Computer Science & Engineering

During the past two decades, society has experienced unprecedented growth in digital technology. This revolution continues to redefine our way of life, our culture and our economy. Computer science and engineering education plays an irreplaceable role in this trend by preparing future technology leaders and innovators. It opens our minds to new horizons, unlocks doors to a broad range of career paths, accelerates professional advancement, and exposes us to ideas that are advancing the frontiers of science and technology beyond the field of computing. Alumni and students continually remind us that pursuing a degree in the Department of Computer Science & Engineering is an experience rarely matched elsewhere.

Master's Programs

The Department of Computer Science & Engineering offers four master's degrees: **Master of Science in Computer Science, Master of Science in Computer Engineering, Master of Science in Cybersecurity Engineering,** and **Master of Engineering in Computer Science and Engineering.** We accept both full-time and part-time students, offering class schedules that are flexible enough for part-time students but that provide enough classes for students to attend full-time. Obtaining a master's degree from the Department of Computer Science & Engineering can be done as a pure course option (MS in Computer Science and MS in Computer Engineering degrees only), or it can incorporate a specialized research experience. Master's research is a great way for our students to easily transition into future doctoral studies. Graduates of our program are also prepared to enter the industry, with many accepting positions at companies like Boeing, Google and Microsoft. Applicants to our master's programs are expected to have completed an undergraduate degree. A major or minor in computer science or computer engineering is helpful, but it is not required. Background requirements are listed within each degree program, along with options for meeting them.

PhD Programs

The Department of Computer Science & Engineering offers **PhD programs in Computer Science and in Computer Engineering.** Computer science research encompasses the fundamentals of software and algorithm design, machine learning and bioinformatics, visual and cyber-physical computing, and human-computer interaction. Computer engineering focuses on the interaction of software and hardware in the design of computing systems and networks. Our research groups have extensive interdisciplinary ties across the university, with collaborations in medicine, science, the humanities and social work. Recent graduates have accepted research and teaching faculty positions as well as research and engineering positions in leading technology companies.

Both PhD programs require a combination of courses, research and teaching. The required courses are often completed early in the program, since students are integrated into research groups during their first year and the program's emphasis is on creative research. The program has milestones that involve both written and oral components, and these provide structure for the five- to six-year degree. The program considers applicants with either bachelor's or master's degrees and has had successful applicants in the past whose backgrounds were outside of the field of computer science.

Phone:	314-935-6132
Email:	admissions@cse.wustl.edu
Website:	https://cse.wustl.edu/academics/graduate/index.html

Faculty

Chair

Roch Guérin (https://engineering.wustl.edu/faculty/Roch-Guerin.html)
Harold B. and Adelaide G. Welge Professor of Computer Science
PhD, California Institute of Technology
Computer networks and communication systems

Professors

Sanjoy Baruah (https://engineering.wustl.edu/faculty/Sanjoy-Baruah.html)
PhD, University of Texas at Austin
Real-time and safety-critical system design, cyber-physical systems, scheduling theory, resource allocation and sharing in distributed computing environments

Aaron Bobick (https://engineering.wustl.edu/faculty/Aaron-Bobick.html)
James M. McKelvey Professor and Dean
PhD, Massachusetts Institute of Technology
Computer vision, graphics, human-robot collaboration

Michael R. Brent (https://engineering.wustl.edu/faculty/Michael-Brent.html)
Henry Edwin Sever Professor of Engineering
PhD, Massachusetts Institute of Technology
Systems biology, computational and experimental genomics, mathematical modeling, algorithms for computational biology, bioinformatics

Jeremy Buhler (https://engineering.wustl.edu/faculty/Jeremy-Buhler.html)
PhD, Washington University
Computational biology, genomics, algorithms for comparing and annotating large biosequences

Roger D. Chamberlain (https://engineering.wustl.edu/faculty/Roger-Chamberlain.html)
DSc, Washington University
Computer engineering, parallel computation, computer architecture, multiprocessor systems

Yixin Chen (https://engineering.wustl.edu/faculty/Yixin-Chen.html)
PhD, University of Illinois at Urbana-Champaign
Mathematical optimization, artificial intelligence, planning and scheduling, data mining, learning data warehousing, operations research, data security

Patrick Crowley (https://engineering.wustl.edu/faculty/Patrick-Crowley.html)
PhD, University of Washington
Computer and network systems, network security

Ron K. Cytron (https://engineering.wustl.edu/faculty/Ron-Cytron.html)
PhD, University of Illinois at Urbana-Champaign
Programming languages, middleware, real-time systems

Christopher D. Gill (https://engineering.wustl.edu/faculty/Christopher-Gill.html)
DSc, Washington University
Parallel and distributed real-time embedded systems, cyber-physical systems, concurrency platforms and middleware, formal models and analysis of concurrency and timing

Raj Jain (https://engineering.wustl.edu/faculty/Raj-Jain.html)
Barbara J. & Jerome R. Cox Jr. Professor of Computer Science
PhD, Harvard University
Network security, blockchains, medical systems security, industrial systems security, wireless networks, unmanned aircraft systems, internet of things, telecommunications networks, traffic management

Tao Ju (https://engineering.wustl.edu/faculty/Tao-Ju.html)
PhD, Rice University
Computer graphics, visualization, mesh processing, medical imaging and modeling

Chenyang Lu (https://engineering.wustl.edu/faculty/Chenyang-Lu.html)
Fullgraf Professor in the Department of Computer Science & Engineering
PhD, University of Virginia
Internet of things, real-time, embedded, and cyber-physical systems, cloud and edge computing, wireless sensor networks

Neal Patwari (https://engineering.wustl.edu/faculty/Neal-Patwari.html)
PhD, University of Michigan
Application of statistical signal processing to wireless networks, and radio frequency signals

Weixiong Zhang (https://engineering.wustl.edu/faculty/Weixiong-Zhang.html)
PhD, University of California, Los Angeles
Computational biology, genomics, machine learning and data mining, and combinatorial optimization

Associate Professors

Kunal Agrawal (https://engineering.wustl.edu/faculty/Kunal-Agrawal.html)
PhD, Massachusetts Institute of Technology
Parallel computing, cyber-physical systems and sensing, theoretical computer science

Roman Garnett (https://engineering.wustl.edu/faculty/Roman-Garnett.html)
PhD, University of Oxford
Active learning (especially with atypical objectives), Bayesian optimization, and Bayesian nonparametric analysis

Brendan Juba (https://engineering.wustl.edu/faculty/Brendan-Juba.html)
PhD, Massachusetts Institute of Technology
Theoretical approaches to artificial intelligence founded on computational complexity theory and theoretical computer science more broadly construed

Caitlin Kelleher (https://engineering.wustl.edu/faculty/Caitlin-Kelleher.html)
Hugo F. & Ina Champ Urbauer Career Development Associate Professor

PhD, Carnegie Mellon University
Human-computer interaction, programming environments, and learning environments

I-Ting Angelina Lee (https://engineering.wustl.edu/faculty/I-Ting-Angelina-Lee.html)
PhD, Massachusetts Institute of Technology
Designing linguistics for parallel programming, developing runtime system support for multi-threaded software, and building novel mechanisms in operating systems and hardware to efficiently support parallel abstractions

William D. Richard (https://engineering.wustl.edu/faculty/William-Richard.html)
PhD, University of Missouri-Rolla
Ultrasonic imaging, medical instrumentation, computer engineering

Yevgeniy Vorobeychik (https://engineering.wustl.edu/faculty/Yevgeniy-Vorobeychik.html)
PhD, University of Michigan
Artificial intelligence, machine learning, computational economics, security and privacy, multi-agent systems

William Yeoh (https://engineering.wustl.edu/faculty/William-Yeoh.html)
PhD, University of Southern California
Artificial intelligence, multi-agent systems, distributed constraint optimization, planning and scheduling

Assistant Professors

Ayan Chakrabarti
PhD, Harvard University
Computer vision computational photography, machine learning

Chien-Ju Ho (https://engineering.wustl.edu/faculty/Chien-Ju-Ho.html)
PhD, University of California, Los Angeles
Design and analysis of human-in-the-loop systems, with techniques from machine learning, algorithmic economics, and online behavioral social science

Ulugbek Kamilov (https://engineering.wustl.edu/faculty/Ulugbek-Kamilov.html)
PhD, École Polytechnique Fédérale de Lausanne, Switzerland
Computational imaging, image and signal processing, machine learning and optimization

Alvitta Ottley (https://engineering.wustl.edu/faculty/Alvitta-Ottley.html)
PhD, Tufts University
Designing personalized and adaptive visualization systems, including information visualization, human-computer interaction, visual analytics, individual differences, personality, user modeling and adaptive interfaces

Netanel Raviv (https://engineering.wustl.edu/faculty/Netanel-Raviv.html)
PhD, Technion, Haifa, Israel
Mathematical tools for computation, privacy and machine learning

Ning Zhang (https://engineering.wustl.edu/faculty/Ning-Zhang.html)
PhD, Virginia Polytechnic Institute and State University
System security, software security

Teaching Professors

Bill Siever (https://engineering.wustl.edu/faculty/Bill-Siever.html)
PhD, Missouri University of Science and Technology
Computer architecture, organization, and embedded systems

Todd Sproull (https://engineering.wustl.edu/faculty/Todd-Sproull.html)
PhD, Washington University
Computer networking and mobile application development

Professor of the Practice

Dennis Cosgrove (https://engineering.wustl.edu/faculty/Dennis-Cosgrove.html)
BS, University of Virginia
Programming environments and parallel programming

Senior Lecturers

Steve Cole (https://engineering.wustl.edu/faculty/Steve-Cole.html)
PhD, Washington University in St. Louis
Parallel computing, accelerating streaming applications on GPUs

Marion Neumann (https://engineering.wustl.edu/faculty/Marion-Neumann.html)
PhD, University of Bonn, Germany
Machine learning with graphs; solving problems in agriculture and robotics

Jonathan Shidal (https://engineering.wustl.edu/faculty/Jonathan-Shidal.html)
PhD, Washington University
Computer architecture and memory management

Douglas Shook (https://engineering.wustl.edu/faculty/Doug-Shook.html)
MS, Washington University
Imaging sensor design, compiler design and optimization

Lecturers

Hila Ben Abraham
PhD, Washington University in St. Louis
Parallel computing, accelerating streaming applications on GPUs, computer and network security, and malware analysis

Brian Garnett (https://engineering.wustl.edu/faculty/Brian-Garnett.html)
PhD, Rutgers University
Discrete mathematics and probability, generally motivated by theoretical computer science

James Orr (https://engineering.wustl.edu/faculty/James-Orr.html)
PhD, Washington University
Real-time systems theory and implementation, cyber-physical systems, and operating systems

Senior Professor

Jonathan S. Turner (https://engineering.wustl.edu/faculty/Jon-Turner.html)
PhD, Northwestern University
Design and analysis of internet routers and switching systems, networking and communications, algorithms

Senior Faculty Emeritus

Jerome R. Cox Jr.
ScD, Massachusetts Institute of Technology
Computer system design, computer networking, biomedical computing

Professors Emeriti

Takayuki D. Kimura
PhD, University of Pennsylvania
Communication and computation, visual programming

Seymour V. Pollack
MS, Brooklyn Polytechnic Institute
Intellectual property, information systems

Degree Requirements

Please visit to the following pages for information about computer science and engineering graduate programs:

- PhD in Computer Science or Computer Engineering (http://bulletin.wustl.edu/grad/engineering/computerscience/phd/)
- Master of Science (MS) in Computer Science (http://bulletin.wustl.edu/grad/engineering/computerscience/ms-computer-science/)
- Master of Science (MS) in Computer Engineering (http://bulletin.wustl.edu/grad/engineering/computerscience/ms-computer-engineering/)
- Master of Science (MS) in Cybersecurity Engineering (http://bulletin.wustl.edu/grad/engineering/computerscience/ms-cybersecurity/)
- Master of Engineering (MEng) in Computer Science and Engineering (http://bulletin.wustl.edu/grad/engineering/computerscience/meng-computer-science/)
- Certificate in Data Mining and Machine Learning (http://bulletin.wustl.edu/grad/engineering/computerscience/certificate-data-mining-machine-learning/)
- Certificate in Cybersecurity Engineering (http://bulletin.wustl.edu/grad/engineering/computerscience/certificate-cybersecurity-engineering/)

Courses

Visit online course listings to view semester offerings for E81 CSE (https://courses.wustl.edu/CourseInfo.aspx?sch=E&dept=E81&crslvl=5:8).

E81 CSE 500 Independent Study
Proposal form can be located at https://cse.wustl.edu/undergraduate/PublishingImages/Pages/undergraduate-research/Independent%20Study%20Form%20400.pdf
Credit variable, maximum 3 units.

E81 CSE 501N Introduction to Computer Science
An introduction to software concepts and implementation, emphasizing problem solving through abstraction and decomposition. Introduces processes and algorithms, procedural abstraction, data abstraction, encapsulation, and object-oriented programming. Recursion, iteration, and simple data structures are covered. Concepts and skills are mastered through programming projects, many of which employ graphics to enhance conceptual understanding. Java, an object-oriented programming language, is the vehicle of exploration. Active-learning sessions are conducted in a studio setting in which students interact with each other and the professor to solve problems collaboratively. Prerequisites: Comfort with algebra and geometry at the high school level is assumed. Patience, good planning, and organization will promote success. This course assumes no prior experience with programming.
Same as E81 CSE 131
Credit 3 units. BU: SCI EN: TU

E81 CSE 502N Data Structures and Algorithms
Study of fundamental algorithms, data structures, and their effective use in a variety of applications. Emphasizes importance of data structure choice and implementation for obtaining the most efficient algorithm for solving a given problem. A key component of this course is worst-case asymptotic analysis, which provides a quick and simple method for determining the scalability and effectiveness of an algorithm. Online textbook purchase required. Prerequisite: CSE 131/501N, and fluency with summations, derivatives, and proofs by induction.
Same as E81 CSE 247
Credit 3 units. EN: TU

E81 CSE 503S Rapid Prototype Development and Creative Programming
This course uses web development as a vehicle for developing skills in rapid prototyping. Students acquire the skills to build a Linux web server in Apache, to write a website from scratch in PHP, to run an SQL database, to perform scripting in Python, to employ various web frameworks, and to develop modern web applications in client-side and server-side JavaScript. The course culminates with a creative project in which students are able to synthesize the course material into a project of their own interest. The course implements an interactive studio format: after the formal presentation of a topic, students develop a related project under the supervision of the instructor. Prerequisite: CSE 131.
Same as E81 CSE 330S
Credit 3 units. EN: BME T, TU

E81 CSE 504N Object-Oriented Software Development Laboratory
Intensive focus on practical aspects of designing, implementing and debugging software, using object-oriented, procedural, and generic programming techniques. The course emphasizes familiarity and proficiency with a wide range of C++ language features through hands-on practice completing studio exercises and lab assignments, supplemented with readings and summary presentations for each session. Prerequisites: CSE 131 and CSE 247
Same as E81 CSE 332S
Credit 3 units. EN: BME T, TU

E81 CSE 505N Introduction to Digital Logic and Computer Design
Introduction to design methods for digital logic and fundamentals of computer architecture. Boolean algebra and logic minimization techniques; sources of delay in combinational circuits and effect on circuit performance; survey of common combinational circuit components; sequential circuit design and analysis; timing analysis of sequential circuits; use of computer-aided design tools for digital logic design (schematic capture, hardware description languages, simulation); design of simple processors and memory subsystems; program execution in simple processors; basic techniques for enhancing processor performance; configurable logic devices. Prerequisite: CSE 131.
Same as E81 CSE 260M
Credit 3 units.

E81 CSE 513T Theory of Artificial Intelligence and Machine Learning
Mathematical foundations for Artificial Intelligence and Machine Learning. An introduction to the PAC-Semantics ("Probably Approximately Correct") as a common semantics for knowledge obtained from learning and declarative sources, and the computational problems underlying the acquisition and processing of such

knowledge. We emphasize the design and analysis of efficient algorithms for these problems, and examine for which representations these problems are known or believed to be tractable. Prerequisite: CSE 347.
Credit 3 units. EN: TU

E81 CSE 514A Data Mining
With the vast advancements in science and technology, the acquisition of large quantities of data is routinely performed in many fields. Examples of large data include various types of data on the internet, high-throughput sequencing data in biology and medicine, extraterrestrial data from telescopes in astronomy, and images from surveillance cameras in security settings. Analyzing a large amount of data through data mining has become an effective means of extracting knowledge from data. This course introduces the basic concepts and methods of data mining and provides hands-on experience for processing, analyzing and modeling structured and unstructured data. Homework problems, exams, and programming assignments will be administrated throughout the course to enhance students' learning. Prerequisites: CSE 247, ESE 326 (or Math 3200 or Engr 328), and Math 233.
Credit 3 units. EN: BME T, TU

E81 CSE 515T Bayesian Methods in Machine Learning
This course will cover machine learning from a Bayesian probabilistic perspective. Bayesian probability allows us to model and reason about all types of uncertainty. The result is a powerful, consistent framework for approaching many problems that arise in machine learning, including parameter estimation, model comparison, and decision making. We will begin with a high-level introduction to Bayesian inference and then proceed to cover more advanced topics. These will include inference techniques (e.g., exact, MAP, sampling methods, the Laplace approximation), Bayesian decision theory, Bayesian model comparison, Bayesian nonparametrics, and Bayesian optimization. Prerequisites: CSE 417T and ESE 326.
Credit 3 units. EN: BME T, TU

E81 CSE 516A Multi-Agent Systems
This course introduces the fundamental techniques and concepts needed to study multi-agent systems, in which multiple autonomous entities with different information sets and goals interact. We will study algorithmic, mathematical, and game-theoretic foundations, and how these foundations can help us understand and design systems ranging from robot teams to online markets to social computing platforms. Topics covered may include game theory, distributed optimization, multi-agent learning and decision-making, preference elicitation and aggregation, mechanism design, and incentives in social computing systems. Prerequisites: CSE 347 (may be taken concurrently), ESE 326 (or Math 3200), and Math 233 or equivalents. Some prior exposure to artificial intelligence, machine learning, game theory, and microeconomics may be helpful, but is not required.
Credit 3 units. EN: BME T, TU

E81 CSE 517A Machine Learning
This course is the second course of a two-course sequence on machine learning (CSE 417T and CSE 517A). It assumes a fundamental understanding of machine learning foundations (both theoretical and practical) and introduces probabilistic machine learning approaches in depth as well as advanced topics at the frontier of the field. Topics to be covered include discriminative and generative probabilistic models, kernel methods (e.g., support vector machines, Gaussian processes), neural networks (deep learning), unsupervised learning techniques, and practical machine learning (e.g., feature engineering, dimensionality reduction, model comparison). Prerequisites: Math 233, CSE 247, ESE 326 or Math 3211, Math 309, and CSE 417T or ESE 417.
Credit 3 units. EN: TU

E81 CSE 518A Human-in-the-Loop Computation
This course is an exploration of the opportunities and challenges of human-in-the-loop computation, an emerging field that examines how humans and computers can work together to solve problems neither can yet solve alone. We will explore ways in which techniques from machine learning, game theory, optimization, online behavioral social science, and human-computer interactions can be used to model and analyze human-in-the-loop systems such as crowdsourcing markets, prediction markets, and user-generated content platforms. We will also look into recent developments in the interactions between humans and AIs, such as learning with the presence of strategic behavior and ethical issues in AI systems. Prerequisites: CSE 247, ESE 326, Math 233, and Math 309.
Credit 3 units. EN: BME T, TU

E81 CSE 519T Advanced Machine Learning
This course provides a close look at advanced machine learning algorithms, including their theoretical guarantees (computational learning theory) and tricks to make them work in practice. In addition, this course focuses on more specialized learning settings, including unsupervised learning, semi-supervised learning, domain adaptation, multi-task learning, structured prediction, metric learning, and learning of data representations. Learning approaches may include graphical models, non-parametric Bayesian statistics, and technical topics such as sampling, approximate inference, and non-linear function optimization. Mathematical maturity and general familiarity with machine learning are required. Prerequisites: CSE 517A.
Credit 3 units. EN: TU

E81 CSE 520S Real-Time Systems
This course covers software systems and network technologies for real-time applications such as automobiles, avionics, industrial automation, and the Internet of Things. Topics include real-time scheduling, real-time operating systems and middleware, quality of service, industrial networks, and real-time cloud computing. Prerequisite: CSE 361S.
Credit 3 units. EN: BME T, TU

E81 CSE 521S Wireless Sensor Networks
Dense collections of smart sensors networked to form self-configuring pervasive computing systems provide a basis for a new computing paradigm that challenges many classical approaches to distributed computing. Naming, wireless networking protocols, data management, and approaches to dependability, real-time, security, and middleware services all fundamentally change when confronted with this new environment. Embedded sensor networks and pervasive computing are among the most exciting research areas with many open research questions. This course will study a large number of research papers that deal with various aspects of wireless sensor networks. Students will perform a project on a real wireless sensor network comprised of tiny devices, each consisting of sensors, a radio transceiver, and a microcontroller. Prerequisite: CSE 361S.
Credit 3 units. EN: BME T, TU

E81 CSE 522S Advanced Operating Systems
This course offers an in-depth hands-on exploration of advanced uses of key OS abstractions, mechanisms and policies, with an increasing focus on understanding and evaluating their behaviors and interactions. Readings, lecture material, studio exercises, and a semester-long project chosen by students are closely integrated in an active-learning environment in which students gain experience and proficiency writing, tracing, and evaluating user-space and kernel-space code. Topics include how memory, processes and threads,

virtual file systems, and other mechanisms can be used by hypervisors, containers, and other advanced OS abstractions, as well as forensic techniques for examining and managing system behavior. Prerequisite: CSE 422S.
Credit 3 units. EN: BME T, TU

E81 CSE 523S Systems Security
This course examines the intersection between computer design and information security. While performance and efficiency in digital systems have improved markedly in recent decades, computer security has worsened overall in this time frame. To understand why, we will explore the role that design choices play in the security characteristics of modern computer and network systems. Students will use and write software to illustrate mastery of the material. Projects will include identifying security vulnerabilities, exploiting vulnerabilities, and detecting and defending against exploits. Prerequisite: CSE 361S.
Credit 3 units. EN: BME T, TU

E81 CSE 527A Natural Language Processing
Natural language processing (NLP) is an important part of artificial intelligence (AI), endowing computers with the ability to process human language. NLP techniques are used in applications such as question answering, automatic language translation, and extracting structured information from text. This course will introduce fundamental ideas and recent research trends in NLP. Students will gain theoretical and practical experience with various NLP techniques (e.g., deep learning) and applications. Pre-reqs: basic linear algebra, basic probability and statistics, basic machine learning (CSE 417T or ESE 417 or instructor consent) and Python programming.
Credit 3 units. EN: TU

E81 CSE 532S Advanced Multiparadigm Software Development
Intensive focus on advanced design and implementation of concurrent and distributed system software in C++. Topics covered include concurrency and synchronization features and software architecture patterns. Prerequisites: CSE 332S or graduate standing and strong familiarity with C++; and CSE 422S.
Credit 3 units. EN: BME T, TU

E81 CSE 533T Coding and Information Theory for Data Science
Coding/information theory emerged in mid 20th century as a mathematical theory of communication with noise. In latter decades it has developed to a vast topic encompassing most aspects of handling large datasets. The course will begin by surveying the classical mathematical theory and its basic applications in communication, and continue to contemporary applications in storage, computation, privacy, machine learning, and emerging technologies such as networks, blockchains, and DNA storage. The course is self-contained, but prior knowledge in algebra (e.g., Math 309, ESE 318), discrete math (e.g., CSE 240, Math 310), and probability (e.g., Math 2200, ESE 326), as well as some mathematical maturity, is assumed. There will be four to five homework assignments, one in-person midterm, and a final reading assignment.
Credit 3 units. EN: BME T, TU

E81 CSE 534A Large-Scale Optimization for Data Science
Large-scale optimization is an essential component of modern data science, artificial intelligence, and machine learning. This graduate-level course rigorously introduces optimization methods that are suitable for large-scale problems arising in these areas. Students will learn several algorithms suitable for both smooth and nonsmooth optimization, including gradient methods, proximal methods, mirror descent, Nesterov's acceleration, ADMM, quasi-Newton methods, stochastic optimization, variance reduction, and distributed optimization. Throughout the course, we will discuss the efficacy of these methods in concrete data science problems, under appropriate statistical models. Students will be required to program in Python or MATLAB. Prerequisites: CSE 247, Math 309, (Math 3200 or ESE 326), ESE 415.
Same as E35 ESE 513
Credit 3 units. EN: TU

E81 CSE 537T Trustworthy Autonomy
Cyber-physical systems are becoming increasingly capable. These are systems consisting of digital and physical components. They are deployed to increase autonomy in critical settings where failure is costly, such as driving, aviation, medicine, and manufacturing. This course covers several approaches for assuring that such systems meet their specifications. We will discuss: (1) how to mathematically model these systems and their specifications, (2) how to formally verify and synthesize models which meet their specifications, and (3) how to address the challenges resulting from incorporating machine-learned components, such as neural networks, as perception and control components in these systems. Prerequisites: CSE240 and CSE247, or approval of instructor. Preferred prerequisites: Math 217, CSE 347, CSE 417T or ESE 417
Credit 3 units.

E81 CSE 538T Modeling and Performance Evaluation of Computer Systems
Modern computing systems consist of multiple interconnected components that all influence performance. The focus of this course is on developing modeling tools aimed at understanding how to design and provision such systems to meet certain performance or efficiency targets and the trade-offs involved. The course covers Markov chains and their applications to simple queues, and it proceeds to explore more complex systems, including server farms and how to optimize their performance through scheduling and task assignment policies. The course includes a brief review of the necessary probability and mathematical concepts. Prerequisite: ESE 326.
Credit 3 units. EN: BME T, TU

E81 CSE 539S Concepts in Multicore Computing
Nowadays, the vast majority of computer systems are built using multicore processor chips. This fundamental shift in hardware design impacts all areas of computer science - one must write parallel programs in order to unlock the computational power provided by modern hardware. The goal of this course is to study concepts in multicore computing. We will examine the implications of the multicore hardware design, discuss challenges in writing high performance software, and study emerging technologies relevant to developing software for multicore systems. Topics include memory hierarchy, cache coherence protocol, memory models, scheduling, high-level parallel language models, concurrent programming (synchronization and concurrent data structures), algorithms for debugging parallel software, and performance analysis. Prerequisites: CSE 247 and CSE 361S.
Credit 3 units. EN: BME T, TU

E81 CSE 541T Advanced Algorithms
Provides a broad coverage of fundamental algorithm design techniques, with a focus on developing efficient algorithms for solving combinatorial and optimization problems. The topics covered include the review of greedy algorithms, dynamic programming, NP-completeness, approximation algorithms, the use of linear and convex programming for approximation, and online algorithms. Throughout this course, there is an emphasis on correctness proofs and the ability to apply the techniques taught to design efficient algorithms for problems from a wide variety of application areas. Prerequisite: CSE 347.
Credit 3 units. EN: BME T, TU

E81 CSE 543S Advanced Secure Software Engineering
The aim of this course is to provide students with broader and deeper knowledge as well as hands-on experience in understanding security techniques and methods needed in software development. Students complete an independent research project which will involve synthesizing multiple software security techniques and applying them to an actual software program or system.
Credit 3 units. EN: TU

E81 CSE 543T Algorithms for Nonlinear Optimization
The course will provide an in-depth coverage of modern algorithms for the numerical solution of multidimensional optimization problems. Unconstrained optimization techniques including Gradient methods, Newton's methods, Quasi-Newton methods, and conjugate methods will be introduced. The emphasis is on constrained optimization techniques: Lagrange theory, Lagrangian methods, penalty methods, sequential quadratic programming, primal-dual methods, duality theory, nondifferentiable dual methods, and decomposition methods. The course will also discuss applications in engineering systems and use of state-of-the-art computer codes. Special topics may include large-scale systems, parallel optimization, and convex optimization. Prerequisites: Calculus I and Math 309.
Credit 3 units. EN: BME T, TU

E81 CSE 544A Special Topics in Application
The material for this course varies among offerings, but this course generally covers advanced or specialized topics in computer application.
Credit 3 units. EN: BME T, TU

E81 CSE 544M Special Topics in Machines
The material for this course varies among offerings, but this course generally covers advanced or specialized topics in computer science machines.
EN: BME T, TU

E81 CSE 544S Special Topics in Systems
The material for this course varies among offerings, but this course generally covers advanced or specialized topics in computer science systems.
Credit 3 units. EN: BME T, TU

E81 CSE 544T Special Topics in Computer Science Theory
The material for this course varies among offerings, but this course generally covers advanced or specialized topics in computer science theory.
Credit 3 units. EN: BME T, TU

E81 CSE 546T Computational Geometry
Computational geometry is the algorithmic study of problems that involve geometric shapes such as points, lines, and polygons. Such problems appear in computer graphics, vision, robotics, animation, visualization, molecular biology, and geographic information systems. This course covers data structures that are unique to geometric computing, such as convex hull, Voronoi diagram, Delaunay triangulation, arrangement, range searching, KD-trees, and segment trees. Also covered are algorithms for polygon triangulation, path planning, and the art gallery problem. Prerequisite: CSE 347.
Credit 3 units.

E81 CSE 549T Theory of Parallel Systems
The course covers various aspects of parallel programming such as algorithms, schedulers and systems from a theoretical perspective. We will cover both classic and recent results in parallel computing. Topics include parallel algorithms and analysis in the work/span model, scheduling algorithms, external memory algorithms and their analysis, cache-coherence protocols, etc. The focus will be on design and analysis. Prerequisite: CSE 247.
Credit 3 units. EN: BME T, TU

E81 CSE 554A Geometric Computing for Biomedicine
With the advance of imaging technologies deployed in medicine, engineering and science, there is a rapidly increasing amount of spatial data sets (e.g., images, volumes, point clouds) that need to be processed, visualized, and analyzed. This course will focus on a number of geometry-related computing problems that are essential in the knowledge discovery process in various spatial-data-driven biomedical applications. These problems include visualization, segmentation, mesh construction and processing, and shape representation and analysis. This course consists of lectures that cover theories and algorithms, and it includes a series of hands-on programming projects using real-world data collected by various imaging techniques (e.g., CT, MRI, electron cryomicroscopy). Prerequisites: CSE 332 (or proficiency in programming in C++ or Java or Python) and CSE 247.
Credit 3 units. EN: BME T, TU

E81 CSE 555A Computational Photography
Computational Photography describes the convergence of computer graphics, computer vision, and the internet with photography. Its goal is to overcome the limitations of traditional photography using computational techniques to enhance the way we capture, manipulate and interact with visual media. In this course we study many interesting, recent image-based algorithms and implement them to the degree that is possible. Topics may include: cameras and image formation, human visual perception, image processing (filtering, pyramids), image blending and compositing, image retargeting, texture synthesis and transfer, image completion/inpainting, super-resolution, deblurring, denoising, image-based lighting and rendering, high dynamic range, depth and defocus, flash/no flash photography, coded aperture photography, single/multiview reconstruction, photo quality assessment, non photorealistic rendering, modeling and synthesis using internet data, and others. Prerequisites: CSE 452A, CSE 554A, or CSE 559A.
Credit 3 units. EN: BME T, TU

E81 CSE 555T Adversarial AI
This course will introduce students to concepts, theoretical foundations, and applications of adversarial reasoning in Artificial Intelligence. Topics will include the use of machine learning in adversarial settings, such as security, common attacks on machine learning models and algorithms, foundations of game theoretic modeling and analysis in security, with a special focus on algorithmic approaches, and foundations of adversarial social choice, with a focus on vulnerability analysis of elections. Prerequisite: CSE417T
Credit 3 units. EN: BME T, TU

E81 CSE 556A Human-Computer Interaction Methods
An introduction to user centered design processes. The course covers a variety of HCI techniques for use at different stages in the software development cycle, including techniques that can be used with and without users. Students will gain experience using these techniques through in-class exercises and then apply them in greater depth through a semester long interface development project. Students who enroll in this course are expected to be comfortable with building user interfaces in at least one framework and be willing to learn whatever

framework is most appropriate for their project. Over the course of the semester, students will be expected to present their interface evaluation results through written reports and in class presentations. Prerequisites: 3xxS or 4xxS.
Credit 3 units. EN: BME T, TU

E81 CSE 557A Advanced Visualization
We are in an era where it is possible to have all of the world's information at our fingertips. However, the more information we can access, the more difficult it is to obtain a holistic view of the data or to determine what's important to make decisions. Computer-based visualization systems provide the opportunity to represent large or complex data visually to aid comprehension and cognition. In this course, we learn about the state of the art in visualization research and gain hands-on experience with the research pipeline. We also learn how to critique existing work and how to formulate and explore sound research questions. We will cover advanced visualization topics including user modeling, adaptation, personalization, perception, and visual analytics for non-experts. Prerequisite: CSE 457A or permission of instructor.
Credit 3 units. EN: BME T, TU

E81 CSE 559A Computer Vision
This course introduces the fundamentals of designing computational systems that can "look at" images and reason about the physical objects and scenes they represent. Topics include the estimation of color, shape, geometry, and motion from images; image classification, segmentation, and object detection; and image restoration, enhancement, and synthesis. The focus of this course will be on mathematical foundations and practical algorithmic approaches, including: the physics and geometry of image formation; robust methods for estimating image motion and geometry; and deep-learning approaches for semantic image understanding. Students will be required to program in Python. Prerequisites: Any of: CSE 417T (ML), ESE417 (ML), CSE514A (Data Mining), CSE517A (ML).
Credit 3 units. EN: BME T, TU

E81 CSE 560M Computer Systems Architecture I
An exploration of the central issues in computer architecture: instruction set design, addressing and register set design, control unit design, memory hierarchies (cache and main memories, virtual memory), pipelining, instruction scheduling, and parallel systems. The course emphasizes understanding the performance implications of design choices, using architecture modeling and evaluation using simulation techniques. Prerequisites: CSE 361S and CSE 260M.
Credit 3 units. EN: BME T, TU

E81 CSE 563M Digital Integrated Circuit Design and Architecture
This is a project-oriented course on digital VLSI design. The course material focuses on bottom-up design of digital integrated circuits, starting from CMOS transistors, CMOS inverters, combinational circuits and sequential logic designs. Important design aspects of digital integrated circuits such as propagation delay, noise margins and power dissipation are covered in the class, and design challenges in sub-micron technology are addressed. The students design combinational and sequential circuits at various levels of abstraction using a state-of-the-art CAD environment provided by Cadence Design Systems. The goal of the course is to design a microprocessor in 0.5 micron technology that will be fabricated by a semiconductor foundry. Prerequisites: CSE 260M and ESE 232.
Same as E81 CSE 463M
Credit 3 units. EN: BME T, TU

E81 CSE 566S High Performance Computer Systems
Many applications make substantial performance demands upon the computer systems upon which those applications are deployed. In this context, performance is frequently multidimensional, including resource efficiency, power, execution speed (which can be quantified via elapsed run time, data throughput, or latency), and so on. Modern computing platforms exploit parallelism and architectural diversity (e.g., co-processors such as graphics engines and/or reconfigurable logic) to achieve the desired performance goals. This course addresses the practical aspects of achieving high performance on modern computing platforms. This includes questions ranging from how the computing platform is designed to how are applications and algorithms expressed to exploit the platform's properties. Particular attention is given to the role of application development tools. Prerequisite: familiarity with software development in Linux preferred, graduate standing or permission of instructor.
Credit 3 units. EN: BME T, TU

E81 CSE 567M Computer Systems Analysis
A comprehensive course on performance analysis techniques. The topics include common mistakes, selection of techniques and metrics, summarizing measured data, comparing systems using random data, simple linear regression models, other regression models, experimental designs, 2**k experimental designs, factorial designs with replication, fractional factorial designs, one factor experiments, two factor full factorial design w/o replications, two factor full factorial designs with replications, general full factorial designs, introduction to queueing theory, analysis of single queues, queueing networks, operational laws, mean-value analysis, time series analysis, heavy tailed distributions, self-similar processes, long-range dependence, random number generation, analysis of simulation results, and art of data presentation. Prerequisites: CSE 260M.
Credit 3 units. EN: BME T, TU

E81 CSE 569S Recent Advances in Computer Security and Privacy
The aim of this course is to provide students with knowledge and hands-on experience in understanding the security techniques and methods needed for IoT, real-time, and embedded systems. Students complete an independent research project which will involve synthesizing multiple security techniques and applying them to an actual IoT, real-time, or embedded system or device.
Credit 3 units. EN: BME T, TU

E81 CSE 570S Recent Advances in Networking
This course covers the latest advances in networking. The topics include Networking Trends, Data Center Network Topologies, Data Center Ethernet, Server Virtualization, Storage Virtualization, Carrier Ethernet, Application Delivery Networking, Virtual Bridging, Big Data, Networking Issues for Big Data, LAN Extension and Virtualization using Layer 3 Protocols, Multi-Tenant Isolation and Network Virtualization in Cloud Data Centers, OpenFlow, OpenFlow Controllers and Tools, Software Defined Networking (SDN), Network Function Virtualization (NFV), Internet of Things (IoT), Networking Protocols for IoT, Blockchains, and Quantum Communications. Students also do a project/survey involving application of these topics.
Credit 3 units. EN: BME T, TU

E81 CSE 571S Network Security
This course covers principles and techniques in securing computer networks. Real world examples will be used to illustrate the rationales behind various security designs. There are three main components in the course, preliminary cryptography, network protocol security and network application security. Topics include IPSec, SSL/TLS, HTTPS, network fingerprinting, network malware, anonymous communication, and blockchain. The class project allows students to take a deep dive into a topic of choice in network security. Prerequisite: CSE 473S.

Credit 3 units. EN: BME T, TU

E81 CSE 574S Recent Advances in Wireless and Mobile Networking
This course provides a comprehensive treatment of wireless data and telecommunication networks. Topics include recent trends in wireless and mobile networking, wireless coding and modulation, wireless signal propagation, IEEE 802.11a/b/g/n/ac wireless local area networks, 60 GHz millimeter wave gigabit wireless networks, vehicular wireless networks, white spaces, Bluetooth and Bluetooth Smart, wireless personal area networks, wireless protocols for the Internet of Things, cellular networks: 1G/2G/3G, LTE, LTE-Advanced, and 5G. Prerequisite: CSE 473S (Introduction to Computer Networks) or permission of instructor.
Credit 3 units. EN: BME T, TU

E81 CSE 581T Approximation Algorithms
Numerous optimization problems are intractable to solve optimally. The intractability of a problem could come from the problem's computational complexity, for instance the problem is NP-Hard, or other computational barriers. To cope with the inability to find an optimal algorithm, one may desire an algorithm that is guaranteed to return a solution that is comparable to the optimum. Such an algorithm is known as an approximation algorithm. Approximation algorithms are a robust way to cope with intractability, and they are widely used in practice or are used to guide the development of practical heuristics. The area of approximation algorithms has developed a vast theory, revealing the underlying structure of problems as well as their different levels of difficulty. The majority of this course will focus on fundamental results and widely applicable algorithmic and analysis techniques for approximation algorithms. Prerequisite: CSE 347.
Credit 3 units. EN: BME T, TU

E81 CSE 584A Algorithms for Biosequence Comparison
This course surveys algorithms for comparing and organizing discrete sequential data, especially nucleic acid and protein sequences. Emphasis is on tools to support search in massive biosequence databases and to perform fundamental comparison tasks such as DNA short-read alignment. Prerequisite: CSE 347 or permission of instructor. These techniques are also of interest for more general string processing and for building and mining textual databases. Algorithms are presented rigorously, including proofs of correctness and running time where feasible. Topics include classical string matching, suffix array string indices, space-efficient string indices, rapid inexact matching by filtering (including BLAST and related tools), and alignment-free algorithms. Students complete written assignments and implement advanced comparison algorithms to address problems in bioinformatics. This course does not require a biology background. Prerequisite: CSE 347 or permission of instructor.
Credit 3 units. EN: BME T, TU

E81 CSE 586A Analysis of Imaging Data
This course focuses on an in-depth study of advanced topics and interests in image data analysis. Students will learn about hardcore imaging techniques and gain the mathematical fundamentals needed to build their own models for effective problem solving. Topics of deformable image registration, numerical analysis, probabilistic modeling, data dimensionality reduction, and convolutional neural networks for image segmentation will be covered. The main focus might change from semester to semester. Prerequisites: Math 309, ESE 326, and CSE 247.
Credit 3 units.

E81 CSE 587A Algorithms for Computational Biology
This course is a survey of algorithms and mathematical methods in biological sequence analysis (with a strong emphasis on probabilistic methods) and systems biology. Sequence analysis topics include introduction to probability, probabilistic inference in missing data problems, hidden Markov models (HMMs), sequence alignment, and identification of transcription-factor binding sites. Systems biology topics include the mapping of gene regulatory networks, quantitative modeling of gene regulatory networks, synthetic biology, and applications of deep learning in computational biology. Prerequisite: CSE 131 or CSE 501N.
Credit 3 units. EN: BME T, TU

E81 CSE 591 Introduction to Graduate Study in CSE
Introduces students to the different areas of research conducted in the department. Provides an introduction to research skills, including literature review, problem formulation, presentation, and research ethics. Lecture and discussion are supplemented by exercises in the different research areas and in critical reading, idea generation, and proposal writing.
Credit 3 units.

E81 CSE 598 Master's Project
Students electing the project option for their master's degree perform their project work under this course. In order to successfully complete this course, students must defend their project before a three-person committee and present a 2-3 page extended abstract. Prerequisite: permission of advisor and submission of a research proposal form.
Credit variable, maximum 6 units.

E81 CSE 599 Master's Research
Students electing the thesis option for their master's degree perform their thesis research under this course. In order to successfully complete a master's thesis, students must enroll in 6 units of this course typically over the course of two consecutive semesters, produce a written thesis, and defend the thesis before a three-person committee. Prerequisite: permission of advisor and submission of a research proposal form.
Credit variable, maximum 6 units.

E81 CSE 637S Software Security
In this course, students will be introduced to the foundations of software security. We will explore different classes of software vulnerabilities and analyze the fundamental problems behind these vulnerabilities, and we will study the methods and techniques used to discover, exploit, prevent, and mitigate these vulnerabilities. Topics of interest include buffer overflow, integer overflow, type confusion, use-after-free, and so on. Throughout the course, we take a defense-in-depth mentality and see how systems can be protected. Students are expected to have a solid understanding of assembly language, C/C++, and operating systems. Prerequisite: CSE 361S.
Credit 3 units.

E81 CSE 659A Advances in Computer Vision
Computer vision is a fast-moving field, with the past few years seeing tremendous advances in the development of computational algorithms for solving visual tasks. This course is designed to introduce students to advanced and recently published techniques for problems in low-level vision, recognition and classification, and computational photography. Prerequisite: CSE 559A.
Credit 3 units.

E81 CSE 699 Doctoral Research
Credit variable, maximum 9 units.

E81 CSE 7001 Research Seminar on Computer Science Pedagogy
This seminar will examine research, techniques, approaches, and strategies for teaching computer science at the undergraduate and graduate levels. Participants will take turns presenting a particular paper or concept and then leading an ensuing discussion. While this seminar may be especially helpful for those contemplating an academic career, the seminar is open to all interested participants.
Credit 1 unit.

E81 CSE 7100 Research Seminar on Machine Learning
Research seminars examine publications, techniques, approaches and strategies within an area of computer science and engineering. Seminars are highly participational: Students are expected to take turns presenting material, to prepare for seminar by reading any required material, and to contribute to the group's discussions. The actual topics covered in a seminar will vary by semester and instructor. Interested students are encouraged to obtain a syllabus from the instructor's webpage or by contacting the instructor.
Credit 1 unit.

E81 CSE 7200 Research Seminar on Robotics and Human-Computer Interaction
Research seminars examine publications, techniques, approaches and strategies within an area of computer science and engineering. Seminars are highly participational: Students are expected to take turns presenting material, to prepare for seminar by reading any required material, and to contribute to the group's discussions. The actual topics covered in a seminar will vary by semester and instructor. Interested students are encouraged to obtain a syllabus from the instructor's webpage or by contacting the instructor.
Credit 1 unit.

E81 CSE 7300 Research Seminar on Software Systems
Research seminars examine publications, techniques, approaches and strategies within an area of computer science and engineering. Seminars are highly participational: Students are expected to take turns presenting material, to prepare for seminar by reading any required material, and to contribute to the group's discussions. The actual topics covered in a seminar will vary by semester and instructor. Interested students are encouraged to obtain a syllabus from the instructor's webpage or by contacting the instructor.
Credit 1 unit.

E81 CSE 7400 Research Seminar on Algorithms and Theory
Research seminars examine publications, techniques, approaches and strategies within an area of computer science and engineering. Seminars are highly participational: Students are expected to take turns presenting material, to prepare for seminar by reading any required material, and to contribute to the group's discussions. The actual topics covered in a seminar will vary by semester and instructor. Interested students are encouraged to obtain a syllabus from the instructor's webpage or by contacting the instructor.
Credit 1 unit.

E81 CSE 7500 Research Seminar on Graphics and Vision
Research seminars examine publications, techniques, approaches and strategies within an area of computer science and engineering. Seminars are highly participational: Students are expected to take turns presenting material, to prepare for seminar by reading any required material, and to contribute to the group's discussions. The actual topics covered in a seminar will vary by semester and instructor. Interested students are encouraged to obtain a syllabus from the instructor's webpage or by contacting the instructor.
Credit 1 unit.

E81 CSE 7600 Research Seminar on Analog Computing
This seminar will focus on classic and recent papers on analog, stochastic and neuromorphic computing. Students will read, present, and discuss journal papers on analog techniques for implementing sensors and processors. Focus will be placed on fundamental advances and challenges of implementing analog processors. No prerequisites.
Credit 1 unit.

E81 CSE 7700 Research Seminar on Networking and Communications
Research seminars examine publications, techniques, approaches and strategies within an area of computer science and engineering. Seminars are highly participational: Students are expected to take turns presenting material, to prepare for seminar by reading any required material, and to contribute to the group's discussions. The actual topics covered in a seminar will vary by semester and instructor. Interested students are encouraged to obtain a syllabus from the instructor's webpage or by contacting the instructor.
Credit 1 unit.

E81 CSE 7800 Research Seminar on Computational Systems Biology
Research seminars examine publications, techniques, approaches and strategies within an area of computer science and engineering. Seminars are highly participational: Students are expected to take turns presenting material, to prepare for seminar by reading any required material, and to contribute to the group's discussions. The actual topics covered in a seminar will vary by semester and instructor. Interested students are encouraged to obtain a syllabus from the instructor's webpage or by contacting the instructor.
Credit 1 unit.

E81 CSE 7900 Research Seminar on Parallel Computing
This seminar will focus on classic and recent papers on parallel computing. Students will read, present and discuss papers on parallel models, algorithms and architectures from top conferences and journals. Focus will be placed on fundamental advances and theoretical models and algorithms, rather than on implementation papers. No prerequisites.
Credit 1 unit.

E81 CSE 801 Pedagogy
A student taking this course studies the fundamentals of teaching in the discipline of computer science and computer engineering. A student enrolled in this course staffs some other course taught by our department, serving as its primary instructor or co-instructor. That student receives frequent mentoring and feedback on preparation and delivery. This course is recommended especially for doctoral students who seek a career in computer science and engineering education.
Credit 3 units.