Harvard University
Faculty of Arts and Sciences
Office of the Registrar

Computer Science

AN HISTORICAL EDITION OF FAS COURSES OF INSTRUCTION

Faculty of the Division of Engineering and Applied Sciences Offering Instruction in Computer Science

David M. Brooks, Assistant Professor of Computer Science on the Gordon McKay Endowment *(Director of Undergraduate Studies)*
Nir Friedman, Visiting Associate Professor of Computer Science *(Hebrew University of Jerusalem)*
Steven J. Gortler, Robert I. Goldman Professor of Computer Science *(on leave 2004-05)*
Craig Gotsman, Visiting Associate Professor of Computer Science *(Technion, Israel Institute of Technology)*
Barbara J. Grosz, Higgins Professor of Natural Sciences
H. T. Kung, William H. Gates Professor of Computer Science and Electrical Engineering
Henry H. Leitner, Senior Lecturer on Computer Science
Harry R. Lewis, Harvard College Professor and Gordon McKay Professor of Computer Science *(on leave fall term)*
Michael D. Mitzenmacher, John L. Loeb Associate Professor of the Natural Sciences
John G. Morrisett, Allen B. Cutting Professor of Computer Science
Radhika Nagpal, Assistant Professor of Computer Science on the Gordon McKay Endowment
Venkatesh Narayanamurti, John A. and Elizabeth S. Armstrong Professor of Engineering and Applied Sciences, Professor of Physics, Dean of the Division of Engineering and Applied Sciences and Dean for Physical Sciences
Anthony G. Oettinger, Gordon McKay Professor of Applied Mathematics and Professor of Information Resources Policy *(on leave 2004-05)*
David C. Parkes, Assistant Professor of Computer Science on the Gordon McKay Endowment
Avrom J. Pfeffer, Associate Professor of Computer Science on the Gordon McKay Endowment
Michael O. Rabin, Thomas J. Watson, Sr. Professor of Computer Science
Norman Ramsey, Associate Professor of Computer Science on the Gordon McKay Endowment
Mema Roussopoulos, Assistant Professor of Computer Science on the Gordon McKay Endowment
Margo I. Seltzer, Herchel Smith Professor of Computer Science
Stuart M. Shieber, Harvard College Professor and James O. Welch, Jr. and Virginia B. Welch Professor of Computer Science
Michael D. Smith, Gordon McKay Professor of Computer Science and Electrical Engineering
Salil P. Vadhan, Thomas D. Cabot Associate Professor of Computer Science
Leslie G. Valiant, T. Jefferson Coolidge Professor of Computer Science and Applied Mathematics
Gu-Yeon Wei, Assistant Professor of Electrical Engineering on the Gordon McKay Endowment
Matthew D. Welsh, Assistant Professor of Computer Science on the Gordon McKay Endowment
Woodward Yang, Visiting Professor of Electrical Engineering and Computer Science
Todd Zickler, Assistant Professor of Electrical Engineering on the Gordon McKay Endowment

Other Faculty Offering Instruction in Computer Science

Marco Iansiti, David Sarnoff Professor of Business Administration *(Business School)*

The Division of Engineering and Applied Sciences offers undergraduate and graduate courses in Applied Mathematics, Applied Physics, Computer Science, Earth and Planetary Sciences, and Engineering Sciences. Recommended course programs at the undergraduate level may be obtained from the Academic Office, Pierce Hall 110. Division faculty also offer several courses in the section entitled General Education Electives.

Primarily for Undergraduates

For information concerning concentration in Computer Science please consult the Director of Undergraduate Studies or the Academic Office, Division of Engineering and Applied Sciences, Pierce Hall 110. The Applied Mathematics and Engineering Sciences sections of the catalog should be consulted for additional courses relevant to computer science. In addition, attention is called to the following courses in related fields: Quantitative Reasoning 20; Applied Mathematics 106, 107; General Education 156; Linguistics 112a, 112b; Philosophy 144; Physics 123; and Statistics 110, 111, 171.

Computer Science 1. Great Ideas in Computer Science
Catalog Number: 6903
Henry H. Leitner
Half course (spring term). M., W., F., at 11. EXAM GROUP: 4
An introduction to the most important discoveries and intellectual paradigms in computer science, designed for students with little or no previous background. Explores problem-solving using high and low-level programming languages; presents an integrated view of computer systems, from switching circuits up through compilers and GUI design. Examines theoretical and practical limitations related to unsolvable and intractable computational problems, and the social and ethical dilemmas presented by such issues as software unreliability and invasions of privacy.
Note: May not be taken for credit after completing Computer Science 50.

Computer Science 50. Introduction to Computer Science I
Catalog Number: 4949
Michael D. Smith
Half course (fall term). M., W., F., at 10. EXAM GROUP: 3
Introduction to the intellectual enterprises of computer science. Algorithms: their design, specification, and analysis. Software development: problem decomposition, abstraction, data structures, implementation, debugging, testing. Architecture of computers: low-level data representation and instruction processing. Computer systems: programming languages, compilers, operating systems. Computers in the real world: networks, security and cryptography, artifical intelligence, social issues. Laboratory exercises include extensive programming in the C language.
Note: No previous computer experience required. This course, when taken for a letter grade, meets the Core area requirement for Quantitative Reasoning.

Computer Science 51. Introduction to Computer Science II
Catalog Number: 3411
Margo I. Seltzer and Radhika Nagpal

Half course (spring term). Tu., Th., 1–2:30. EXAM GROUP: 15, 16
Abstract models for computational processes and their concrete realizations. Functional, imperative, object-oriented and event-driven styles of programming. The structure, interpretation and compilation of programming languages. State-space search, finite-state processes, formal logic, and syntactic and semantic formalisms as examples of useful abstractions. The engineering of complex software through procedural and data abstractions. Laboratory exercises using LISP, C++, and Java.
Prerequisite: Computer Science 50 or equivalent.

***Computer Science 91r. Supervised Reading and Research**
Catalog Number: 0361
David M. Brooks
Half course (fall term; repeated spring term). Hours to be arranged.
Supervised individual study of advanced topics in computer science. A student wishing to enroll in Computer Science 91r must be accepted by a faculty member who will supervise the course work. A form available from Pierce Hall 110, must be filled out and signed by the student and faculty supervisor. Students writing theses may enroll in this course while conducting thesis research and writing.
Note: At most two terms of Computer Science 91r may be taken for academic credit. May not be taken Pass/Fail. Students wishing more information about the range of suitable projects or faculty supervisors should consult the Director of Undergraduate Studies.

[*Computer Science 96. System Design Projects]
Catalog Number: 7499 Enrollment: Limited to 20.

Half course (fall term). Tu., Th., 2:30–4. EXAM GROUP: 16, 17
Cooperative design, development, and testing of a sizable and realistic computer network system. Students gain experience both in software development and system lifecycle issues, and in the area of application. We concentrate on mathematical modeling for prediction. The target application is prediction of student enrollments based on historical data, as raised by the recent discussions of preregistration. Students work in groups; both student participation in the classroom and effective group cooperation outside the classroom are stressed.
Note: Expected to be given in 2005–06.
Prerequisite: Computer Science 51.

For Undergraduates and Graduates

[Computer Science 120. Introduction to Cryptography]
Catalog Number: 5911
Salil P. Vadhan
Half course (spring term). M., W., 1–2:30. EXAM GROUP: 6, 7
Algorithms to guarantee privacy and authenticity of data during communication and computation. Rigorous proofs of security based on precise definitions and assumptions. Topics may include one-way functions, private-key and public-key encryption, digital signatures, pseudorandom generators, higher-level protocols such as electronic cash, and the role of cryptography in network and systems security.

Note: Expected to be given in 2005–06.
Prerequisite: Computer Science 121 or Computer Science 124.

Computer Science 121. Introduction to Formal Systems and Computation
Catalog Number: 0669
Salil P. Vadhan
Half course (fall term). Tu., Th., 10–11:30. EXAM GROUP: 12, 13
General introduction to formal systems and the theory of computation. Elementary treatment of automata, formal languages, computability, uncomputability, computational complexity, NP–completeness, and mathematical logic.

Computer Science 124. Data Structures and Algorithms
Catalog Number: 5207
Michael D. Mitzenmacher
Half course (spring term). Tu., Th., 11:30–1. EXAM GROUP: 13, 14
Design and analysis of efficient algorithms and data structures. Algorithm design methods, graph algorithms, approximation algorithms, and randomized algorithms are covered.
Prerequisite: Computer Science 51; some exposure to discrete applied mathematics, such as Applied Mathematics 106 or 107 or Computer Science 121 or Statistics 110, is helpful.

Computer Science 127. Computational Geometry
Catalog Number: 7377
Craig Gotsman (Technion, Israel Institute of Technology)
Half course (spring term). Tu., Th., 10–11:30. EXAM GROUP: 12, 13
Basic techniques, data structures, combinatorics, and algorithms for solving geometric problems. Examples are convex hulls, Voronoi diagrams, point set, and polygon triangulation. Range search, linear programming, and point location. Some theoretical and programming exercises.

Computer Science 141. Computing Hardware
Catalog Number: 4357
Woodward Yang
Half course (fall term). M., W., 1–2:30, and a two-hour weekly laboratory. EXAM GROUP: 6, 7
Introduction to the design, structure, and operation of digital computers; logic circuits and digital electronics; computer arithmetic; computer architecture; and machine language programming. Consideration of the design interactions between hardware and software systems.
Prerequisite: Computer Science 50.

Computer Science 143. Computer Networks
Catalog Number: 6401
H. T. Kung
Half course (fall term). Tu., Th., 11:30–1. EXAM GROUP: 13, 14
Principles, design, implementation, and performance of computer networks. Topics include: Internet protocols and routing, local area networks, TCP, performance analysis, congestion control, network address translation, voice and video over IP, switching and routing, mobile IP, peer-to-peer overlay networks, network security, and other current research topics. Programming

assignments on protocol implementation and analysis.
Prerequisite: Computer Science 51.

Computer Science 144r. Networks Design Projects
Catalog Number: 5415
H. T. Kung and Marco Iansiti (Business School)
Half course (spring term). M., W., 2:30–4. EXAM GROUP: 7, 8
Cooperative design and development of advanced network-based systems with both technology and business considerations. Students will work in 2 person teams. Student work will include reading assignments, homework sets, a project proposal, and project reports and presentations. At the end of the class, all teams will defend their approaches and results in front of the class and invited guests.
Note: Preference given to upper-class undergraduates or graduate students in computer science or in business.
Prerequisite: Computer Science 143 or equivalent experience.

Computer Science 146. Computer Architecture
Catalog Number: 6520
David M. Brooks
Half course (spring term). M., W., 1–2:30. EXAM GROUP: 6, 7
Review of fundamental structures in modern microprocessor and computer system architecture design. Topics include computer organization, instruction set design, memory system design, pipelining, and other techniques to exploit parallelism. System level topics include storage subsystems and basics of multiprocessor systems. Emphasis on quantitative evaluation of design alternatives while considering design metrics such as performance and power dissipation.
Prerequisite: Computer Science 141.

Computer Science 148. Design of VLSI Circuits and Systems
Catalog Number: 1772 Enrollment: Limited to 16.
Gu-Yeon Wei
Half course (spring term). Tu., Th., 11:30–1. EXAM GROUP: 13, 14
Presentation of concepts and techniques for the design and fabrication of VLSI systems and digital MOS integrated circuits. Topics include: basic semiconductor theory; MOS transistors and digital MOS circuits design; synchronous machines, clocking, and timing issues; high-level description and modeling of VLSI systems; synthesis and place and route design flows; and testing of VLSI circuits and systems. Various CAD tools for design, simulation, and verification are extensively used.
Prerequisite: Computer Science 141 or permission of instructor.

Computer Science 152. Principles of Programming Languages
Catalog Number: 6841
Norman Ramsey
Half course (spring term). M., W., F., at 11. EXAM GROUP: 4
Intellectual tools needed to design, evaluate, and choose programming languages. Historical influences on language design. Case studies, reinforced by programming exercises. Advanced languages, abstraction mechanisms. Includes functional, object-oriented, and logic paradigms.

Focuses on practice, but covers formal topics crucial for intellectual rigor: abstract syntax, lambda calculus, type systems, and dynamic semantics. Grounding sufficient to read professional literature.
Prerequisite: Computer Science 121. Students must have excellent programming skills, be comfortable with recursion, basic mathematical ideas and notations.

Computer Science 153. Principles of Programming Language Compilation
Catalog Number: 2842
John G. Morrisett
Half course (fall term). M., W., F., at 11. EXAM GROUP: 4
Implementation of efficient interpreters and compilers for programming languages. Associated algorithms and pragmatic issues. Emphasizes practical applications including those outside of programming languages proper. Also shows relationships to programming-language theory and design. Participants build a working compiler including lexical analysis, parsing, type checking, code generation, and register allocation. Exposure to run-time issues and optimization.
Prerequisite: Computer Science 51.

Computer Science 161. Operating Systems
Catalog Number: 4347
Matthew D. Welsh
Half course (spring term). Tu., Th., 1–2:30. EXAM GROUP: 15, 16
The fundamental principles of resource management and abstraction in modern operating systems. Control abstractions: threads, processes, scheduling, synchronization. Storage abstractions: dynamic memory allocation, virtual memory, file system design. Communication abstractions: interprocess communication, networking. Case studies. Design and implementation of parts of a multiuser multitasking virtual-memory operating system.
Prerequisite: Computer Science 51.

Computer Science 164. Internet Technologies
Catalog Number: 7295
Mema Roussopoulos
Half course (fall term). M., W., 2:30–4. EXAM GROUP: 7, 8
Survey of current authoring, distributing, and browsing technologies used in the Internet. Topics include: HTTP, DNS and TCP/IP overview, HTML techniques for text, links, forms, and images, client/server paradigm, server-side programming, CGI scripts, dynamic content with Java, how web browsers and web servers work, web caching and replication.
Prerequisite: Computer Science 50.

Computer Science 165. Information Management
Catalog Number: 0560
Margo I. Seltzer
Half course (fall term). Tu., Th., 11:30–1. EXAM GROUP: 13, 14
Covers the fundamental concepts of database and information management. Data models: relational, object-oriented, and other; implementation techniques of database management systems, such as indexing structures, concurrency control, recovery, and query processing;

management of unstructured data; terabyte-scale databases.
Prerequisite: Computer Science 51.

Computer Science 175. Computer Graphics
Catalog Number: 3771
Craig Gotsman (Technion, Israel Institute of Technology)
Half course (fall term). M., W., 4–5:30. EXAM GROUP: 9
The computational aspects of computer graphics. Two major themes are image rendering (viewing transformations, clipping, visible-surface processing, raster algorithms, reflection models, lighting models, surface shading, antialiasing, ray tracing, radiosity, and volume rendering) and scene modeling (modeling transformations, curves and surfaces, texture mapping, data-amplification techniques, constructive solid geometry, scalar- and vector-field data, and animation). Ancillary topics include color compression, image compression, image compositing, graphical user interfaces, and special machine architectures for computer graphics.
Prerequisite: Computer Science 51, Applied Mathematics 21b or Mathematics 21b.

Computer Science 181. Intelligent Machines: Perception, Learning, and Uncertainty
Catalog Number: 6454
Avrom J. Pfeffer
Half course (spring term). M., W., 2:30–4. EXAM GROUP: 7, 8
Introduction to artificial intelligence, focusing on problems of perception, machine learning and reasoning under uncertainty. Supervised learning algorithms. Neural networks and applications to character recognition. Statistical pattern recognition. Bayesian networks: representation, inference and learning. Hidden Markov models and applications to speech recognition. Markov decision processes and reinforcement learning.
Prerequisite: Computer Science 51 and Computer Science 121. Statistics 110 is recommended.

Computer Science 182. Intelligent Machines: Reasoning, Actions, and Plans
Catalog Number: 0134
David C. Parkes
Half course (fall term). M., W., 2:30–4. EXAM GROUP: 7, 8
Introduction to AI, focused on problems in reasoning about action and rational decision making. Search: constraint satisfaction; informed search and optimization; game playing. Knowledge representation and logical inference. Planning: representation, search and heuristics. Bounded rationality, situated agents. Multiagent systems. Discussion of relevant work in philosophy, economics, and decision theory. Applications to scheduling, robotics and e-commerce.
Prerequisite: Computer Science 51; Computer Science 121 (may be taken concurrently).

[Computer Science 187. Computational Linguistics]
Catalog Number: 0249
Stuart M. Shieber
Half course (spring term). Tu., Th., 10–11:30. EXAM GROUP: 12, 13
Introduction to computational linguistics, the study of human language using the tools and techniques of computer science, with applications to a variety of natural-language-processing problems. Representing syntactic structure: context-free, augmented context-free, and trans-context-free grammars. Representing semantic structure: first-order and higher-order logics.

Computing with syntactic and semantic representations: Prolog programming; parsing and generation algorithms. Low-level language processing with finite-state methods.
Note: Expected to be given in 2005–06.
Prerequisite: Computer Science 121.

Primarily for Graduates

[Computer Science 220r. Cryptography: Trust and Adversity]
Catalog Number: 1637
Michael O. Rabin
Half course (fall term). Tu., Th., 11:30–1. EXAM GROUP: 13, 14
Topics in modern cryptography. Primality testing, finite fields, elliptic curves. Protocols: Public-key encryptions, digital signatures, key exchanges, zero-knowledge proofs, authentication oblivious transfer, secret sharing, proactive security, fair contract signing, distributed agreements. Foundations: Probablistic encryption and semantic security. Attacks and countermeasures: Non-malleability, plaintext awareness and proofs of plaintext knowledge. Absolutely secure encryptions. Prerequisites will be discussed in sections.
Note: Expected to be given in 2005–06.

[Computer Science 221. Computational Complexity]
Catalog Number: 5812
Salil P. Vadhan
Half course (spring term). Tu., Th., 1–2:30. EXAM GROUP: 15, 16
A quantitative theory of the resources needed for computing and the impediments to efficient computation. The models of computation considered include ones that are finite or infinite, deterministic, randomized, quantum or nondeterministic, discrete or algebraic, sequential or parallel.
Note: Expected to be given in 2005–06.
Prerequisite: Computer Science 121 or equivalent.

[Computer Science 222. Algorithms at the Ends of the Wire]
Catalog Number: 2493
Michael D. Mitzenmacher
Half course (fall term). Tu., Th., 1–2:30. EXAM GROUP: 15, 16
Covers topics related to algorithms for big data, especially related to networks. Themes include compression, cryptography, coding, and information retrieval related to the World Wide Web. Requires a major final project.
Note: Expected to be given in 2005–06.
Prerequisite: Computer Science 124.

Computer Science 223. Probabilistic Analysis and Algorithms
Catalog Number: 4740
Michael D. Mitzenmacher
Half course (fall term). Tu., Th., 1–2:30. EXAM GROUP: 15, 16
Probabilistic techniques and tools for the design and analysis of algorithms. Designed for all first-year graduate students in all areas.

Prerequisite: Computer Science 124. Preferably additional probability, such as in Computer Science 226r, Statistics 110, or Mathematics 191.

[Computer Science 224r. Randomness in Computation]
Catalog Number: 3380
Michael O. Rabin
Half course (fall term). Tu., Th., 11:30–1. EXAM GROUP: 13, 14
The surprising efficacy of randomization in the solution of algorithmic and computer science problems. Applications include number theory, cryptography, finite fields, computational geometry, routing, parallel algorithms, pattern matching, distributed systems, self-checking programs, probabilistically checkable proofs.
Note: Expected to be given in 2005–06.

[Computer Science 225. Pseudorandomness]
Catalog Number: 4869
Salil P. Vadhan
Half course (spring term). Tu., Th., 1–2:30. EXAM GROUP: 15, 16
Efficiently generating objects that "look random" despite being constructed using little or no randomness. Connections and applications to computational complexity, cryptography, and combinatorics. Pseudorandom generators, randomness extractors, expander graphs, error-correcting codes, hash functions.
Note: Expected to be given in 2005–06.
Prerequisite: Exposure to randomized algorithms (as in Computer Science 124), computational complexity (as in Computer Science 121), and algebra (as in Applied Mathematics 106, Mathematics 123, or Computer Science 226r).

Computer Science 226r. Efficient Algorithms
Catalog Number: 1749
Michael O. Rabin
Half course (fall term). Tu., Th., 11:30–1. EXAM GROUP: 13, 14
A survey of important computer algorithms for numerical and data manipulation problems and their applications in actual computing situations. Topics include combinatorial algorithms, string matching, wavelet algorithms, FFT and its applications, algebraic computations, randomized algorithms in algebra number theory and geometry, maximal flows, error correcting codes, public key cryptography, protocols for distributed systems, and parallel algorithms.

Computer Science 227. Advanced Computational Geometry
Catalog Number: 1789
Craig Gotsman (Technion, Israel Institute of Technology)
Half course (spring term). Tu., Th., 10–11:30. EXAM GROUP: 12, 13
The contents and course requirements are identical to those of Computer Science 127, with the exception that students enrolled in Computer Science 227 are expected to conduct a modest research project.

Computer Science 228. Computational Learning Theory
Catalog Number: 0364

Leslie G. Valiant
Half course (spring term). Tu., Th., 2:30–4. EXAM GROUP: 16, 17
Possibilities of and limitations to performing learning by computational agents. Topics include computational models, polynomial time learnability, learning from examples and learning from queries to oracles. Computational limitations. Statistical limitations. Applications to Boolean functions, automata and geometric functions. Learning algorithms for models of neural computation.
Prerequisite: Computer Science 121 or equivalent.

Computer Science 229r (formerly Computer Science 229). Topics in the Theory of Computation
Catalog Number: 3730
Salil P. Vadhan
Half course (spring term). Tu., Th., 1–2:30. EXAM GROUP: 15, 16
Students read, present, and critically evaluate current research papers in theoretical computer science. See syllabus and web site for specific topics of focus.
Prerequisite: Computer Science 121 or equivalent.

Computer Science 232. Introduction to Computational Molecular Biology
Catalog Number: 9480
Nir Friedman (Hebrew University of Jerusalem)
Half course (spring term). Tu., Th., 10–11:30. EXAM GROUP: 12, 13
Computational methods in molecular biology research. Emphasis on modeling and algorithmic aspects for computational tasks posed by recent genomic data. Sequence models, alignment, sequence motifs, protein families, phylogenetic inference, gene expression analysis, clustering, regulation networks.
Prerequisite: Computer Science 121, 124, and 181, or approval of the instructor. Background in biology, especially molecular biology of the cell, is useful but not required.

Computer Science 244r. Advanced Networks Design Projects
Catalog Number: 3018
H. T. Kung and Marco Iansiti (Business School)
Half course (spring term). M., W., 2:30–4. EXAM GROUP: 7, 8
The contents and course requirements are similar to those of Computer Science 144r, with the exception that students enrolled in Computer Science 244r are expected to do substantial system implementation and perform graduate-level work.
Note: Preference given to upper-class undergraduates or graduate students in computer science or in business who are proficient in computer programming or in business software.
Prerequisite: Computer Science 143 or equivalent experience.

Computer Science 246r. Advanced Computer Architecture
Catalog Number: 0979
David M. Brooks
Half course (fall term). M., W., 1–2:30. EXAM GROUP: 6, 7
Covers technology trends in computer system design, with an emphasis on power-aware computing for mobile, embedded, and traditional systems. System design areas include

implementation, architecture, system software, and applications.
Note: Taught seminar style after the first several lectures.
Prerequisite: Computer Science 146 or Computer Science 161 or Computer Science 153 recommended. Consult instructor if unsure.

[Computer Science 251. Advanced Systems Programming]
Catalog Number: 5566
Norman Ramsey
Half course (fall term). M., W., F., 1–2:30. EXAM GROUP: 6, 7
Case studies of classic problems in computer systems. Students read, understand, implement, and present each study. Develops deep understanding of programming techniques used in systems research. Emphasizes programming, discussion, and presentation. Cases matched to student interests.
Note: Expected to be given in 2005–06.
Prerequisite: One of the following: Computer Science 143, Computer Science 152, Computer Science 153, Computer Science 161, or Computer Science 175.

[Computer Science 252r. Advanced Topics in Programming Languages]
Catalog Number: 1986
Norman Ramsey
Half course (fall term). W., F., 2:30–4. EXAM GROUP: 7, 8
Advanced functional programming. Lazy evaluation, monads. Folds and unfolds. Combinators for parsing and prettyprinting. Modules systems. Type systems: polymorphism and overloading, type and constructor classes, higher-order kinds, higher-rank polymorphism, polytypic programming. Implementation: heap profiling.
Note: Expected to be given in 2005–06.
Prerequisite: Computer Science 152 or permission of the instructor.

Computer Science 253r. Advanced Topics in Programming Language Compilation
Catalog Number: 2901 Enrollment: Limited to 18. Preference given to graduate students or upper-class concentrators.
Michael D. Smith
Half course (spring term). Tu., Th., 10–11:30. EXAM GROUP: 12, 13
In-depth introduction to computer optimization. Topics include scalar optimization, register allocation, instruction scheduling, dependence analysis, interprocedural analysis, and cache optimization.
Prerequisite: Computer Science 153 or equivalent.

[Computer Science 255. Topics in Language-Based Security]
Catalog Number: 6216
John G. Morrisett
Half course (spring term). M., W., 2:30–4. EXAM GROUP: 7, 8
Reviews research in programming language-based security mechanisms. Topics include compiler and run-time techniques for enforcing policies; type and proof systems for expressing policies; and static analyses for establishing policies.

Note: Expected to be given in 2005–06. Taught in seminar style.
Prerequisite: Computer Science 152, 153, 252r, 253r, or permission of the instructor.

Computer Science 256. Programming Language and Semantics
Catalog Number: 1554
John G. Morrisett
Half course (spring term). M., W., 2:30–4. EXAM GROUP: 7, 8
An overview of operational, denotational, and axiomatic semantics; type systems, program analysis, and program equivalence.
Prerequisite: Computer Science 152.

[Computer Science 261. Research Topics in Operating Systems]
Catalog Number: 6706
Margo I. Seltzer
Half course (spring term). Tu., Th., 1–2:30. EXAM GROUP: 15, 16
A quantitative approach to operating system design and evaluation. Discussion of recent research including extensible operating system architectures, distributed systems, and performance analysis. Overview of research techniques and methodology.
Note: Expected to be given in 2005–06.
Prerequisite: Computer Science 161, or equivalent.

[Computer Science 262. Introduction to Distributed Computing]
Catalog Number: 7949

Half course (spring term). M., W., 4–5:30. EXAM GROUP: 9
Examination of the special problems associated with distributive computing (e.g., partial failure and lack of global knowledge) and protocols that function in the face of these problems. Emphasis on causal ordering, event and RPC-based systems.
Note: Expected to be given in 2005–06.
Prerequisite: Computer Science 161 or permission of instructor.

Computer Science 263. Wireless Sensor Networks
Catalog Number: 6846
Matthew D. Welsh
Half course (fall term). Tu., Th., 2:30–4. EXAM GROUP: 16, 17
Recent advances in wireless communications and sensor networks. Wireless networking, routing, standards including 802.11, Bluetooth, and 802.15.4. Embedded OS, programming tools, applications, and security. Students read research papers and undertake a research project.
Prerequisite: Computer Science 161 or Computer Science 143.

Computer Science 264. Peer-to-Peer Systems
Catalog Number: 6069 Enrollment: Limited to 24.
Mema Roussopoulos
Half course (spring term). Tu., Th., 11:30–1. EXAM GROUP: 13, 14
Discusses research papers on peer-to-peer systems. Topics include: routing, search, caching, security, reputation and trust, incentives, and applications. Students undertake a major research

project and lead discussions of readings.
Note: Preference to graduate students or upper-level concentrators.
Prerequisite: Computer Science 161 or Computer Science 143.

[Computer Science 265. Database Systems]
Catalog Number: 4104
Margo I. Seltzer
Half course (fall term). Tu., Th., 1–2:30. EXAM GROUP: 15, 16
A research-oriented introduction to Database Management systems. First third covers database design, implementation, and use. Topics include: network, relational, and object oriented database models, system architectures, transaction processing, system implementation, and SQL. Remaining two-thirds address research literature surrounding database systems, including an historical perspective, the emergence of relational and object-oriented systems, concurrency control, and distributed systems. Students will be expected to undertake a final research project.
Note: Expected to be given in 2005–06.
Prerequisite: Computer Science 51.

Computer Science 266. Biologically-Inspired Distributed and Multi-Agent Systems
Catalog Number: 0766 Enrollment: Limited to 20.
Radhika Nagpal
Half course (fall term). Tu., Th., 10–11:30. EXAM GROUP: 12, 13
Surveys biologically-inspired approaches to designing distributed systems. Focus is on algorithms, analysis, and programming paradigms. Topics: swarm intelligence, amorphous computing, immune-inspired systems, synthetic biology. Discussion of research papers and a research project required.
Note: Geared toward graduate students of all levels as well as advanced undergraduates. Preference given to graduate students or upper-level concentrators.
Prerequisite: Computer Science 161 or Computer Science 143 required.

[Computer Science 277. Geometric Modeling in Computer Graphics]
Catalog Number: 3067
Steven J. Gortler
Half course (spring term). M., W., 2:30–4. EXAM GROUP: 7, 8
Advanced seminar in computer graphics focusing on geometric representations and processing. Topics include: direct manipulation, implicit surfaces, spline presentations, recursively subdivided surfaces, model simplification, surface parameterization and processing, mesh generation, and motion capture processing.
Note: Expected to be given in 2005–06.
Prerequisite: Computer Science 175.

Computer Science 278. Rendering and Image Processing in Computer Graphics
Catalog Number: 4883
Craig Gotsman (Technion, Israel Institute of Technology)
Half course (spring term). Tu., Th., 2:30–4. EXAM GROUP: 16, 17
Advanced course in computer graphics focusing on image rendering and processing. Topics include: light transport, efficient rendering, image based rendering, texture processing,

interactive image processing.
Prerequisite: Computer Science 175 or permission of instructor.

[Computer Science 281r. Artificial Intelligence: Reasoning and Planning Systems]
Catalog Number: 0707
Avrom J. Pfeffer
Half course (fall term). M., W., 1–2:30. EXAM GROUP: 6, 7
In-depth introduction to formalisms for knowledge representation and techniques for reasoning and planning. Topics: formal logic-based representations; probabilistic reasoning; nonmonotonic logics; truth-maintenance systems; qualitative reasoning; inheritance hierarchies; computational approaches to reasoning about actions and time, including actions of multiple agents, nonlinear planning, plan recognition; reasoning about knowledge, belief, and action.
Note: Expected to be given in 2005–06.
Prerequisite: Computer Science 182 or permission of instructor.

[Computer Science 282. Probabilistic Reasoning]
Catalog Number: 3158
Avrom J. Pfeffer
Half course (fall term). M., W., 1–2:30. EXAM GROUP: 6, 7
In-depth study of principles and techniques for probabilistic reasoning and decision-theoretic planning. Topics include: Bayesian networks and Markov networks; exact and approximate probabilistic inference algorithms; learning Bayesian networks from data; temporal probability models; integrating logic and probability; influence diagrams; Markov decision processes; reinforcement learning.
Note: Expected to be given in 2005–06.
Prerequisite: Computer Science 181 or permission of instructor.

Computer Science 283. Computer Vision
Catalog Number: 4475
Roger W. Brockett and Todd Zickler
Half course (fall term). M., W., F., at 10. EXAM GROUP: 3
Vision as an ill-posed inverse problem: two-dimensional signal processing; image enhancement and restoration; feature analysis; image segmentation and analysis; structure from motion, texture, and shading; binocular stereo; pattern classification; and applications.

Computer Science 285. Multi-agent Planning Systems
Catalog Number: 1060
Barbara J. Grosz
Half course (spring term). Tu., Th., 1–2:30. EXAM GROUP: 15, 16
Theories and techniques for multi-agent planning, including formal models of rational agents, collaborative plans, and social systems; computational approaches to distributed planning and problem solving, negotiation, and decision theory for planning; collaborative systems design.
Prerequisite: Computer Science 181 or 182, or permission of instructor.

Computer Science 286r. Topics at the Interface between Computer Science and Economics
Catalog Number: 1099 Enrollment: Limited to 20. Preference given to graduate students or

upper-class concentrators.
David C. Parkes
Half course (spring term). M., W., 1–2:30. EXAM GROUP: 6, 7
Interplay between computation and incentives within open decentralized computational systems. Mechanisms and market design, negotiation, social-choice, information-economics and privacy. Readings from theoretical CS, AI, operations research, and economics. Seminar style. Spring 2005: Computational Mechanism Design.
Prerequisite: Mathematics 21b, Applied Mathematics 21b, or equivalent; Computer Science 121, 124, and 181 or 182, or equivalents; or permission of instructor.

Computer Science 287r. Natural Language Processing
Catalog Number: 3306
Stuart M. Shieber
Half course (fall term). Tu., Th., 2:30–4. EXAM GROUP: 16, 17
In-depth investigation of natural-language-processing techniques. Topics include: finite-state, context-free, and trans-context-free formalisms, syntactic analysis, semantic interpretation, weighted automata and transducers. Students discuss research papers and undertake a significant research project.
Prerequisite: Computer Science 187 or permission of instructor.

[Computer Science 288. Computational Models of Discourse]
Catalog Number: 1392
Barbara J. Grosz
Half course (spring term). Tu., Th., 2:30–4. EXAM GROUP: 16, 17
Computational theories of discourse (text and dialogue) structure and processing. Topics include: anaphora, focusing, plans and speech acts, plan recognition algorithms, models of collaborative planning, intonation. Discussion of dialogue and text understanding systems. Application to the design of human-computer interface systems.
Note: Expected to be given in 2005–06.
Prerequisite: Computer Science 182 or 287r or equivalent, or permission of instructor.

Computer Science 299r. Special Topics in Computer Science
Catalog Number: 4592
Margo I. Seltzer
Half course (fall term; repeated spring term). Hours to be arranged.
Supervision of experimental or theoretical research on acceptable computer science problems and supervision of reading on topics not covered by regular courses of instruction.
Note: Open to graduate students and AB/SM candidates only. Students must arrange such work with a member of the Division. This course is graded and is ordinarily taken with the approval of the Committee on Higher Degrees. Applicants must file a project sheet before study cards are filed. Project sheets may be obtained from the Academic Office, Pierce Hall 110.

Graduate Courses of Reading and Research

Reading courses are odd-numbered; research courses are even-numbered.
***Computer Science 305,306. Information Resources: Technology and Policy**

Catalog Number: 6364,3478
Anthony G. Oettinger 2403 (on leave 2004-05)

***Computer Science 307,308. Biologically-Inspired Multi-Agent Systems, Distributed Systems, and Computational Biology**
Catalog Number: 8289,8308
Radhika Nagpal 5068

***Computer Science 309,310. Computational Mechanism Design, Electronic Marketplaces, and Multi-Agent Systems**
Catalog Number: 8764,0931
David C. Parkes 4202

***Computer Science 311,312. Collaborative Systems, AI Planning, and Natural Language Processing**
Catalog Number: 4677,6223
Barbara J. Grosz 1599

***Computer Science 317,318. Distributed Systems, Networking, and Mobile Computing**
Catalog Number: 9388,7137
Mema Roussopoulos 4822

***Computer Science 319,320. Distributed Systems, Operating Systems, and Networks**
Catalog Number: 8038,8568
Matthew D. Welsh 4600

***Computer Science 321,322. Databases, Operating System, and Software Design**
Catalog Number: 4085,4086
Margo I. Seltzer 3371

***Computer Science 323,324. Human-Computer Communication through Natural, Graphical, and Artificial Languages**
Catalog Number: 2450,2453
Stuart M. Shieber 2456

***Computer Science 325,326. Programming Languages and Tools**
Catalog Number: 8055,0747
Norman Ramsey 2831

***Computer Science 327,328. Mathematical Logic, Theory of Computation**
Catalog Number: 1160,3576
Harry R. Lewis 4455 (on leave fall term)

***Computer Science 343,344. Computer Architecture: Modeling and Design**
Catalog Number: 3932,9266
David M. Brooks 4222

***Computer Science 345,346. High-Performance Computer Systems**
Catalog Number: 6154,6156
Michael D. Smith 3372

***Computer Science 347,348. Computer Vision**
Catalog Number: 1882,8831
Todd Zickler 5143

***Computer Science 351,352. Complexity of Computations: Concurrent Programming and Synchronization**
Catalog Number: 0218,0255
Michael O. Rabin 7003

***Computer Science 353,354. Representation and Reasoning, Machine Learning and Decision Making**
Catalog Number: 6816,1843
Avrom J. Pfeffer 2830

***Computer Science 355,356. Computational Complexity, Parallel Computation, Computational Learning, Neural Computation**
Catalog Number: 0345,0346
Leslie G. Valiant 7396

***Computer Science 357,358. Computational Complexity, Cryptography, and Pseudorandomness**
Catalog Number: 3485,8641
Salil P. Vadhan 3833

***Computer Science 359,360. Online Algorithms and Randomized Algorithms**
Catalog Number: 2104,1477
Michael D. Mitzenmacher 7748

***Computer Science 361,362. Programming Languages and Semantics**
Catalog Number: 8672,8366
John G. Morrisett 4853

***Computer Science 375,376. Computer Graphics**
Catalog Number: 6832,7313
Steven J. Gortler 2824 (on leave 2004-05)