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 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.
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. Accordingly, the descriptions below, a snapshot of our current active research, is 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 and visualization, high performance computing, human-computer interaction, computer science education, and interdisciplinary research in computing in the physical, biological, and social sciences.

### Artificial intelligence
Research spans the spectrum from foundational work in statistical machine learning to computer vision and computational linguistics. The AI group has strong ties to CAMI, the University’s Computational and Applied Mathematics Initiative.

### 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, human-computer interactions, 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, self-tuning systems, and emerging technologies.

### Theoretical computer science
We investigate the fundamental concepts underlying computation using and developing mathematical techniques, as well as topics in discrete mathematics. Our faculty specialize in complexity theory, algorithms, discrete mathematics, and combinatorics.

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, social sciences, and humanities; the James Frank Institute, which focuses on condensed matter physics; the Institute for Biophysical Dynamics, which provides a forum for studying questions that arise at the boundary between the biological and physical sciences; and the Institute for Molecular Engineering. In addition, we have collaborations with faculty in academic departments, including the geophysical sciences, linguistics, mathematics, physics, psychology, and statistics, and 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. We also have almost seamless collaborations with the Toyota Technological Institute on campus, especially in the areas of Theoretical Computer Science and Machine Learning.

### Academic 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: they must do especially well in the five core courses with a 3.25 average. Details about the requirements can be found in the departmental web page. 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 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), 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 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 and group use which 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.

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 31150. Mathematical Toolkit. 100 Units.**
Introduction to mathematical techniques of linear algebra and probability used in different areas of computer science. Topics in include Linear Algebra (Hilbert spaces, eigenvalues and eigenvectors, SVD, least squares), discrete probability, Gaussian variables, concentration inequalities and dimension reduction, Linear Programming and LP duality. Time permitting, martingales, stochastic processes.
Instructor(s): Tulsiani Terms Offered: Autumn

**CMSC 31230. Fundamentals of Deep Learning. 100 Units.**
Introduction to fundamental principles of deep learning. Although deep learning systems are evolving rapidly, this course attempts to teach material that will remain relevant and useful as the field changes. The course will emphasize theoretical and intuitive understanding to the extent possible.

- Expected outcomes:
  - Ability to design and train novel deep learning architectures.
  - An understanding of the general issues and phenomenon sufficient to guide architecture design and training.

Instructor(s): David McAllester Terms Offered: Winter
Prerequisite(s): Introduction to machine learning course.
Equivalent Course(s): TTIC 31230
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 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 Terms Offered: Autumn

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 33001. Topics in Systems. 100 Units.
This course covers a selection of advanced topics in computer systems.
Terms Offered: Autumn, Spring, Winter
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 Terms Offered: Autumn
Prerequisite(s): Consent of department counselor and instructor

CMSC 33210. Usable Security and Privacy. 100 Units.
Questions of usability and privacy in computer systems, including human factors.
Instructor(s): Ur Terms Offered: Spring
Prerequisite(s): Consent of department counselor and instructor

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.
Instructor(s): A. Feldman Terms Offered: Autumn
Prerequisite(s): Consent of department counselor and instructor

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 (IP, IPv6, ARP, etc.); end-to-end 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.
In recent years, large distributed systems have taken a prominent role not just in scientific inquiry, but also in our daily lives. When we perform a search on Google, stream content from Netflix, place an order on Amazon, or catch up on the latest comings-and-goings on Facebook, our seemingly minute requests are processed by complex systems that sometimes include hundreds of thousands of computers, connected by both local and wide area networks. Recent papers in the field of Distributed Systems have described several solutions (such as MapReduce, BigTable, Dynamo, Cassandra, etc.) for managing large-scale data and computation. However, building and using these systems pose a number of more fundamental challenges: How do we keep the system operating correctly even when individual machines fail? How do we ensure that all the machines have a consistent view of the system’s state? (And how do we ensure this in the presence of failures?) How can we determine the order of events in a system where we can’t assume a single global clock? Many of these fundamental problems were identified and solved over the course of several decades, starting in the 1970s. To better appreciate the challenges of recent developments in the field of Distributed Systems, this course will guide students through seminal work in Distributed Systems from the 1970s, '80s, and '90s, leading up to a discussion of recent work in the field.
Instructor(s): B. Sotomayor Terms Offered: Not offered 2017-2018.
Prerequisite(s): CMSC 23300 with at least a B+, or by consent.
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: Not offered 2017-2018.
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 33520. Data Intensive Computer Systems. 100 Units.
Big Data and Data Analytics have become hot topics as well as drivers of multi-billion dollar industries. With unprecedented data collection from e-commerce, the WWW, scientific instruments, mobile phones, and IoT. The course objective is to expose students to the technical challenges of data-intensive computing systems, including canonical driving problems, research systems, and emerging technologies. While other classes focus on analysis algorithms (or even underlying statistical or machine learning methods), we focus on the computer systems and technology needed to achieve scalable and efficient data-intensive computing systems. Through paper reading, discussions, presentation, and in-depth projects, students will develop a broad familiarity with current challenges and hands-on experience with a range of systems which together provide a solid preparation for research in the area. Course topics include: parallel filesystems, SQL databases, NoSQL/Mapreduce systems, storage class memories (from Flash to Memristor to ReRAM), and popular open source infrastructures such as Spark, Succinct, Hadoop, VoltDBHadoopDB, Cassandra, Memcached, MongoDB, and others.
Instructor(s): Chien Terms Offered: Spring
Prerequisite(s): Consent of department counselor and instructor

CMSC 33550. Introduction to Databases. 100 Units.
This course is an introduction to database design and programming using the relational model. Topics include DBMS architecture, entity-relationship and relational models, relational algebra, relational calculus, functional dependencies and normal forms, web DBs and PHP, query optimization, and physical data organization. The lab section will guide students through the collaborative 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 collaborative software development.
Instructor(s): Elmore Terms Offered: Winter
Prerequisite(s): Consent of department counselor and instructor

CMSC 33600. Type Systems for Programming Languages. 100 Units.
This course covers the basics 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 33700. 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. These include coordinate systems and transformations, the graphics pipeline, basic geometric algorithms, texture mapping, level-of-detail optimizations, and shadows. Students are required to complete both written assignments and programming projects using OpenGL.
Instructor(s): J. Reppy Terms Offered: TBD
Prerequisite(s): Consent of department counselor and instructor

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 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. Experience in scientific computing recommended
Note(s): Not offered in 2016-17, 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 sensor 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 34900. Topics in Scientific Computing. 100 Units.
This course covers a selection of advanced topics in Scientific Computing.
Instructor(s): Scott Terms Offered: Autumn
Prerequisite(s): Consent of department counselor and instructor

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 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, Bayesian learning, graphical models, clustering, dimensionality reduction, kernel methods including SVMs, matrix completion, neural networks, and an introduction to statistical learning theory.
Instructor(s): I. Kondor Terms Offered: Spring
Prerequisite(s): Consent of instructor
Equivalent Course(s): STAT 37710
CMSC 35470. 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.
Prerequisite(s): Consent of department counselor and instructor

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-Holden 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 advanced methods for algorithm design and analysis. Topics covered include asymptotic analysis, greedy algorithms, dynamic programming, amortized analysis, randomized algorithms and probabilistic methods, combinatorial optimization and approximation algorithms, linear programming, and advanced data structures.

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.

- Understand linear programming duality and applications to problems such as max-flow/min-cut. Be able to write duals for linear programs.

Instructor(s): Yury Makarychev Terms Offered: Winter
Prerequisite(s): Assumes familiarity with proofs and the asymptotic notation. Some basic knowledge of the notion of NP-hardness is also required.
Equivalent Course(s): TTIC 31010

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.
Note(s): Not offered in 2016-17.
CMSC 37220. 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.

Instructor(s): Ohannessian, Mesrob
Terms Offered: Autumn
Prerequisite(s): None.
Equivalent Course(s): TTIC 31150

CMSC 37400. Constructive Combinatorics. 100 Units.
This course covers constructive combinatorial techniques in areas such as 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.
Note(s): Not offered in 2016-17.

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: Winter
Prerequisite(s): Algorithms (TTIC 31010 or CMSC 37000)
Equivalent Course(s): TTIC 31080
CMSC 37701. Topics in Bioinformatics. 100 Units.
This course covers current topics in bioinformatics.
Terms Offered: Autumn, Winter, Spring
Prerequisite(s): 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
Note(s): Not offered in 2016-17.

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

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.
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 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
Note(s): Not offered in 2016-17.

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
Note(s): Not offered in 2016-17.
Equivalent Course(s): MATH 30500

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.
Instructor(s): A. Razborov Terms Offered: Spring
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.
Note(s): Not offered in 2016-17.

CMSC 39010. Computational and Metric Geometry. 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.
Instructor(s): Makarychev, Yury Terms Offered: Winter

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.
Instructor(s): Shmuel Weinberger Terms Offered: Winter
Prerequisite(s): Undergraduate mathematics, the idea of a Turing machine and what an algorithm is, ideally a quarter of each of algebra, algebraic topology, differential topology and complex variables (even at the undergraduate level) and the willingness to work.
Equivalent Course(s): MATH 38900
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
Font Notice

This document should contain certain fonts with restrictive licenses. For this draft, substitutions were made using less legally restrictive fonts. Specifically:

- Times was used instead of Trajan.
- Times was used instead of Palatino.

The editor may contact Leepfrog for a draft with the correct fonts in place.