In the past two decades, society has experienced unprecedented growth in digital technology. This revolution continues to redefine our way of life, culture and 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 a part-time student but 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 or Computer Engineering degrees only) or 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, though 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, and 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 in their first year and the program emphasis is on creative research. The program has milestones with both written and oral components that provide structure to 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 background is outside of computer science.

**Phone:** 314-935-6132  
**Email:** admissions@cse.wustl.edu  
**Website:** https://cse.wustl.edu/graduate/programs

**Faculty**

**Chair**  
Roch Guérin  
Harold B. and Adelaide G. Welge Professor of Computer Science  
PhD, California Institute of Technology  
Computer networks and communication systems

**Professors**  
Sanjoy Baruah  
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  
James M. McKelvey Professor and Dean  
PhD, Massachusetts Institute of Technology  
Computer vision, graphics, human-robot collaboration

Michael R. Brent  
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  
PhD, Washington University  
Computational biology, genomics, algorithms for comparing and annotating large biosequences
Roger D. Chamberlain (https://engineering.wustl.edu/Profiles/Pages/Roger-Chamberlain.aspx)
DSc, Washington University
Computer engineering, parallel computation, computer architecture, multiprocessor systems

Yixin Chen (https://engineering.wustl.edu/Profiles/Pages/Yixin-Chen.aspx)
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/Profiles/Pages/Patrick-Crowley.aspx)
PhD, University of Washington
Computer and network systems, network security

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

Christopher D. Gill (https://engineering.wustl.edu/Profiles/Pages/Christopher-Gill.aspx)
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/Profiles/Pages/Raj-Jain.aspx)
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/Profiles/Pages/Tao-Ju.aspx)
PhD, Rice University
Computer graphics, visualization, mesh processing, medical imaging and modeling

Chenyang Lu (https://engineering.wustl.edu/Profiles/Pages/Chenyang-Lu.aspx)
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

Weixiong Zhang (https://engineering.wustl.edu/Profiles/Pages/Weixiong-Zhang.aspx)
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/Profiles/Pages/Kunal-Agrawal.aspx)
PhD, Massachusetts Institute of Technology
Parallel computing, cyber-physical systems & sensing, theoretical computer science

Sanmay Das (https://engineering.wustl.edu/Profiles/Pages/Sanmay-Das.aspx)
PhD, Massachusetts Institute of Technology
Design of algorithms for complex environments, computational social science, machine learning

Caitlin Kelleher (https://engineering.wustl.edu/Profiles/Pages/Caitlin-Kelleher.aspx)
Hugo F. & Ina Champ Urbauer Career Development Associate Professor
PhD, Carnegie Mellon University
Human-computer interaction, programming environments, and learning environments

Assistant Professors

Ayan Chakrabarti (https://engineering.wustl.edu/Profiles/Pages/Ayan-Chakrabarti.aspx)
PhD, Harvard University
Computer vision computational photography, machine learning

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

Chien-Ju Ho (https://engineering.wustl.edu/Profiles/Pages/Chien-Ju-Ho.aspx)
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

Brendan Juba (https://engineering.wustl.edu/Profiles/Pages/Brendan-Juba.aspx)
PhD, Massachusetts Institute of Technology
Theoretical approaches to artificial intelligence founded on computational complexity theory and theoretical computer science more broadly construed
Ulugbek Kamilov ([https://engineering.wustl.edu/Profiles/Pages/Ulugbek-Kamilov.aspx](https://engineering.wustl.edu/Profiles/Pages/Ulugbek-Kamilov.aspx))
PhD, École Polytechnique Fédérale de Lausanne, Switzerland
Computational imaging, image and signal processing, machine learning and optimization

Brian Kocoloski ([https://cse.wustl.edu/faculty/Pages/faculty.aspx?bio=115](https://cse.wustl.edu/faculty/Pages/faculty.aspx?bio=115))
PhD, University of Pittsburgh
Scalable parallel computing, cloud computing, operating systems, virtualization

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

Alvitta Ottley ([https://cse.wustl.edu/faculty/Pages/faculty.aspx?bio=109](https://cse.wustl.edu/faculty/Pages/faculty.aspx?bio=109))
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

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

**Professor of the Practice**

Dennis Cosgrove ([https://cse.wustl.edu/faculty/Pages/Dennis-Cosgrove.aspx](https://cse.wustl.edu/faculty/Pages/Dennis-Cosgrove.aspx))
BS, University of Virginia
Programming environments and parallel programming

**Lecturers**

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

Jonathan Shidal ([https://cse.wustl.edu/faculty/Pages/Jon-Shidal.aspx](https://cse.wustl.edu/faculty/Pages/Jon-Shidal.aspx))
PhD, Washington University
Computer architecture and memory management

Douglas Shook ([https://cse.wustl.edu/faculty/Pages/Doug-Shook.aspx](https://cse.wustl.edu/faculty/Pages/Doug-Shook.aspx))
MS, Washington University
Imaging sensor design, compiler design and optimization

William Siever ([https://cse.wustl.edu/faculty/Pages/Bill-Siever.aspx](https://cse.wustl.edu/faculty/Pages/Bill-Siever.aspx))
Principal Lecturer
PhD, Missouri University of Science and Technology

Todd Sproull ([https://cse.wustl.edu/faculty/Pages/Todd-Sproull.aspx](https://cse.wustl.edu/faculty/Pages/Todd-Sproull.aspx))
PhD, Washington University
Computer networking and mobile application development

**Senior Professors**

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

Mark A. Franklin
Hugo F. and Ina Champ Urbauer Professor of Engineering
PhD, Carnegie Mellon University
Computer architecture, systems analysis and parallel processing, storage systems design

Jonathan S. Turner
PhD, Northwestern University
Design and analysis of internet routers and switching systems, networking and communications, algorithms

**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 the following pages for information about the

- Master of Science (MS) in Computer Science ([http://bulletin.wustl.edu/grad/engineering/computerscience/ms-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](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](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](http://bulletin.wustl.edu/grad/engineering/computerscience/meng-computer-science))
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
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. Prerequisite: CSE 240.
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 the AngularJS web framework, 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 a 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: 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 247. Same as E81 CSE 332S
Credit 3 units. EN: 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. EN: TU

E81 CSE 507A Technology Entrepreneurship
This is a course for students who plan to be, or work with, entrepreneurs. An entrepreneurial mindset is needed to create or grow economically viable enterprises, be they new companies, new groups within companies, or new university laboratories. This course aims to cultivate an entrepreneurial perspective with particular emphasis on information technology (IT)-related activities. The course is jointly offered for business and CSE students, allowing for acculturation between these disciplines. In addition to an introductory treatment of business and technology fundamentals, course topics will include: business ethics, opportunity assessment, team formation, financing, intellectual property, and technology transfer. The course will feature significant participant and guest instruction from experienced practitioners. Prerequisites: none.
Credit 3 units.

E81 CSE 511A Introduction to Artificial Intelligence
The discipline of artificial intelligence (AI) is concerned with building systems that think and act like humans or rationally on some absolute scale. This course is an introduction to the field, with special emphasis on sound modern methods. The topics include knowledge representation, problem solving via search, game playing, logical and probabilistic reasoning, planning, dynamic programming, and reinforcement learning. Programming exercises concretize the key methods. The course targets graduate students and advanced undergraduates. Evaluation is based on written and programming assignments, a midterm exam and a final exam. Prerequisites: CSE 247, ESE 326, Math 233.
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 advancement in science and technology, data acquisition in large quantities are routinely done 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. Mining a large amount of data through data mining has become an effective means to extracting knowledge from data. This course introduces the basic concepts and methods for 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 learning. Prerequisites: CSE 247 and ESE 326 (or Math 3200).
Credit 3 units. EN: 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, then proceed to cover more-advanced topics. These will include inference techniques (exact, MAP, sampling methods, the Laplace approximation, etc.), Bayesian decision theory, Bayesian model comparison, Bayesian nonparametrics, and Bayesian optimization. Prerequisites: CSE 247 and ESE 326 (or Math 3200).
Credit 3 units. EN: 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: TU

E81 CSE 517A Machine Learning

This course provides a close look at advanced machine learning algorithms — 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 of machine learning is required. Prerequisites: CSE 517A, CSE 511A, and CSE 571A.
Credit 3 units. EN: TU

E81 CSE 519T Advanced Machine Learning

This course provides a close look at advanced machine learning approaches — 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 of machine learning is required. Prerequisites: CSE 517A, CSE 511A, and CSE 571A.
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 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 422S.
Credit 3 units. EN: 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 class studies a large number of research papers that deal with various aspects of wireless sensor networks. Students perform a project on a real wireless sensor network composed of tiny devices each
E81 CSE 522S Advanced Operating Systems
This course offers an in-depth hands-on exploration of core OS abstractions, mechanisms and policies, with an increasing focus on understanding and evaluating their behaviors and interactions. Readings, lecture material, studio exercises, and lab assignments 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: inter-process communication, real-time systems, memory forensics, file-system forensics, timing forensics, process and thread forensics, hypervisor forensics, and managing internal or external causes of anomalous behavior. Prerequisite: CSE 422S.
Credit 3 units. EN: 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: TU

E81 CSE 530S Database Management Systems
A study of data models and the database management systems that support these data models. The design theory for databases is developed and various tools are utilized to apply the theory. General query languages are studied and techniques for query optimization are investigated. Integrity and security requirements are studied in the context of concurrent operations on a database, where the database may be distributed over one or more locations. The unique requirements for engineering design databases, image databases, and long transaction systems are analyzed. Prerequisite: CSE 247.
Credit 3 units.

E81 CSE 531S Theory of Compiling and Language Translation
Algorithms and intermediate representations for automatic program analysis are examined, with an emphasis on practical methods and efficient engineering of program optimization and transformations. The course includes a thorough treatment of monotone data flow frameworks: a mathematical model in which most optimization problems can be specified and solved. The course primarily covers optimizations that are applicable to any target architecture; however, optimizations specific to parallel, distributed and storage-hierarchical systems also are discussed. Prerequisite: CSE 431S or CSE 425S.
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: TU

E81 CSE 536S Distributed System Design: Models and Languages
Modern computing environments are highly distributed. This has been the result of major advances in networking technology and their rapid assimilation by a society that functions in a highly distributed and decentralized manner. The goal of this course is to familiarize students with basic concepts, models and languages that shaped recent developments in distributed computing. The focus is on exploring new ways of thinking about computing and communication that made the development of distributed software systems possible. Competing concepts and design strategies will be examined both from a theoretical and a practical perspective. Prerequisites: CSE 240 and CSE 247.
Credit 3 units. EN: TU

E81 CSE 538T Modeling and Performance Evaluation of Interconnected Computer Systems
Modern computing systems consist of multiple interconnected components, which 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 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: 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 332S and CSE 361S.
Credit 3 units. EN: TU

E81 CSE 541T Advanced Algorithms
Provides a broad coverage of fundamental algorithm design techniques with the focus on developing efficient algorithms for solving combinatorial and optimization problems. The topics covered include: greedy algorithms, dynamic programming, linear programming, NP-completeness, approximation algorithms, lower bound techniques, 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. Prerequisites: CSE 347. Credit 3 units. EN: TU

E81 CSE 542T Advanced Data Structures and Algorithms
This course is concerned with the design and analysis of efficient algorithms, focusing principally on algorithms for combinatorial optimization problems. A key element in the course is the role of data structures in algorithm design and the use of amortized complexity analysis to determine how data structures affect performance. The course is organized around a set of core problems and algorithms, including the classical network optimization algorithms, as well as newer and more efficient algorithms. This core is supplemented by algorithms selected from the recent technical literature. Prerequisite: CSE 247. Credit 3 units.

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, 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.

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. A description for a given semester's offering will appear in that semester's course guide. Credit 3 units. EN: 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, shortest paths, the post office problem, and the art gallery problem. Prerequisite: CSE 247. Credit 3 units.

E81 CSE 547T Introduction to Formal Languages and Automata
An introduction to the theory of computation, with emphasis on the relationship between formal models of computation and the computational problems solvable by those models. Specifically, this course covers finite automata and regular languages; Turing machines and computability; and basic measures of computational complexity and the corresponding complexity classes. Prerequisites: CSE 240 and CSE 247. Credit 3 units.

E81 CSE 548T Concurrent Systems: Design and Verification
Formerly CSE 563T. Concurrency presents programmers with unprecedented complexity further exacerbated by our limited ability to reason about concurrent computations. Yet, concurrent algorithms are central to the development of software executing on modern multiprocessors or across computer networks. This course reviews several important classes of concurrent algorithms and presents a formal method for specifying, reasoning about, verifying, and deriving concurrent algorithms. The selected algorithms are judged to have made significant contributions to our understanding of concurrency. Rigorous treatment of the design and programming process is emphasized. Students entering this course must be familiar with predicate calculus and sequential algorithms. Upon completion of this course students will be able to reason completely formally about small concurrent programs and to apply systematically and correctly their formal skills to larger problems. Prerequisite: CSE 247. 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: TU

E81 CSE 552A Advanced Computer Graphics
This course covers advanced topics in graphics in the areas of modeling, rendering, volume rendering, image-based rendering and image processing. Topics include, but are not limited to, subdivision surfaces, splines, mesh simplification, implicit or blobby modeling, radiosity, procedural textures, filtering, BRDFs and procedural modeling. The class has several structured programming assignments and an optional final group project. Students are exposed to the wide variety of techniques available in graphics and also pick one area to study in depth. Prerequisites: CSE 332S and CSE 452A. Credit 3 units. EN: 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 (images, volumes, point clouds, etc.) 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
computing a sequence of motions that transforms a given (initial) arrangement of physical objects to another (goal) arrangement of those objects. Many motion planning methods were developed in the realm of robotics research. For example, a typical problem might be to find a sequence of motions (called a path) to move a robot from one position to another without colliding with any objects in its workspace. However, the general motion planning problem that will be studied arises in many other application domains as well. For example, assembly planning (e.g., finding a valid order for adding the parts when building an engine), mechanical CAD studies (e.g., can you remove a certain part from an engine without taking the engine apart), artificial life simulations (e.g., moving a herd of animals from one location to another), and medicine (e.g., can a drug molecule reach a protein molecule). Prerequisite: CSE 247. Credit 3 units. EN: 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: 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: TU

E81 CSE 557A Information 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 and/or complex data visually to aid comprehension and cognition. In this course, we study the principles for transforming abstract data into effective information visualizations. We learn about the state-of-the-art in visualization research and development, and we gain hands-on experience with designing and developing information visualizations. We also learn how to critique existing visualizations and how to evaluate the systems we build. Readings will include current research papers from the Information Visualization community. Prerequisites: CSE 247 and CSE 330S. Credit 3 units. EN: TU

E81 CSE 558A Motion Planning
This course studies the general motion planning problem: computing a sequence of motions that transforms a given (initial) arrangement of physical objects to another (goal) arrangement of those objects. Many motion planning methods were developed in the realm of robotics research. For example, a typical problem might be to find a sequence of motions (called a path) to move a robot from one position to another without colliding with any objects in its workspace. However, the general motion planning problem that will be studied arises in many other application domains as well. For example, assembly planning (e.g., finding a valid order for adding the parts when building an engine), mechanical CAD studies (e.g., can you remove a certain part from an engine without taking the engine apart), artificial life simulations (e.g., moving a herd of animals from one location to another), and medicine (e.g., can a drug molecule reach a protein molecule). Prerequisite: CSE 247. Credit 3 units. EN: TU

E81 CSE 559A Computer Vision
This course introduces the fundamentals of designing computer vision systems: that can “look at” images and videos and reason about the physical objects and scenes they represent. Topics include image restoration and enhancement; estimation of color, shape, geometry, and motion from images; and image segmentation, recognition, and classification. The focus of the course will be on the mathematical tools and intuition underlying algorithms for these tasks: models for the physics and geometry of image formation, and statistical and machine learning-based techniques for inference. Prerequisites: Math 309 and CSE 247. Credit 3 units. EN: 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, microprogramming, memory hierarchies (cache and main memories, mass storage, virtual memory), pipelining, and bus organization. The course emphasizes understanding the performance implications of design choices, using architecture modeling and evaluation using VHDL and/or instruction set simulation. Prerequisites: CSE 361S and CSE 260M. Credit 3 units. EN: TU

E81 CSE 561M Computer Systems Architecture II
Advanced techniques in computer system design. Selected topics from: processor design (multithreading, VLIW, data flow, chip-multiprocessors, application specific processors, vector units, large MIMD machines), memory systems (topics in locality, prefetching, reconfigurable and special-purpose memories), system specification and validation, and interconnection networks. Prerequisites: CSE 560M. Credit 3 units. EN: TU

E81 CSE 564M Advanced Digital Systems Engineering
This course focuses on advanced sensor design. The class covers various basic analog and digital building blocks that are common in most sensor integrated circuits. The class extensively uses state-of-the-art CAD program Cadence to simulate and analyze various circuit blocks. The first half of the course focuses on analyzing various operational amplifiers, analog filters, analog memory and analog to digital converters. The second half of the course focuses on understanding the basic building blocks of imaging sensors. The class has a final project consisting of designing a smart sensor using Cadence tools. Prerequisites: ESE 232 and CSE 362M. Credit 3 units. EN: TU
E81 CSE 565M Acceleration of Algorithms in Reconfigurable Logic
Reconfigurable logic, in the form of Field-Programmable Gate Arrays (FPGAs), enables the deployment of custom hardware for individual applications. To exploit this capability, the application developer is required to specify the design at the register-transfer level. This course explores techniques for designing algorithms that are amenable to hardware acceleration as well as provides experience in actual implementation. Example applications are drawn from a variety of fields, such as networking, computational biology, etc. Prerequisites: basic digital logic (CSE 260M) and some experience with a hardware description language (e.g., VHDL or Verilog). Credit 3 units. EN: 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), etc. 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: 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^n experimental designs, factorial designs with replication, fractional factorial designs, one factor experiments, two factor full factorial design with replications, two factor full factorial designs with replications, general full factorial designs, introduction to queuing 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: TU

E81 CSE 568M Imaging Sensors
This course will cover topics on digital imaging sensors including basic operations of silicon photodetectors; CCD and CMOS passive and active sensor operation; temporal and spatial noise in CMOS sensors; spatial resolution and MTF; SNR and dynamic range; high dynamic range architectures and application specific imaging sensors such as polarization imaging and fluorescent imaging sensors. Prerequisites: CSE 260M and ESE 232. Credit 3 units.

E81 CSE 569M Parallel Architectures and Algorithms
A number of contemporary parallel computer architectures are reviewed and compared. The problems of process synchronization and load balancing in parallel systems are studied. Several selected applications problems are investigated and parallel algorithms for their solution are considered. Selected parallel algorithms are implemented in both a shared memory and distributed memory parallel programming environment. Prerequisites: graduate standing and knowledge of the C programming language. Credit 3 units. EN: TU

E81 CSE 569S Advanced IoT, Real-Time, and Embedded Systems Security
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: TU

E81 CSE 570S Recent Advances in Networking

E81 CSE 571S Network Security

E81 CSE 573S Protocols for Computer Networks
An introduction to the design, performance analysis and implementation of existing and emerging computer network protocols. Protocols include multiple access protocols (e.g., CSMA/CD, token ring), internet, working with the internet protocol (IP), transport protocols (e.g., UDP, TCP), high-speed bulk transfer protocols, and routing protocols (e.g., BGP, OSPF). General topics include error control, flow control, packet switching, mechanisms for reliable, ordered and bounded-time packet delivery, host-network interfacing and protocol implementation models. Substantial programming exercises
supplement lecture topics. Prerequisite: CSE 473S or permission of the instructor. Credit 3 units. EN: TU

E81 CSE 574S Wireless and Mobile Networking
First course in wireless networking providing 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, IEEE 802.22 regional area networks, Bluetooth and Bluetooth Smart, wireless personal area networks, wireless protocols for Internet of Things, ZigBee, cellular networks: 1G/2G/3G, LTE, LTE-Advanced, and 5G. Prerequisites: CSE 473S or permission of the instructor. Credit 3 units. EN: TU

E81 CSE 577M Design and Analysis of Switching Systems
Switching is a core technology in a wide variety of communication networks, including the internet, circuit-switched telephone networks and optical fiber transmission networks. The last decade has been a time of rapid development for switching technology in the internet. Backbone routers with 10 Gb/s links and aggregate capacities of hundreds of gigabits per second are becoming common, and advances in technology are now making multi-terabit routers practical. This course is concerned with the design of practical switching systems and evaluation of their performance and complexity. Prerequisites: CSE 473S and ESE 326. Credit 3 units. EN: 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-complete 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: TU

E81 CSE 582T Complexity Theory
An introduction to the quantitative theory of computation with limited resources. The course examines the relative power of limited amounts of basic computational resources, such as time, memory, circuit size, and random bits, as well as parallel, nondeterministic, alternating, and interactive machine models. Models that capture special kinds of computational problems, such as counting problems or approximate solutions, will also be introduced and related to the standard models. This examination will emphasize surprising relationships between seemingly disparate resources and kinds of computational problems. The course will also discuss some meta-theory, illuminating the weaknesses of standard mathematical techniques of the field against its notorious open conjectures. Prerequisites: CSE 347. Credit 3 units. EN: TU

E81 CSE 583A Topics in Computational Molecular Biology
In-depth discussion of problems and methods in computational molecular biology. Each year three topics will be covered and those will change yearly. Prerequisite: Biol 5495 or instructor's consent. Same as L41 Biol 5497 Credit 2 units.

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. 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 multiple alignment. Students complete written assignments and implement advanced comparison algorithms to address problems in bioinformatics. This course does not require a biology background. Prerequisites: CSE 347. Credit 3 units. EN: TU

E81 CSE 585T Sparse Modeling for Imaging and Vision
Sparse modeling is at the heart of modern imaging, vision, and machine learning. It is a fascinating new area of research that seeks to develop highly effective data models. The core idea in sparse modeling is a novel redundant transform, where the number of transform coefficients is larger compared to the original data dimension. Together with redundancy comes an opportunity of seeking the sparest possible representation, or the one with the fewest non-zeros. This core idea leads to a series of beautiful theoretical and practical results with many applications such as regression, prediction, restoration, extrapolation, compression, detection, and recognition. In this course, we will explore sparse modeling by covering theoretical as well as algorithmic aspects with applications in computational imaging and computer vision. Prerequisites: ESE 318, Math 233, Math 309, and Math 429, or equivalents. Coding with MATLAB or Python. Credit 3 units. EN: TU

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), profile HMMs, sequence alignment, and identification of transcription-factor binding sites. Systems biology topics include discovery of gene regulatory networks, quantitative modeling of gene regulatory networks, synthetic biology, and (in some years) quantitative modeling of metabolism. Prerequisites: CSE 131. Credit 3 units.
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 adviser 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 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 adviser and submission of a research proposal form. Credit variable, maximum 6 units.

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

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.

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.

E81 CSE 883 Master’s Continuing Student Status