**COMPUTER SCIENCE**

A. K. Watson Hall, 203.432.1246  
http://cpsc.yale.edu  
M.S., M.Phil., Ph.D.

**Chair**  
Zhong Shao

**Director of Graduate Studies**  
Vladimir Rokhlin (108 AKW, 203.432.1283, vladimir.rokhlin@yale.edu)

**Professors**  Dana Angluin, James Aspnes, Dirk Bergemann,* Ronald Coifman,* Julie Dorsey, Stanley Eisenstat, Joan Feigenbaum, Michael Fischer, David Gelernter, Mark Gerstein,* Rajit Manohar,* Drew McDermott, Dragomir Radev, Vladimir Rokhlin,† Holly Rushmeier, Brian Scassellati, Martin Schultz (Emeritus), Zhong Shao, Avi Silberschatz, Daniel Spielman, Leandros Tassiulas,* Y. Richard Yang, Steven Zucker†

**Associate Professors**  Mahesh Balakrishnan, Minlan Yu

**Assistant Professors**  Wenjun Hu,* Julian Jara-Ettinger,* Amin Karbasi,* Smita Krishnaswamy,* Sahand Negahban,* Ruzica Piskac, Mariana Raykova, Jakub Szefer*

**Senior Lecturers**  Kyle Jensen,* Stephen Slade

**Lecturers**  Benedict Brown, James Glenn, Scott Petersen, Brad Rosen, Andrew Sherman, Xiyin Tang [Sp]

* A secondary appointment with primary affiliation in another department or school.  
† A joint appointment with another department.

**FIELDS OF STUDY**

Algorithms and computational complexity, artificial intelligence, data networking, databases, graphics, machine learning, programming languages, robotics, scientific computing, security and privacy, systems.

**RESEARCH FACILITIES**

The department operates a high-bandwidth, local-area computer network based mainly on distributed workstations and servers, with connections to worldwide networks. Workstations include Dell dual-processor PCs (running Linux or Windows/XP). Laboratory contains specialized equipment for graphics, vision, and robotics research. Various printers, including color printers, as well as image scanners, are also available. The primary educational facility consists of thirty-seven PC workstations supported by a large Intel PC server. This facility is used for courses and unsponsored research by Computer Science majors and first-year graduate students. Access to computing, through both the workstations and remote login facilities, is available to everyone in the department.

**SPECIAL ADMISSIONS REQUIREMENTS**

Applicants for admission should have strong preparation in mathematics, engineering, or science. They should be competent in programming but need no computer science beyond that basic level. The GRE General Test is required.

**SPECIAL REQUIREMENTS FOR THE PH.D. DEGREE**

There is no foreign language requirement. To be admitted to candidacy, a student must (1) pass ten courses (including CPSC 690 and CPSC 691) with at least two grades of Honors, the remainder at least High Pass, including three advanced courses in an area of specialization; (2) take six advanced courses in areas of general computer science; (3) successfully complete a research project in CPSC 690, CPSC 691, and submit a written report on it to the faculty; (4) pass a qualifying examination in an area of specialization; (5) be accepted as a thesis student by a regular department faculty member; (6) serve as a teaching assistant for two terms at a TA level; and (7) submit a written dissertation prospectus, with a tentative title for the dissertation. To satisfy the distribution requirement (requirement 2 above), the student must take one course in programming languages or systems, one programming-intensive course, two theory courses, and two in application areas. In order to gain teaching experience, all graduate students are required to serve as teaching assistants for two terms during their first three years of study. All requirements for admission to candidacy must be completed prior to the end of the third year. In addition to all other requirements, students must successfully complete CPSC 991, Ethical Conduct of Research, prior to the end of their first year of study. This requirement must be met prior to registering for a second year of study.

**MASTER’S DEGREES**

M.Phil. See Degree Requirements under Policies and Regulations.

M.S. (en route to the Ph.D.) To qualify for the M.S., the student must pass eight courses at the 500 level or above from an approved list. An average grade of at least High Pass is required, with at least one grade of Honors.
Terminal Master's Degree Program  Students may also be admitted to a terminal master’s degree program directly. The requirements are the same as for the M.S. en route to the Ph.D. This program is normally completed in one year, but a part-time program may be spread over as many as four years.

A brochure providing additional information about the department, faculty, courses, and facilities is available from the Graduate Coordinator, Department of Computer Science, Yale University, PO Box 208285, New Haven CT 06520-8285; e-mail, cs-admissions@cs.yale.edu.

COURSES

CPSC 512a / ECON 562a, Designing the Digital Economy  Eric Weyl
Information technology is transforming how almost every market works: finance has been transformed by algorithmic trading and bitcoin, ride-sharing is changing the nature of public transportation, Amazon is revolutionizing logistics, and Airbnb is now the most valuable accommodation provider in the world. This transformation, which has been led by start-ups and newly dominant technology companies, inherently combines technical and economic aspects, as entrepreneurs take advantage of the potential of technology to facilitate exchanges that were previously infeasible. This crash course in the key tools from economics and computer science that are being used to design digital markets exposes students to a range of concrete and topical practical problems in the area.

CPSC 522a, Operating Systems  Zhong Shao
The design and implementation of operating systems. Topics include synchronization, deadlocks, process management, storage management, file systems, security, protection, and networking.

CPSC 524b, Parallel Programming Techniques  Andrew Sherman
Practical introduction to parallel programming, emphasizing techniques and algorithms suitable for scientific and engineering computations. Aspects of processor and machine architecture. Techniques such as multithreading, message passing, and data parallel computing using graphics processing units. Performance measurement, tuning, and debugging of parallel programs. Parallel file systems and I/O.

CPSC 526a, Building Distributed Systems  Ennan Zhai
Ubiquitous services such as Google, Facebook, and Amazon run on the back of massive distributed systems. This course covers the fundamental principles, abstractions, and mechanisms that inform the design of such systems, as well as the practical details of real-world implementations. Technical topics covered include properties such as consistency, availability, durability, isolation, and failure atomicity; as well as protocols such as RPC, consensus, consistent hashing, and distributed transactions. The final project involves implementing a real-world distributed service.

CPSC 527a, Object-Oriented Programming  James Glenn
Object-oriented programming as a means to efficient, reliable, modular, reusable code. Use of classes, derivation, templates, name-hiding, exceptions, polymorphic functions, and other features of C++.

CPSC 531a, Computer Music: Algorithmic and Heuristic Composition  Scott Petersen
Study of the theoretical and practical fundamentals of computer-generated music. Music and sound representations, acoustics and sound synthesis, scales and tuning systems, algorithmic and heuristic composition, and programming languages for computer music. Theoretical concepts are supplemented with pragmatic issues expressed in a high-level programming language.

CPSC 532b, Computer Music: Sound Representation and Synthesis  Scott Petersen
Study of the theoretical and practical fundamentals of computer-generated music, with a focus on low-level sound representation, acoustics and sound synthesis, scales and tuning systems, and programming languages for computer music generation. Theoretical concepts are supplemented with pragmatic issues expressed in a high-level programming language. Prerequisite: ability to read music.

CPSC 533a, Computer Networks  Yang Yang
An introduction to the design, implementation, analysis, and evaluation of computer networks and their protocols. Topics include layered network architectures, applications, transport, congestion, routing, data link protocols, local area networks, performance analysis, multimedia networking, network security, and network management. Emphasis on protocols used in the Internet.

CPSC 537a, Introduction to Database Systems  Abraham Silberschatz

CPSC 539b, Software Engineering  Ruzica Piskac
Introduction to building a large software system in a team. Learning how to collect requirements and write a specification. Project planning and system design. Increasing software reliability: debugging, automatic test generation. Introduction to type systems, static analysis, and model checking.

CPSC 545a, Introduction to Data Mining  Guy Wolf
A study of algorithms and systems that allow computers to find patterns and regularities in databases, to perform prediction and forecasting, and to improve their performance generally through interaction with data.
CPSC 553a / CB&B 555a / GENE 555a, Machine Learning for Biology  Smita Krishnaswamy
This course introduces biology as a systems and data science through open computational problems in biology, the types of high-throughput data that are being produced by modern biological technologies, and computational approaches that may be used to tackle such problems. We cover applications of machine-learning methods in the analysis of high-throughput biological data, especially focusing on genomic and proteomic data, including denoising data; nonlinear dimensionality reduction for visualization and progression analysis; unsupervised clustering; and information theoretic analysis of gene regulatory and signaling networks. Students’ grades are based on programming assignments, a midterm, a paper presentation, and a final project. Prerequisite: GENE 760 or permission of the instructor.

CPSC 554a, Software Analysis and Verification  Ruzica Piskac
Introduction to concepts, tools, and techniques used in the formal verification of software. State-of-the-art tools used for program verification; detailed insights into algorithms and paradigms on which those tools are based, including model checking, abstract interpretation, decision procedures, and SMT solvers.

CPSC 557b, Sensitive Information in a Wired World  Michael Fischer
Issues of ownership, control, privacy, and accuracy of the huge amount of sensitive information about people and organizations that is collected, stored, and used by today’s ubiquitous information systems. Readings consist of research papers that explore both the power and the limitations of existing privacy-enhancing technologies such as encryption and "trusted platforms."

CPSC 567a, Cryptography and Computer Security  Michael Fischer
A survey of such private and public key cryptographic techniques as DES, RSA, and zero-knowledge proofs, and their application to problems of maintaining privacy and security in computer networks. Focus on technology, with consideration of such societal issues as balancing individual privacy concerns against the needs of law enforcement, vulnerability of societal institutions to electronic attack, export regulations and international competitiveness, and development of secure information systems.

CPSC 568b, Computational Complexity  Dana Angluin
Introduction to the theory of computational complexity. Basic complexity classes, including polynomial time, nondeterministic polynomial time, probabilistic polynomial time, polynomial space, logarithmic space, and nondeterministic logarithmic space. The roles of reductions, completeness, randomness, and interaction in the formal study of computation.

CPSC 570a, Artificial Intelligence  Dragomir Radev
Introduction to artificial intelligence research, focusing on reasoning and perception. Topics include knowledge representation, predicate calculus, temporal reasoning, vision, robotics, planning, and learning.

CPSC 572b, Intelligent Robotics  Brian Scassellati
Introduction to the construction of intelligent, autonomous systems. Sensory-motor coordination and task-based perception. Implementation techniques for behavior selection and arbitration, including behavior-based design, evolutionary design, dynamical systems, and hybrid deliberative-reactive systems. Situated learning and adaptive behavior.

CPSC 574b, Computational Intelligence for Games  James Glenn

CPSC 575a / ENAS 575a, Computational Vision and Biological Perception  Steven Zucker
An overview of computational vision with a biological emphasis. Suitable as an introduction to biological perception for computer science and engineering students, as well as an introduction to computational vision for mathematics, psychology, and physiology students.

CPSC 576b / AMTH 667b / ENAS 576b, Advanced Computational Vision  Steven Zucker
Advanced view of vision from a mathematical, computational, and neurophysiological perspective. Emphasis on differential geometry, machine learning, visual psychophysics, and advanced neurophysiology. Topics include perceptual organization, shading, color, and texture.

CPSC 577b, Natural Language Processing  Dragomir Radev
Linguistic, mathematical, and computational fundamentals of natural language processing (NLP). Topics include part of speech tagging, Hidden Markov models, syntax and parsing, lexical semantics, compositional semantics, machine translation, text classification, discourse, and dialogue processing. Additional topics such as sentiment analysis, text generation, and deep learning for NLP.

CPSC 578a, Computer Graphics  Holly Rushmeier
Introduction to the basic concepts of two- and three-dimensional computer graphics. Topics include affine and projective transformations, clipping and windowing, visual perception, scene modeling and animation, algorithms for visible surface determination, reflection models, illumination algorithms, and color theory.

CPSC 579b, Advanced Topics in Computer Graphics  Julie Dorsey
An in-depth study of advanced algorithms and systems for rendering, modeling, and animation in computer graphics. Topics vary and may include reflectance modeling, global illumination, subdivision surfaces, NURBS, physically based fluids systems, and character animation.

CPSC 610b, Topics in Computer Science and Law  Joan Feigenbaum
This course focuses on socio-technical problems in computing, i.e., problems that cannot be solved through technological progress alone but rather require legal, political, or cultural progress as well. Examples include but are not limited to computer security, intellectual property protection, cyber crime, cyber war, surveillance, and online privacy. The course is addressed to graduate students in Computer
Science who are interested in socio-technical issues but whose undergraduate work may not have addressed them; it is designed to bring these students rapidly to the point at which they can do research on socio-technical problems. Students do term projects (either papers or software artifacts) and present them at the end of the term. In order to ensure that there is enough time for both midterm feedback on project proposals and in-class presentation of the finished projects, enrollment is limited to fifteen. If fewer than fifteen Computer Science graduate students enroll, Yale College undergraduates will be allowed to enroll with permission of the instructor. Prerequisites: the basics of cryptography and computer security (as covered in CPSC 467), networks (as covered in CPSC 433), and databases (as covered in CPSC 437), or permission of the instructor.

**CPSC 640b, Topics in Numerical Computation**  Vladimir Rokhlin
This course discusses several areas of numerical computing that often cause difficulties to non-numericists, from the ever-present issue of condition numbers and ill-posedness to the algorithms of numerical linear algebra to the reliability of numerical software. The course also provides a brief introduction to "fast" algorithms and their interactions with modern hardware environments. The course is addressed to Computer Science graduate students who do not necessarily specialize in numerical computation; it assumes the understanding of calculus and linear algebra and familiarity with (or willingness to learn) either C or FORTRAN. Its purpose is to prepare students for using elementary numerical techniques when and if the need arises.

**CPSC 663b, Machine Learning**  Staff

**CPSC 679a, Computational Issues in 3-D Design and Fabrication**  Benedict Brown
This course focuses on computational methods for designing and fabricating 3-D objects. The course considers the data structures and algorithms for the complete process, from specifying physical source material to the production of a new physical object. The process begins with obtaining the shapes of existing 3-D objects in digital form using active 3-D scanning or photogrammetry. The digital shape is then edited with a variety of local operators and global filters. The shape description is then prepared for input to a numerically controlled machine. Production by various means is considered, including fused deposition modeling (FDM), milling, and laser cutting.

**CPSC 690a, Independent Project I**  Staff
By arrangement with faculty.

**CPSC 691a, Independent Project II**  Staff
By arrangement with faculty.

**CPSC 692a, Independent Project**  Staff
Individual research for students in the M.S. program. Requires a faculty supervisor and the permission of the director of graduate studies.

**CPSC 745b / AMTH 745b / CB&B 745b, Advanced Topics in Machine Learning and Data Mining**  Smita Krishnaswamy and Guy Wolf
An overview of advances in the past decade in machine learning and automatic data-mining approaches for dealing with the broad scope of modern data-analysis challenges, including deep learning, kernel methods, dictionary learning, and bag of words/features. This year, the focus is on a broad scope of biomedical data-analysis tasks, such as single-cell RNA sequencing, single-cell signaling and proteomic analysis, health care assessment, and medical diagnosis and treatment recommendations. The seminar is based on student presentations and discussions of recent prominent publications from leading journals and conferences in the field. Prerequisite: basic concepts in data analysis (e.g., CPSC 545 or 563) or permission of the instructor.

**CPSC 752b / CB&B 752b / MB&B 752b / MCB 752b, Biomedical Data Science: Mining and Modeling**  Mark Gerstein
Biomedical data science encompasses the analysis of gene sequences, macromolecular structures, and functional genomics data on a large scale. It represents a major practical application for modern techniques in data mining and simulation. Specific topics to be covered include sequence alignment, large-scale processing, next-generation sequencing data, comparative genomics, phylogenetics, biological database design, geometric analysis of protein structure, molecular-dynamics simulation, biological networks, normalization of microarray data, mining of functional genomics data sets, and machine-learning approaches to data integration. Prerequisites: biochemistry and calculus, or permission of the instructor.

**CPSC 800a or b, Directed Readings**  Staff
By arrangement with faculty.

**CPSC 990a, Ethical Conduct of Research for Master's Students**  Holly Rushmeier
This course meets on four consecutive Fridays: Sept. 8, 15, 22, and Oct. 6.

**CPSC 991a / MATH 991a, Ethical Conduct of Research**  Vladimir Rokhlin
0 Course cr