromorphic Computing. 100 Units.
The human brain consumes around 20 watts-less energy than a lightbulb-but can perform tasks, such as understanding natural language and interpreting images, that tax megawatt-scale supercomputers. Thus, we may wonder: can we achieve energy efficiencies similar to those of the human brain by building analog electronic circuits that mimic the neuro-biological architectures found in animal nervous systems? This concept, named neuromorphic computing, has become increasingly popular as the energy demands of conventional computers increase. Research in this area is producing a great variety of new computational architectures, microelectronics concepts, algorithmic approaches, and even neuroscience insights. Our goal in this course is to introduce students to the state of knowledge in neuromorphic computing and thus to prepare them to undertake original research in this area. The class will be organized primarily around reading, presenting, and discussing research papers. Topics to be covered include: Neuromorphic concepts Theory and algorithms Microelectronics and devices Programming models and environments Applications: Machine learning, deep learning, robot control, ... Platforms: BrainScaleS, Loihi, SpiNNaker Neuroscience
Terms Offered: Spring TBD

CMSC 35360. Autonomous Laboratories. 100 Units.
Autonomous laboratories leverage robotics, machine learning (ML), and artificial intelligence (AI) methods to accelerate discovery and design processes in science and engineering. They build on advances in robotic instruments, which enable faster, cheaper, and more reproducible experiments, but go further, leveraging AI/ML methods to explore large and complex search spaces more efficiently than humans can do unaided. This course introduces students to the principles and practice of automated laboratories, with a particular focus on AI/ML methods such as active learning, Bayesian optimization, and reinforcement learning. The class will combine lectures, readings, programming assignments on AI/ML methods, and a project in which students will investigate an autonomous laboratory problem.
Instructor(s): Foster, Stevens

CMSC 35380. Programming the Continuum. 100 Units.
In the early days of high-speed networking, George Gilder observed that "when the network is as fast as the computer's internal links, the machine disintegrates across the net into a set of special purpose appliances." 2+ decades later, our networks are 1,000 times faster, our appliances are increasingly specialized, and our computer systems are indeed disintegrating. As hardware acceleration overcomes speed-of-light delays, time and space merge into a computing continuum. Familiar questions like "where should I compute," "for what workloads should I design computers," and "where should I place my computers" seem to allow for exciting new answers. Are there concepts that can help guide us as we design applications and computer systems in a world that is untethered from familiar landmarks like center, cloud, edge? In this class we will discuss papers on these and related topics; hear from visionary leaders in the field; and conduct projects that allow for practical exploration of concepts and techniques. The class is graduate level but is also open to qualified upper-level undergraduates.

CMSC 35400. Machine Learning. 100 Units.
This course provides hands-on experience with a range of contemporary machine learning algorithms, as well as an introduction to the theoretical aspects of the subject. Topics covered include: the PAC framework, Bayesian learning, graphical models, clustering, dimensionality reduction, kernel methods including SVMs, matrix completion, neural networks, and an introduction to statistical learning theory.
Terms Offered: To be determined
Prerequisite(s): Must be a PhD or MS student in Statistics, Computer Science, or Computational and Applied Mathematics, and has taken any one of: CMSC 35300/STAT 27700, STAT 31430, STAT 30900, STAT 24300, STAT 24500, or STAT 24510. Or consent of the instructor.
Equivalent Course(s): CAAM 37710, STAT 37710

CMSC 35401. Topics in Machine Learning: Applied Machine Learning. 100 Units.
Graduate study in current topics in machine learning.

CMSC 35410. Spectral Methods for Machine Learning and Network Analysis. 100 Units.
An introduction to spectral algorithms, emphasizing their power to tackle practical problems in the analysis of networks and high-dimensional data. Topics include spectral graph theory, random walks over networks and their convergence, spectral clustering, subspace projections and embeddings, and numerical algorithms for fundamental linear-algebraic problems.
Instructor(s): Lorenzo Orecchia

CMSC 35422. Machine Learning for Computer Systems. 100 Units.
This course will cover topics at the intersection of machine learning and systems, with a focus on applications of machine learning to computer systems. Topics covered will include applications of machine learning models to security, performance analysis, and prediction problems in systems; data preparation, feature selection, and feature extraction; design, development, and evaluation of machine learning models and pipelines; fairness, interpretability, and explainability of machine learning models; and testing and debugging of machine learning models. The topic of machine learning for computer systems is broad. Given the expertise of the instructor, many of the examples this term will focus on applications to computer networking. Yet, many of these principles apply broadly, across computer systems. You can and should think of this course as a practical hands-on introduction to machine learning models and concepts that will allow you to apply these models in practice. We'll focus on examples from networking, but you will walk away from the course with a good understanding of how to apply machine learning models to real-world datasets, how to use machine learning to help computer systems operate better, and the practical challenges with deploying machine learning models in practice."
Instructor(s): Nick Feamster
Prerequisite(s): CMSC 14300 or CMSC 15400
Equivalent Course(s): DATA 25422, CMSC 25422, DATA 35422

CMSC 35425. Topics in Statistical Machine Learning. 100 Units.
Topics in Statistical Machine Learning" is a second graduate level course in machine learning, assuming students have had previous exposure to machine learning and statistical theory. The emphasis of the course is on statistical methodology, learning theory, and algorithms for large-scale, high dimensional data. The selection of topics is influenced by recent research results, and students can take the course in more than one quarter.
Terms Offered: To be determined
Equivalent Course(s): STAT 37790

CMSC 35430. Machine Learning on Graphs, Groups and Manifolds. 100 Units.
In many domains, including applications of machine learning to scientific problems, social phenomena and computer vision/graphics, the data that learning algorithms operate on naturally lives on structured objects such as graphs or low dimensional manifolds. There are many connections between these cases; further, since groups capture symmetries, there are also natural connections to the theory of learning on groups and group equiviariant algorithms. This course provides a mathematical introduction to these topics both in the context of kernel based learning and neural networks. Specific topics covered include graph kernels, manifold learning, graph wavelets, graph neural networks, permutation equivariant learning, rotational equivariant networks for scientific applications and imaging, gauge equivariant networks and steerable nets.
Equivalent Course(s): STAT 37788, CAAM 37788

CMSC 35440. Machine Learning in Biology and Medicine. 100 Units.
[Machine Learning in Biology and Medicine]
Note(s): [still awaiting course description from instructors]

CMSC 35450. Geometric Deep Learning. 100 Units.
This course will cover the basics of learning on unstructured geometric data. In particular, the course will focus on learning on 3D data representations (such as meshes and point clouds). The course is project-based, hands-on, and practical. Students will implement, design, and train deep neural networks using Python and PyTorch. Students should have familiarity with fundamental/basic machine learning.

CMSC 35470. Mathematical Computation IIA: Convex Optimization. 100 Units.
The course will cover techniques in unconstrained and constrained convex optimization and a practical introduction to convex duality. The course will focus on (1) formulating and understanding convex optimization problems and studying their properties; (2) understanding and using the dual; and (3) presenting and

understanding optimization approaches, including interior point methods and first order methods for non-smooth problems. Examples will be mostly from data fitting, statistics and machine learning.
Instructor(s): Zhiyuan Li Terms Offered: Winter
Prerequisite(s): STAT 30900 or STAT 31430 or consent of instructor.
Note(s): In addition to the required prerequisites, background in analysis in R^n (at the level of MATH 20400) is recommended.
Equivalent Course(s): BUSN 36903, CAAM 31015, STAT 31015, TTIC 31070

CMSC 35480. Topics in Optimization. 100 Units.
Graduate study of topic in optimization.

CMSC 35490. Special Topics in Machine Learning. 100 Units.
Learned emulators leverage neural networks to increase the speed of physics simulations in climate models, astrophysics, high-energy physics, and more. Recent empirical results have illustrated that these emulators can speed up traditional simulations by up to eight orders of magnitude. However, little is understood about these emulators. While it is possible that recent results are representative of what is possible in most settings, a more likely scenario is that these approaches are more effective for some simulators than others, and that learned emulators achieve strong average-case performance but fail to capture rare but important phenomena. In this graduate seminar course we will provide an overview and investigate recent literature on this topic, focusing on the following questions: 1. Introduction to learned emulators: how do they work, where have they been successful so far and what are the goals in this field? 2. Two different paradigms of learned emulation: physics vs. data driven. What are the advantages and pitfalls of each? 3. Robustness of emulation to noise: what is known so far? 4. Parameter estimation: how to handle parameter uncertainty? We will provide a list of papers covering the above topics and students will be evaluated on in-class presentations.
Instructor(s): Dana Mendelson (Math) and Rebecca Willett (CS/Stats) Terms Offered: Autumn
Prerequisite(s): Students should be familiar with a numerical programming language like Python, Julia, R, or Matlab and the content of CMSC 35400. Students should also have familiarity with the contents of MATH 27300 and MATH 27500 or similar.
Note(s): Because this is a seminar course, it will be capped at 15 students, 4 Math, 4 CS/Stats, and 7 with instructor permission.
Equivalent Course(s): STAT 37794, CAAM 37794, MATH 37794

CMSC 35530. Fundamentals of Sampled Data Systems. 100 Units.
Virtually all of modern data science, machine learning, and statistics relies on data which has been sampled: converted from continuous values in continuous time to discretely sampled, quantized data. This class will cover the signal processing fundamentals of sampled data systems, including signal representations, signal transforms (such as the Fourier and Z-transforms), fundamentals of sampling theory, models of quantization error, filtering, fast algorithms (including the FFT), basic communications theory, and then briefly touch on modern ideas of sparsity and overcomplete representations. Prerequisites include multivariable calculus, probability, and linear algebra. There will be weekly readings, problem sets, and quizzes.

CMSC 35600. Image Processing/Computer Vision. 100 Units.
Equivalent Course(s): MPHY 39600

CMSC 35620. Computational Linguistics. 100 Units.
This course is a mixed level introduction to topics at the intersection of computation and language. We will study computational linguistics from both scientific and engineering angles: the use of computational modeling to address scientific questions in linguistics and cognitive science, as well as the design of computational systems to solve engineering problems in natural language processing (NLP). The course will combine analysis and discussion of these approaches with training in the programming and mathematical foundations necessary to put these methods into practice. The course is designed to accommodate students both with and without prior programming experience. Our goal is for all students to leave the course able to engage with and critically evaluate research in cognitive/linguistic modeling and NLP, and to be able to implement intermediate-level computational models for novel computational linguistics research.
Instructor(s): Allyson Ettinger Terms Offered: Autumn
Equivalent Course(s): LING 38620, COGS 22015, LING 28620

CMSC 35900. Topics in Artificial Intelligence. 100 Units.
Graduate study of current topics in artificial intelligence.
Terms Offered: Autumn Spring Winter
Prerequisite(s): Consent of department counselor and instructor

CMSC 36500. Algorithms in Finite Groups. 100 Units.
We consider the asymptotic complexity of some of the basic problems of computational group theory. The course demonstrates the relevance of a mix of mathematical techniques, ranging from combinatorial ideas, the elements of probability theory, and elementary group theory, to the theories of rapidly mixing Markov chains, applications of simply stated consequences of the Classification of Finite Simple Groups (CFSG), and, occasionally, detailed information about finite simple groups. No programming problems are assigned.
Prerequisite(s): Consent of department counselor. Linear algebra, finite fields, and a first course in group theory (Jordan-Holder and Sylow theorems) required; prior knowledge of algorithms not required

Note(s): This course is offered in alternate years.
Equivalent Course(s): MATH 37500

CMSC 37000. Algorithms. 100 Units.
This is a graduate level course on algorithms with the emphasis on central combinatorial optimization problems and methods for algorithm design and analysis. Topics covered include greedy algorithms, dynamic programming, algorithms for maximum flow and minimum cut, applications of linear programming, randomized algorithms, combinatorial optimization, and approximation algorithms. Time permitting, additional topics, such as online algorithms and probabilistic method will be covered. The course textbook is "Algorithm Design" by Kleinberg and Tardos Specific topics covered: Greedy algorithms Dynamic programming Max flow, min cut, bipartite matching, and their applications Linear programming, LP-duality NP-hardness Approximation algorithms Randomized algorithms (optional): online algorithms; probabilistic method Expected outcomes: Ability to design and rigorously analyze algorithms using paradigms such as greedy or dynamic programming. Understand the use of linear programming in optimization. Be able to formulate problems as linear programs. Knowledge of maximum flow / minimum cut problems and algorithms for solving them, and ability to apply this knowledge to suitable graph optimization problems. Understand the notion of NP-hardness and ability to prove NP-hardness of problems via reductions. Understand linear programming duality and applications to problems such as max- flow/min-cut. Be able to write duals for linear programs.
Instructor(s): Julia Chuzhoy Terms Offered: Autumn
Prerequisite(s): Assumes familiarity with proofs and formal reasoning, knowledge of basic graph notions (such as graphs, trees, paths etc) and algorithms (such as BFS, DFS, Minimum Spanning Tree etc). Also assumes familiarity with asymptotic notation and running time analysis of algorithms, as well as basic knowledge of the notion of NP-hardness.
Equivalent Course(s): TTIC 31010

CMSC 37110. Discrete Mathematics. 100 Units.
This course emphasizes mathematical discovery and rigorous proof, illustrated on a variety of accessible and useful topics, including basic number theory, asymptotic growth of sequences, combinatorics and graph theory, discrete probability, and finite Markov chains. This course includes an introduction to linear algebra.
Instructor(s): L. Babai
Prerequisite(s): Consent of department counselor and instructor

CMSC 37115. Introduction to Mathematical Reasoning via Discrete Mathematics. 100 Units.
In this course, students with little prior exposure to rigorous mathematical reasoning gain experience in approaching mathematical questions, developing concepts, formalizing ideas, turning intuition into rigorous proof. These phases of mathematical thinking are illustrated on a variety of accessible and useful topics. Students practice the quantifier notation both as a shorthand and as one of the organizing principles of formal statements. New concepts are built from such basic mathematical primitives as numbers, sets, and functions. Basic counting is a recurring theme and provides a source for sequences, another recurring theme, which in turn feeds into the study of asymptotic behavior (rates of growth). Further topics to be covered include proof by induction; the elements of number theory (gcd, congruences, the Chinese Remainder Theorem, Fermat's little Theorem); recurrences, Fibonacci numbers, generating functions; the elements of graph theory (trees, paths and cycle, chromatic number, independent sets and cliques, connectivity, planarity, directed graphs), finite probability spaces, random variables, expected value and variance, independence, concentration inequalities, and random graphs.
Prerequisite(s): One quarter of calculus

CMSC 37200. Combinatorics. 100 Units.
Methods of enumeration, construction, and proof of existence of discrete structures are discussed. The course emphasizes applications of linear algebra, number theory, and the probabilistic method to combinatorics. Applications to the theory of computing are indicated, and open problems are discussed.
Instructor(s): L. Babai
Prerequisite(s): Consent of department counselor. Linear algebra, basic combinatorics, or consent of instructor.

CMSC 37220. Information and Coding Theory. 100 Units.
This course is meant to serve as an introduction to some basic concepts in information theory and error-correcting codes, and some of their applications in computer science and statistics. We plan to cover the following topics: Introduction to entropy and source coding. Some applications of entropy to counting problems. Mutual information and KL-divergence. Method of types and hypothesis testing. I-projections and applications. Introduction to error-correcting codes. Unique and list decoding of Reed-Solomon and Reed-Muller codes. Applications of information theory to lower bounds in computational complexity and communication complexity. Expected outcomes: Familiarity with concepts such as Entropy, Mutual information and KL-divergence. Familiarity with source and channel coding. Understanding of the method of types and ability to derive large-deviation bounds using information-theoretic concepts. Understanding of the notions of unique and list decoding for various codes.
Instructor(s): Tulsiani, Madhur Terms Offered: Winter
Prerequisite(s): Discrete probability. Some knowledge of finite-field algebra is required for the part on error-correcting codes but required basics are reviewed in class.
Equivalent Course(s): TTIC 31200

CMSC 37503. Approximation Algorithms. 100 Units.
This is a basic course on approximation algorithms, with the main focus on approximation algorithms for central combinatorial optimization problems. We will mostly focus on classical algorithmic results, but will also present some state of the art results and challenges in the area of approximation. The course will cover major algorithmic techniques, including LP-rounding, primal-dual schema, metric methods, SDP rounding and so on. While the main focus of the course is on algorithms, we will also discuss lower bounds on approximation and connections between algorithm design and lower bound proofs. Assumes the knowledge of material covered in the Algorithms course. Expected outcomes: Understand concepts such as approximation factor, polynomial time approximation schemes and hardness of approximation. Understand applications of linear programs (LPs) to design of approximation algorithms. Learn to analyze rounding algorithms for LPs and understand integrality gaps. Be able to apply LP duality. Understand semi-definite programming and its applications to approximation.
Instructor(s): Julia Chuzhoy Terms Offered: Autumn
Equivalent Course(s): TTIC 31080

CMSC 37530. Graph Theory. 100 Units.
This course covers the basics of the theory of finite graphs. Topics include shortest paths, spanning trees, counting techniques, matchings, Hamiltonian cycles, chromatic number, extremal graph theory, Turan's theorem, planarity, Menger's theorem, the max-flow/min-cut theorem, Ramsey theory, directed graphs, strongly connected components, directly acyclic graphs, and tournaments. Techniques studied include the probabilistic method.
Instructor(s): Laszlo Babai

CMSC 37712. Foundations of Machine Learning and AI - Part II. 100 Units.
Deep generative models have become a staple of modern machine learning research. This course is meant as an introduction to the way generative models are structured and trained: students will learn the mechanics of generative models as well as getting their hands dirty building them. We will discuss open questions for which we lack complete theoretical or empirical answers, with importance placed on analyzing, interpreting, and making arguments from necessarily incomplete empirical evidence. We will have a specific focus on Autoregressive Transformers and their use as Large Language Models (LLMs), but will also touch on Diffusion Models as well as Reinforcement Learning. The goal of this course is to get students to be proficient enough with the inner workings of deep generative models-along with the theoretical and empirical support for their design-to be able to understand and reason about cutting-edge research. This is an advanced machine learning course, and assumes a familiarity with basic machine learning concepts (generalization, overfitting, etc.) and techniques (regularization, stochastic gradient descent, etc).
Instructor(s): A. Holtzman Terms Offered: Winter
Prerequisite(s): DATA 37711; Consent of Instructor unless graduate student in Data Science
Equivalent Course(s): DATA 37712

CMSC 37799. Topics in Machine Learning: Machine Learning and Inverse Problems. 100 Units.
In many scientific and medical settings, we cannot directly observe phenomena of interest, such as images of a person's internal organs, the microscopic structure of materials or cells, or observations of distant stars and galaxies. Rather, we use MRI scanners, microscopes, and satellites to collect indirect data that require sophisticated numerical methods to interpret. This course will explore a variety of machine learning techniques for solving inverse problems, ranging from linear inverse problems to PDE parameter estimation and data assimilation.
Instructor(s): R. Willett Terms Offered: Winter
Prerequisite(s): STAT 37710/CAAM 37710/CMSC 35400 or consent of instructor.
Equivalent Course(s): STAT 37799, CAAM 37799

CMSC 37810. Mathematical Computation I: Matrix Computation Course. 100 Units.
This is an introductory course on numerical linear algebra, which is quite different from linear algebra. We will be much less interested in algebraic results that follow from axiomatic definitions of fields and vector spaces but much more interested in analytic results that hold only over the real and complex fields. The main objects of interest are real- or complex-valued matrices, which may come from differential operators, integral transforms, bilinear and quadratic forms, boundary and coboundary maps, Markov chains, correlations, DNA microarray measurements, movie ratings by viewers, friendship relations in social networks, etc. Numerical linear algebra provides the mathematical and algorithmic tools for analyzing these matrices. Topics covered: basic matrix decompositions LU, QR, SVD; Gaussian elimination and LU/LDU decompositions; backward error analysis, Gram-Schmidt orthogonalization and QR/complete orthogonal decompositions; solving linear systems, least squares, and total least squares problem; low-rank matrix approximations and matrix completion. We shall also include a brief overview of stationary and Krylov subspace iterative methods; eigenvalue and singular value problems; and sparse linear algebra.
Terms Offered: Autumn
Prerequisite(s): Linear algebra (STAT 24300 or equivalent) and some previous experience with statistics.
Equivalent Course(s): STAT 30900, CAAM 30900

CMSC 37812. Mathematical Computation III: Numerical Methods for PDE's. 100 Units.
The first part of this course introduces basic properties of PDE's; finite difference discretizations; and stability, consistency, convergence, and Lax's equivalence theorem. We also cover examples of finite difference schemes; simple stability analysis; convergence analysis and order of accuracy; consistency analysis and errors (i.e.,

dissipative and dispersive errors); and unconditional stability and implicit schemes. The second part of this course includes solution of stiff systems in 1, 2, and 3D; direct vs. iterative methods (i.e., banded and sparse LU factorizations); and Jacobi, Gauss-Seidel, multigrid, conjugate gradient, and GMRES iterations.
Terms Offered: Spring
Prerequisite(s): Some prior exposure to differential equations and linear algebra
Equivalent Course(s): STAT 31100, MATH 38309, CAAM 31100

CMSC 38000-38100. Computability Theory I-II.
The courses in this sequence are offered in alternate years.

CMSC 38000. Computability Theory I. 100 Units.
We investigate the computability and relative computability of functions and sets. Topics include mathematical models for computations, basic results such as the recursion theorem, computably enumerable sets, and priority methods.
Instructor(s): D. Hirschfeldt Terms Offered: Spring
Prerequisite(s): Consent of department counselor. MATH 25500 or consent of instructor.
Equivalent Course(s): MATH 30200

CMSC 38100. Computability Theory II. 100 Units.
CMSC 38100 treats classification of sets by the degree of information they encode, algebraic structure and degrees of recursively enumerable sets, advanced priority methods, and generalized recursion theory.
Instructor(s): D. Hirschfeldt Terms Offered: Spring
Prerequisite(s): Consent of department counselor. MATH 25500 or consent of instructor.
Equivalent Course(s): MATH 30300

CMSC 38130. Complexity Theory. 100 Units.
Computability topics are discussed (e.g., the s-m-n theorem and the recursion theorem, resource-bounded computation). This course introduces complexity theory. Relationships between space and time, determinism and non-determinism, NP-completeness, and the P versus NP question are investigated.

CMSC 38300. Numerical Solutions to Partial Differential Equations. 100 Units.
This course covers the basic mathematical theory behind numerical solution of partial differential equations. We investigate the convergence properties of finite element, finite difference and other discretization methods for solving partial differential equations, introducing Sobolev spaces and polynomial approximation theory. We emphasize error estimators, adaptivity, and optimal-order solvers for linear systems arising from PDEs. Special topics include PDEs of fluid mechanics, max-norm error estimates, and Banach-space operator-interpolation techniques.
Instructor(s): L. R. Scott Terms Offered: This course is offered in alternate years.
Prerequisite(s): Consent of department counselor and instructor
Equivalent Course(s): MATH 38300

CMSC 38400. Cryptography. 100 Units.
Cryptography is the use of algorithms to protect information from adversaries. Though its origins are ancient, cryptography now underlies everyday technologies including the Internet, wifi, cell phones, payment systems, and more. This course is an introduction to the design and analysis of cryptography, including how "security" is defined, how practical cryptographic algorithms work, and how to exploit flaws in cryptography. The course will cover algorithms for symmetric-key and public-key encryption, authentication, digital signatures, hash functions, and other primitives.
Instructor(s): David Cash

CMSC 38401. Topics in Cryptography. 100 Units.
Graduate study of topics in cryptography.

CMSC 38405. Arithmetic Combinatorics. 100 Units.
This course covers a variety of topics in arithmetic combinatorics such as inverse problems, incidence geometry, uniformity, regularity and pseudo-randomness. A special attention will be paid to connections to classical mathematics and theoretical computer science.
Instructor(s): Alexander Razborov Terms Offered: Spring
Equivalent Course(s): MATH 38405

CMSC 38410. Quantum Computing. 100 Units.
This course covers mathematical and complexity aspects of quantum computing, putting aside all questions pertaining to its physical realizability. Possible topics include: (1) quantum model of computation, quantum complexity classes, and relations to their classical counterparts; (2) famous quantum algorithms (including Shor and Grover); (3) black-box quantum models (lower and upper bounds); (4) quantum communication complexity (lower and upper bounds); and (5) quantum information theory.
Instructor(s): A. Razborov Terms Offered: This course is offered in alternate years.
Prerequisite(s): Consent of department counselor. Basic knowledge of computational complexity and linear algebra required; knowledge of quantum mechanics not required
Note(s): Not offered in 2016-17.
Equivalent Course(s): MATH 38410

CMSC 38420. Mathematics of Quantum Computing. 100 Units.
This course is a gentle introduction to mathematical foundations of quantum computing taught in completely rigorous format: we will completely disregard physical aspects and specific questions pertaining to particular implementations. An (approximate) list of topics: reversible, probabilistic and quantum computation. Quantum complexity classes and relations to their classical counterparts. Fundamental quantum algorithms, notably Grover's search and Shor's factoring algorithm. Quantum (query) complexity theory and quantum communication complexity. Quantum probability, super-operators and non-unitary quantum computation. Basics of quantum information theory and quantum error-correction.
Equivalent Course(s): MATH 38420

CMSC 38500. Computability and Complexity Theory. 100 Units.
Part one of this course consists of models for defining computable functions: primitive recursive functions, (general) recursive functions, and Turing machines; the Church-Turing Thesis; unsolvable problems; diagonalization; and properties of computably enumerable sets. Part two of this course deals with Kolmogorov (resource bounded) complexity: the quantity of information in individual objects. Part three of this course covers functions computable with time and space bounds of the Turing machine: polynomial time computability, the classes P and NP, NP-complete problems, polynomial time hierarchy, and P-space complete problems.
Instructor(s): A. Razborov
Prerequisite(s): Consent of department counselor and instructor
Note(s): Not offered in 2016-17.
Equivalent Course(s): MATH 30500, TTIC 31060

CMSC 38502. Topics in Combinatorics and Logic. 100 Units.
We will discuss several ideas, methods and results in Combinatorics and those parts of Mathematical Logic that are close to Theoretical Computer Science. Complexity Theory itself is excluded this year since I will teach a more systematic course on the subject in the Spring Quarter.
Instructor(s): Alexander Razborov Terms Offered: Winter
Prerequisite(s): None
Equivalent Course(s): MATH 38502

CMSC 38700. Complexity Theory B. 100 Units.
This course covers topics in computational complexity theory, with an emphasis on combinatorial problems in complexity.
Instructor(s): A. Razborov
Prerequisite(s): Consent of department counselor and instructor
Equivalent Course(s): MATH 38703

CMSC 38800. Complexity Theory. 100 Units.
Complexity Theory is the branch of Theoretical Computer Science that studies inherent limitations on the efficiency of performing various computational tasks. In this course I hope to cover at least the most fundamental results from uniform (Turing) complexity, circuit complexity, communication complexity, algebraic complexity and proof complexity.
Instructor(s): Alexander Razborov
Prerequisite(s): None, but some familiarity with the book "Computational Complexity" by Arora and Barak might be helpful.
Equivalent Course(s): MATH 38800

CMSC 38815. Geometric Complexity. 100 Units.
This course provides a basic introduction to geometric complexity theory, an approach to the P vs. NP and related problems through algebraic geometry and representation theory. No background in algebraic geometry or representation theory will be assumed.
Instructor(s): K. Mulmuley Terms Offered: This course is offered in alternate years.
Prerequisite(s): Consent of department counselor and instructor
Note(s): Background in algebraic geometry or representation theory not required
Equivalent Course(s): MATH 38815

CMSC 39000. Computational Geometry. 100 Units.
This course is a seminar on topics in computational geometry.
Instructor(s): K. Mulmuley Terms Offered: This course is offered in alternate years.
Note(s): Not offered in 2016-17.

CMSC 39010. Geometric Methods in Computer Science. 100 Units.
The course covers fundamental concepts, algorithms and techniques in computational and metric geometry. Topics covered include: convex hulls, polygon triangulations, range searching, segment intersection, Voronoi diagrams, Delaunay triangulations, metric and normed spaces, low-distortion metric embeddings and their applications in approximation algorithms, padded decomposition of metric spaces, Johnson-Lindenstrauss transform and dimension reduction, approximate nearest neighbor search and locality-sensitive hashing. Expected outcomes: -- Know standard algorithms and data structures for solving geometric problems -- Be able to design efficient algorithms and data structures for solving geometric problems -- Understand basic concepts of metric geometry such as metric and normed space, low distortion embedding, dimension reduction, nearest

neighbor search. -- Understand applications of metric geometry to the field of approximation algorithms and other areas of computer science.
Instructor(s): Makarychev, Yury Terms Offered: Spring
Prerequisite(s): Undergraduate-level algorithms, linear algebra and probability classes; a good background in mathematical analysis/calculus.
Equivalent Course(s): TTIC 31100

CMSC 39020. Geometry, Complexity and Algorithms. 100 Units.
This course will try to explore these three topics and their interactions. Among the topics likely to be discussed are metric measure geometry (e.g. concentration of measure) and its use designing algorithms, machine learning, manifold learning, the complexity of the construction of isotopies and nullcobordisms, the Blum-Cucker-Smale theory of real computation and estimates for the complexity of root finding and related problems, persistence homology and applications, and other topics that seem like a good idea as the course develops.
Equivalent Course(s): MATH 38900

CMSC 39100. The Physics of Computation. 100 Units.
The class will focus on the theory of quantum information, covering specific topics at the intersection of theoretical computer science and physics and bringing students to the research frontier in these areas. Of particular interest are the theoretical foundations needed to characterize the computational power of quantum experiments in the "Near-term Intermediate Scale Quantum" era, Hamiltonian complexity and the theory of QMA completeness, and the power of resource limited models of quantum computation. While no formal prerequisites are required, a working familiarity with the basics of computational complexity theory and quantum mechanics is recommended.
Instructor(s): Fefferman

CMSC 39600. Topics in Theoretical Computer Science. 100 Units.
Graduate study in current topics in theoretical computer science.
Equivalent Course(s): MATH 39600

CMSC 39800. Rdg/Rsch: Computer Science. 300.00 Units.
Directed reading and research in computer science, under the guidance of a faculty member.

CMSC 70000. Advanced Study: Computer Science. 300.00 Units.
Advanced Study: Computer Science