Chair
• Todd Dupont
Professors
• Yali Amit
• Laszlo Babai
• Andrew Chien
• Frederic Chong
• Todd Dupont
• Ian Foster
• John Goldsmith
• Stuart A. Kurtz
• John Lafferty
• Ketan Mulmuley
• Michael J. O Donnell
• Alexander Razborov
• John Reppy
• L. Ridgway Scott
• Janos Simon
• Rick L. Stevens
Associate Professors
• Shan Lu
• Anne Rogers
Assistant Professors
• Ravi Chugh
• Andrew Drucker
• Aaron Elmore
• Ariel Feldman
• Haryadi Gunawi
• Henry Hoffmann
• Gordon Kindlmann
• Risi Kondor
Adjunct faculty
• Geraldine Brady (adjunct assistant professor)
• Todd Nugent (adjunct assistant professor)
• Mark Shacklette (adjunct professor)
Research

We construe the field of computer science broadly to include the complementary concepts of computation, information, and communication. We employ modes of inquiry and creation from pure mathematics to experiment and observation to design and engineering. We investigate computation, information, and communication as inherently interesting phenomena; we also investigate the many ways in which computational concepts engage other topics: computational tools for science and scholarship, computational infrastructure for society.

Our current research may be classified into computational linguistics, computer security, computer vision, discrete mathematics, machine learning, programming languages, scientific computing and visualization, systems, and theoretical computer science.

Artificial Intelligence

We use language, vision, and learning as the organizing themes driving work in artificial intelligence.

Computational Mathematics

Our faculty and students study the foundations of simulation technology. This includes the development and mathematical analysis of numerical algorithms for approximating partial differential equations. We also study language and systems aspects of numerical computing, as exemplified in the FEniCS Project. Parallel and high performance computing are an integral part of our efforts.

Systems

Our faculty advance principles and understanding of a broad range of areas, including systems and networking, programming languages and software engineering, software and hardware architecture, data-intensive computing and databases, graphics and visualization, computer security, and systems biology. Particular areas of focus include formal definition, design, and implementation of programming languages, data-intensive computing systems and algorithms, large scale distributed and collaborative systems, heterogeneous computer architectures, reliable computing systems, and self-tuning systems.

Theoretical Computer Science

We investigate the fundamental descriptive and algorithmic concepts underlying the computational process and the intrinsic limitations to efficient computation. Our faculty specialize in complexity theory, computational geometry, algorithms, discrete random processes, combinatorics, computability theory, and programming language semantics.
In addition to these more traditional areas, we have a growing commitment to research in applied computing. Examples include: developing mathematical and computational methods to measure and graphically depict structure in three-dimensional imaging modalities (like MRI and CT) and combining molecular dynamics simulations with chemical experimental data to gain an understanding of the motions and kinetics of biological molecules.

These efforts are enhanced by strong connections to the Computation Institute, which develops computational tools and techniques for a broad range of disciplines, including biological and physical sciences, medicine, law, the arts, and humanities; the James Franck Institute, which focuses on condensed matter physics; and the Institute for Biophysical Dynamics, which provides a forum for studying questions that arise at the boundary between the biological and physical sciences. In addition, we have collaborations with faculty in academic departments, including the geophysical sciences, linguistics, mathematics, physics, psychology, and statistics, as well as with the Division of Mathematics and Computer Science at Argonne National Laboratory (ANL), which is operated by the University of Chicago for the US Department of Energy.

**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, 1100 East 58th Street, Chicago, 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 (http://csphd.sites.uchicago.edu/page/admission-phd-program), by writing to Admissions, Department of Computer Science, University of Chicago, 1100 East 58th Street, Chicago, IL 60637, or by telephoning 773.702.6011.

General information about our department is available from the departmental website (http://www.cs.uchicago.edu).

**THE PHD PROGRAM**

The department offers two PhD tracks: a standard track and a computational mathematics track.
The detailed requirements for the PhD degree and for the MS degree within the PhD program can be found by visiting the Department’s web page (http://www.cs.uchicago.edu). Here is a brief summary:

To obtain an MS degree within the PhD program, students in the PhD program must fulfill the following requirements:

• Course requirements. Five core courses and four electives. The core courses include two in Theory, two in Systems, and one in Machine Learning. Please refer to the web page for details regarding the core courses.

A modified set of core courses applies to the computational mathematics track (see the website). The list of electives is frequently updated; we refer you to the web page.

Students must complete the course requirements by the end of their second year of study. To receive an MS degree within the PhD program, students must receive a grade of at least B in all the nine courses and have a GPA of at least 3.00 in the five core courses, and write a Master’s paper and pass a Master’s examination.

To obtain a PhD degree, students must meet enhanced MS requirements, including at least B on each of the nine courses and a GPA of at least 3.25 on the five core courses; plus the following:

• Pass a Candidacy Exam
• Write and defend a Doctoral Thesis that contains significant original research in computer science.

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 general University computing facilities and our Undergraduate Computing Laboratory (which contains about four dozen Macintosh computers and two dozen Linux workstations with extensive peripherals and software), the Ryerson Research Computing Service 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 services on a number of shared Unix/Linux computing servers and workstations interconnected by high speed ethernet; a workstation on each desktop (a total of more than 230 workstations); wireless connections; substantial amounts of personal file storage, backed up nightly for reliability and accessible transparently from all departmental computers; printer service; web servers and access to the Internet; Linux clusters for research in parallel computing and High Performance Computing. The department also has access to highly parallel machines at ANL, and at the campus research computing center.
COURSES

For the list of courses offered and the course descriptions, please consult the courses section of the departmental web page (http://www.cs.uchicago.edu/courses).

COMPUTER SCIENCE COURSES

CMSC 31100. Big Ideas in Computer Science. 100 Units.
This course introduces many of the important concepts in the broad area of computer science. Each week a different professor gives a three-lecture sequence on a big idea in their field of specialty. Previous ideas have included undecidability, randomness, cryptography, stability of numerical algorithms, structural operational semantics, software engineering, and the Internet.
Terms Offered: Autumn
Prerequisite(s): Consent of department counselor and instructor

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

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

CMSC 32620. Implementation of Computer Languages II. 100 Units.
This course is a continuation of CMSC 22610, covering compilers for general-purpose languages. Topics include compiler-immediate representations, continuation-passing style, runtime representations, code generation, code optimization, register allocation, instruction scheduling, and garbage collection. This is a project-based course in which students construct a complete, working compiler for a small language using Standard ML.
Instructor(s): J. Reppy Terms Offered: Not offered in 2015-16. Generally offered alternate years.
Prerequisite(s): CMSC 22610 required; CMSC 22100 strongly recommended
Note(s): Generally offered alternate years.
Equivalent Course(s): CMSC 22620

CMSC 33300. Networks and Distributed Systems. 100 Units.
This course focuses on the principles and techniques used in the development of networked and distributed software. Topics include programming with sockets; concurrent programming; data link layer (Ethernet, packet switching, etc.); internet and routing protocols (UDP, TCP); and other commonly used network protocols and techniques. This is a project-oriented course in which students are required to develop software in C on a UNIX environment.
Instructor(s): B. Sotomayor Terms Offered: Winter
Prerequisite(s): CMSC 15400
Equivalent Course(s): CMSC 23300
CMSC 33310. Advanced Distributed Systems. 100 Units.
This course explores advanced topics in distributed systems. Topics include supercomputing (architectures, applications, programming models, etc.); grid computing with an emphasis on Globus technologies; Infrastructure-as-a-Service clouds (virtual infrastructure management, Amazon EC2, etc.), Platform-as-a-Service clouds (Google App Engine, etc.), and the Software-as-a-Service model; and other current topics related to using and building distributed systems. The course includes a substantial practical component but also requires students to read papers and articles on current advances in the field.
Instructor(s): B. Sotomayor Terms Offered: Spring
Prerequisite(s): CMSC 23300 or consent of instructor
Equivalent Course(s): CMSC 23310

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: Winter
Prerequisite(s): CMSC 23000 or 23300 or equivalent are required.
Equivalent Course(s): CMSC 23400

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

CMSC 33600. Type Systems for Programming Languages. 100 Units.
This course covers the basic ideas of type systems, their formal properties, their role in programming language design, and their implementation. Exercises involving design and implementation explore the various options and issues.
Terms Offered: Winter
Prerequisite(s): Consent of department counselor
Note(s): CMSC 22100 recommended.
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 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 scientific computing on scientific data. Programming projects will be in C.
Instructor(s): G. Kindlmann Terms Offered: Winter
Prerequisite(s): Strong programming skills and basic knowledge of linear algebra and calculus
Note(s): This course is offered in alternate years.

CMSC 33950. Data Visualization: Aesthetics, Intent, and Practice. 100 Units.
This course investigates how data visualizations are made and used today. Addressing a lack of both critical attention and technical literacy in how our society engages with increasingly common and sophisticated data-driven representations, we will retrace some history of the form as well as investigate its production and consumption. From uses in the sciences to economics to the popular media, data visualization serves various purposes framed by divergent intentions. Through reading, discussion, and crucially, team-based production, we will examine these myriad forms. While the course will not dwell on the deep computational details of data processing and requires no special technical skills, we will introduce various methodologies for creation and distribution such as D3, Processing, and P5.js. Projects and critique resulting from these inquiries enable an understanding for how any data visualization is the result of countless subjective judgments, design decisions, and persuasive intentions.
Instructor(s): J. Salavon and G. Kindlmann Terms Offered: Winter
Equivalent Course(s): CDIN 40333, CMST 38007, ARTV 40333

CMSC 34000. Scientific Parallel Computing. 100 Units.
This course covers the use of multiple processors cooperating to solve a common task, as well as related issues in computer architecture, performance analysis, prediction and measurement, programming languages, and algorithms for large-scale computation. Programming at least one parallel computer is required. Possibilities include one of the clusters of workstations connected by high-speed networks on campus. We focus on state-of-the-art parallel algorithms for scientific computing. Topics are based on interest. General principles of parallel computing are emphasized.
Instructor(s): L. R. Scott Terms Offered: Autumn
Prerequisite(s): Consent of department counselor and instructor required; experience in scientific computing recommended
Note(s): This course is offered in alternate years.
CMSC 34200. Numerical Hydrodynamics. 100 Units.
This course covers numerical methods for the solution of fluid flow problems. We also make a theoretical evaluation of the methods and experimental study based on the opinionated book Fundamentals of Computational Fluid Dynamics by Patrick J. Roache.
Instructor(s): T. Dupont Terms Offered: Winter
Prerequisite(s): Consent of department counselor. Ability to program; and familiarity with elementary numerical methods and modeling physical systems by systems of differential equations

CMSC 34710. Wireless Sensor Networks. 100 Units.
This course introduces the concepts and technologies for building embedded systems and wireless sensors nets by focusing on four areas: low-power hardware, wireless networking, embedded operating systems, and sensors. Two assignments provide hands-on experience by deploying small wireless sensor motes running TinyOS to form an ad-hoc peer-to-peer network that can collect environmental data and forward it back to an 802.11b-equipped embedded Linux module. Students also read and summarize papers, participate in classroom discussions, and work on a team research project.
Instructor(s): R. Stevens
Prerequisite(s): Consent of department counselor. Graduate-level understanding of Unix/Linux operating systems, networking, computer architecture, and programming

CMSC 35000. Introduction to Artificial Intelligence. 100 Units.
This course introduces the theoretical, technical, and philosophical aspects of Artificial Intelligence. We emphasize computational and mathematical modes of inquiry into the structure and function of intelligent systems. Topics include learning and inference, speech and language, vision and robotics, and reasoning and search.

CMSC 35050. Computational Linguistics. 100 Units.
This is a course in the Computer Science department, intended for upper-level undergraduates, or graduate students, who have good programming skills. There will be weekly programming assignments in Python. We will look at several current topics in natural language processing, and discuss both the theoretical basis for the work and engaging in hands-on practical experiments with linguistic corpora. In line with most current work, our emphasis will be on systems that draw conclusions from training data rather than relying on the encoding of generalizations obtained by humans studying the data. As a consequence of that, in part, we will make an effort not to focus on English, but to look at a range of human languages in our treatments.
Instructor(s): J. Goldsmith Terms Offered: Spring
Prerequisite(s): CMSC 12200, 15200 or 16200, or by consent
Equivalent Course(s): LING 28600, LING 38600, CMSC 25020
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 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, elements of computational learning theory, the VC dimension, boosting, Bayesian learning, graphical models, clustering, dimensionality reduction, linear classifiers, kernel methods including SVMs, and an introduction to statistical learning theory.
Terms Offered: Spring
Prerequisite(s): Consent of instructor
Equivalent Course(s): STAT 37710

CMSC 35500. Computer Vision. 100 Units.
This course covers deformable models for detecting objects in images. Topics include one-dimensional models to identify object contours and boundaries; two-dimensional models for image matching; and sparse models for efficient detection of objects in complex scenes. Mathematical tools needed to define the models and associated algorithms are developed. Applications include detecting contours in medical images, matching brains, and detecting faces in images. Neural network implementations of some of the algorithms are presented, and connections to the functions of the biological visual system are discussed.
Instructor(s): Y. Amit Terms Offered: Spring
Equivalent Course(s): CMSC 25050, STAT 37900

CMSC 35900. Topics in Artificial Intelligence. 100 Units.
This course covers topics in artificial intelligence.
Terms Offered: Autumn, Winter, Spring
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.
Instructor(s): L. Babai Terms Offered: Spring
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.
The focus of this course is the analysis and design of efficient algorithms, with emphasis on ideas rather than on implementation. Algorithmic questions include sorting and searching, discrete optimization, algorithmic graph theory, algorithmic number theory, and cryptography. Design techniques include “divide-and-conquer” methods, dynamic programming, greedy algorithms, and graph search, as well as the design of efficient data structures. Methods of algorithm analysis include asymptotic notation, evaluation of recurrent inequalities, the concepts of polynomial-time algorithms, and NP-completeness.
Instructor(s): L. Babai Terms Offered: Winter
Prerequisite(s): Consent of instructor.

CMSC 37100. Topics in Algorithms. 100 Units.
This course covers current topics in algorithms.
Terms Offered: Autumn, Winter, Spring
Prerequisite(s): Consent of department counselor. CMSC 27200 or consent of instructor.

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 Terms Offered: Autumn
Prerequisite(s): Consent of department counselor and instructor

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 Terms Offered: Winter
Prerequisite(s): Consent of department counselor. Linear algebra, basic combinatorics, or consent of instructor.
CMSC 37400. Constructive Combinatorics. 100 Units.
This course covers constructive combinatorial techniques in areas such as
e enumerative combinatorics, invariant theory, and representation theory of
symmetric groups. Constructive techniques refer to techniques that have algorithmic
flavor, such as those that are against purely existential techniques based on
counting.
Instructor(s): K. Mulmuley Terms Offered: Spring
Prerequisite(s): Consent of department counselor. Advanced knowledge of
mathematics and consent of instructor.

CMSC 37701. Topics in Bioinformatics. 100 Units.
This course covers current topics in bioinformatics.
Terms Offered: Autumn, Winter, Spring
Prerequisite(s): Consent of Consent of department counselor and instructor

CMSC 37720. Computational Systems Biology. 100 Units.
This course introduces concepts of systems biology. We also discuss computational
methods for analysis, reconstruction, visualization, modeling, and simulation of
complex cellular networks (e.g., biochemical pathways for metabolism, regulation,
and signaling). Students explore systems of their own choosing and participate
in developing algorithms and tools for comparative genomic analysis, metabolic
pathway construction, stoichiometric analysis, flux analysis, metabolic modeling,
and cell simulation. We also focus on understanding the computer science
challenges in the engineering of prokaryotic organisms.
Instructor(s): R. Stevens Terms Offered: Autumn
Prerequisite(s): Consent of department counselor and instructor

CMSC 37800. Numerical Computation. 100 Units.
This course covers topics in numerical methods and computation that are useful
in statistical research (e.g., simulation, random number generation, Monte Carlo
methods, quadrature, optimization, matrix methods).
Terms Offered: Autumn. Not offered 2011-12.
Prerequisite(s): Consent of departmental counselor. STAT 34300 or consent of
instructor.
Equivalent Course(s): STAT 30700
Department of Computer Science

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

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

CMSC 38000. Computability Theory I. 100 Units.
CMSC 38000 is concerned with recursive (computable) functions and sets generated by an algorithm (recursively enumerable sets). Topics include various mathematical models for computations (e.g., Turing machines and Kleene schemata, enumeration and s-m-n theorems, the recursion theorem, classification of unsolvable problems, priority methods for the construction of recursively enumerable sets and degrees).
Instructor(s): R. Soare Terms Offered: Winter
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): R. Soare Terms Offered: Winter, Spring
Prerequisite(s): Consent of department counselor. MATH 25500 or consent of instructor.
Equivalent Course(s): MATH 30300
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): R. Soare Terms Offered: Winter, Spring
Prerequisite(s): Consent of department counselor. MATH 25500 or consent of instructor.
Equivalent Course(s): MATH 30300

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: Spring. This course is offered in alternate years.
Prerequisite(s): Consent of department counselor and instructor
Equivalent Course(s): MATH 38300

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: Winter. 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.

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 Terms Offered: Winter
Prerequisite(s): Consent of department counselor and instructor
Equivalent Course(s): MATH 30500
CMSC 38512. Kolmogorov Complexity. 100 Units.
This course introduces the theory of Kolmogorov Complexity with an emphasis on its use in theoretical computer science, mostly in computational complexity. If time permits, we may briefly touch on its uses in statics, prediction, and learning.
Prerequisite(s): Consent of department counselor and instructor

CMSC 38600. Complexity Theory A. 100 Units.
This course covers topics in computational complexity theory, with an emphasis on machine-based complexity classes.
Terms Offered: Spring
Prerequisite(s): Consent of department counselor and instructor

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

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: Autumn. 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: Spring. This course is offered in alternate years.

CMSC 39600. Topics in Theoretical Computer Science. 100 Units.
This course is a seminar on current research in theoretical computer science.
Terms Offered: Autumn, Winter, Spring
Prerequisite(s): Consent of department counselor and instructor