broad interdisciplinary field of African and African American Studies, with a focus on the Humanities (Literature, Art, Music, and Religion).
*Note:* Required for all graduates in African and African American Studies in their first year.

# Computer Science

## AN HISTORICAL EDITION OF FAS COURSES OF INSTRUCTION

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

Ryan Prescott Adams, Assistant Professor of Computer Science
Daniel A. Armendariz, Preceptor in Computer Science
David M. Brooks, Haley Family Professor of Computer Science
Cristopher R. Cecka, Lecturer on Computational Science
Yiling Chen, John L. Loeb Associate Professor of the Natural Sciences
Stephen N. Chong, Associate Professor of Computer Science
David Cox, Assistant Professor of Molecular and Cellular Biology and of Computer Science
Finale Doshi-Velez, Assistant Professor of Computer Science
Krzysztof Z. Gajos, Associate Professor of Computer Science
Steven J. Gortler, Robert I. Goldman Professor of Computer Science
Barbara J. Grosz, Higgins Professor of Natural Sciences
Thouis Ray Jones, Lecturer on Computational Science
Verena S. Kaynig-fittkau, Lecturer on Computational Science
Edward W. Kohler, Associate Professor of Computer Science, Microsoft Professor of Computer Science
H. T. Kung, William H. Gates Professor of Computer Science and Electrical Engineering
Henry H. Leitner, Senior Lecturer on Computer Science
Harry R. Lewis, Interim Dean of the School of Engineering and Applied Sciences, and Gordon McKay Professor of Computer Science
Alexander Lex, Lecturer on Computer Science
David J. Malan, Gordon McKay Professor of the Practice of Computer Science
Michael D. Mitzenmacher, Thomas J. Watson, Sr. Professor of Computer Science
John G. Morrisett, Allen B. Cutting Professor of Computer Science
Cherry Murray, John A. and Elizabeth S. Armstrong Professor of Engineering and Applied Sciences, and Professor of Physics *(on leave spring term)*
Radhika Nagpal, Fred Kavli Professor of Computer Science
Jelani Nelson, Assistant Professor of Computer Science



Yaacov Nissim Kobliner, Visiting Associate Professor in Computer Science
David C. Parkes, George F. Colony Professor of Computer Science, Harvard College Professor, Area Dean for Computer Science
Hanspeter Pfister, An Wang Professor of Computer Science *(on leave 2014-15)*
Margo I. Seltzer, Herchel Smith Professor of Computer Science
Or Sheffet, Lecturer on Computer Science
Stuart M. Shieber, James O. Welch, Jr. and Virginia B. Welch Professor of Computer Science
Yaron Singer, Assistant Professor of Computer Science
Michael D. Smith, Edgerley Family Dean of the Faculty of Arts and Sciences, and John H. Finley, Jr. Professor of Engineering and Applied Sciences
Salil P. Vadhan, Vicky Joseph Professor of Computer Science and Applied Mathematics
Leslie G. Valiant, T. Jefferson Coolidge Professor of Computer Science and Applied Mathematics
James H. Waldo, Gordon McKay Professor of the Practice of Computer Science
Stratos Idreos, Assistant Professor of Computer Science
Todd Zickler, William and Ami Kuan Danoff Professor of Electrical Engineering and Computer Science
Jonathan L. Zittrain, Professor of Computer Science

The School of Engineering and Applied Sciences *(www.seas.harvard.edu)* offers undergraduate and graduate courses in Applied Computation, Applied Mathematics, Applied Physics, Computer Science, and Engineering Sciences. Engineering and Applied Sciences faculty also offer several courses in the section entitled Freshman Seminars, Extra-Departmental Courses, and House Seminars.

***Primarily for Undergraduates***

For information concerning concentration in Computer Science please consult the Director of Undergraduate Studies or the Office of Student Affairs, School 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.

**Computer Science 1. Great Ideas in Computer Science**
Catalog Number: 6903
*Henry H. Leitner*
*Half course (spring term). Tu., Th., 10–11:30. EXAM GROUP: 12*
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. This course, when taken for a letter grade, meets the General Education requirement for Empirical and Mathematical Reasoning.



**[*Computer Science 2. Digital Platforms]**
Catalog Number: 31335 Enrollment: Limited to 5.
*Instructor to be determined*
*Half course (spring term). Th., 5–7 p.m.*
The Internet operates in layers, and so does much of the technology that hooks up to it: PCs, mobile phones, tablets. Nearly two decades ago those platforms were conceptually simple: a "generative" base offered by one manufacturer, on which any third party could build. (Think: Windows and the programs that run on it.) Some efforts by platform makers to tip the scales in their favor in the layer above resulted in extended controversy and regulatory efforts, such as over Windows coming bundled with Internet Explorer. Today platforms are just as vital but far more complex. We have hybrids like the iOS and Android operating systems or the Facebook and Twitter platforms, where the platform makers offer their systems as services rather than products, influencing and sometimes outright limiting connection between users and independent developers for those platforms. How should we think about these new platforms? What counts as a "level playing field," and what responsibility, if any, is there for public authorities to enforce it? What lessons, if any, do the prior tangles offer for today?
*Note:* This course is jointly-offered with the Kennedy School as DPI-668 and with the Law School as 2601.

**Computer Science 20. Discrete Mathematics for Computer Science**
Catalog Number: 22235
*Harry R. Lewis*
*Half course (spring term). M., W., F., at 10. EXAM GROUP: 5*
Widely applicable mathematical tools for computer science, including topics from logic, set theory, combinatorics, number theory, probability theory, and graph theory. Practice in reasoning formally and proving theorems.
*Note:* Covers material used in Computer Science 121 and Computer Science 124. This course, when taken for a letter grade, meets the General Education requirement for Empirical and Mathematical Reasoning.

**Computer Science 50 (SAT/UNS). Introduction to Computer Science I**
Catalog Number: 43861
*David J. Malan*
*Half course (fall term). M., W., at 1-2:30, and a weekly section. EXAM GROUP: 1*
Introduction to the intellectual enterprises of computer science and the art of programming. This course teaches students how to think algorithmically and solve problems efficiently. Topics include abstraction, algorithms, data structures, encapsulation, resource management, security, software engineering, and web development. Languages include C, PHP, and JavaScript plus SQL, CSS, and HTML. Problem sets inspired by real-world domains of biology, cryptography, finance, forensics, and gaming. Designed for concentrators and non-concentrators alike, with or without prior programming experience.
*Note:* Undergraduates, GSAS students, and cross-registered students may take CS50 either Satisfactory/Unsatisfactory (SAT/UNS) or for a letter grade. To take CS50 SAT/UNS, register for catalog number 43861. To take CS50 for a letter grade, register for catalog number 4949. When taken for a letter grade, this course meets the General Education requirement for undergraduates for Empirical and Mathematical Reasoning. See course's website for FAQs. This

course will also meet on Fri 9/5 and Fri 10/17. Students with conflicts may watch those lectures online.

**Computer Science 50 (Letter Grade). Introduction to Computer Science I**
Catalog Number: 4949
*David J. Malan*
*Half course (fall term). M., W., 1-2:30, and a weekly section. EXAM GROUP: 1*
Introduction to the intellectual enterprises of computer science and the art of programming. This course teaches students how to think algorithmically and solve problems efficiently. Topics include abstraction, algorithms, data structures, encapsulation, resource management, security, software engineering, and web development. Languages include C, PHP, and JavaScript plus SQL, CSS, and HTML. Problem sets inspired by real-world domains of biology, cryptography, finance, forensics, and gaming. Designed for concentrators and non-concentrators alike, with or without prior programming experience.
*Note:* Undergraduates, GSAS students, and cross-registered students may take CS50 either Satisfactory/Unsatisfactory (SAT/UNS) or for a letter grade. To take CS50 SAT/UNS, register for catalog number 43861. To take CS50 for a letter grade, register for catalog number 4949. When taken for a letter grade, this course meets the General Education requirement for undergraduates for Empirical and Mathematical Reasoning. See course's website for FAQs. This course will also meet on Fri 9/5 and Fri 10/17. Students with conflicts may watch those lectures online.

**Computer Science 51. Introduction to Computer Science II**
Catalog Number: 3411
*John G. Morrisett*
*Half course (spring term). Tu., Th., 1-2:30, and an additional 90-minute section to be arranged. EXAM GROUP: 1*
Abstraction and design in computation. Topics include functional and object-oriented styles of programming, software engineering in the small, and models of computation. Our main goal is to understand how to design large programs to make them readable, maintainable, elegant, and efficient. Exercises in OCaml.
*Prerequisite:* Computer Science 50 or equivalent.

**Computer Science 61. Systems Programming and Machine Organization**
Catalog Number: 3461
*Edward W. Kohler*
*Half course (fall term). Tu., Th., 2:30–4. EXAM GROUP: 14*
Fundamentals of computer systems programming, machine organization, and performance tuning. This course provides a solid background in systems programming and a deep understanding of low-level machine organization and design. Topics include C and assembly language programming, program optimization, memory hierarchy and caching, virtual memory and dynamic memory management, concurrency, threads, and synchronization.
*Prerequisite:* CS50 or some experience programming in C.

***Computer Science 90na (formerly *Computer Science 42). The Internet: Governance and Power**

Catalog Number: 37293 Enrollment: Limited
*Jack L. Goldsmith (Law School) and Bruce Schneier (Law School)*
*Half course (spring term). Tu., 5–7 p.m. EXAM GROUP: 6*
This seminar will examine the individuals and institutions that control the Internet, and how the Internet affects the distribution and operation of power, broadly conceived. We will examine technologies of control (such as surveillance, censorship, propaganda, and use control) and of evading control, the individuals and institutions that seek to regulate the Internet (such as governments, the IETF, and hackers), the relationship between cybersecurity, national security, and Internet governance, the economics of Internet communications, and more.
*Note:* Offered jointly with the Kennedy School and the Law School. This course does not count for concentration credit in Computer Science.
*Prerequisite:* Seminar is by permission of instructor. Please send a statement of interest and resume to qashat@law.harvard.edu. Please include your year and program information as well. The application deadline for HLS/FAS/SEAS students is October 31. The application deadline for HKS students is November 7.

***Computer Science 91r. Supervised Reading and Research**
Catalog Number: 0361
*Steven J. Gortler (spring term) and Harry R. Lewis (fall term)*
*Half course (fall term; repeated spring term). Hours to be arranged. EXAM GROUP: Fall: 1; Spring: 8*
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 the Student Affairs Office, 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.
*Instructor to be determined*
*Half course (spring term). M., Th., 4–6.*
Cooperative design, development, and testing of a sizable and realistic computer system. Students work as a group with a client on a real-world open-ended problem, and gain experience in problem definition, software development, and system lifecycle issues, and in the area of application. Students work in groups; both student participation in the classroom and effective group cooperation outside the classroom are stressed.
*Prerequisite:* Computer Science 51 or 61.

***For Undergraduates and Graduates***

***Computer Science 105. Privacy and Technology**
Catalog Number: 9751 Enrollment: Limited to 30.
*James H. Waldo*



*Half course (fall term). Tu., Th., 1–2:30. EXAM GROUP: 8*
What is privacy, and how is it affected by recent developments in technology? This course critically examines popular concepts of privacy and uses a rigorous analysis of technologies to understand the policy and ethical issues at play. Case studies: database anonymity, research ethics, wiretapping, surveillance, and others. Course relies on some technical material, but is open and accessible to all students, especially those with interest in economics, engineering, political science, computer science, sociology, biology, law, government, philosophy.
*Note:* This course, when taken for a letter grade, meets the General Education requirement for Culture and Belief.

**Computer Science 109. Data Science**
Catalog Number: 70866
*Rafael A. Irizarry (Public Health) and Verena S. Kaynig-Fittkau*
*Half course (fall term). Tu., Th., 2:30–4, and a weekly section. EXAM GROUP: 14*
Learning from data in order to gain useful predictions and insights. This course introduces methods for five key facets of an investigation: data wrangling, cleaning, and sampling to get a suitable data set; data management to be able to access big data quickly and reliably; exploratory data analysis to generate hypotheses and intuition; prediction based on statistical methods such as regression and classification; and communication of results through visualization, stories, and interpretable summaries. Built around three modules: prediction and elections, recommendation and business analytics, and sampling and social network analysis.
*Note:* Only one of CS 109, AC 209, or Stat 121 can be taken for credit. Only admitted graduate students can take AC 209, in which case we expect significant differences in readings, assignments, and projects.
*Prerequisite:* Programming knowledge at the level of CS 50 or above, and statistics knowledge at the level of Stat 100 or above (Stat 110 recommended).

**Computer Science 121. Introduction to the Theory of Computation**
Catalog Number: 0669
*Harry R. Lewis*
*Half course (fall term). Tu., Th., 10–11:30. EXAM GROUP: 12*
General introduction to the theory of computation, teaching how to reason precisely about computation and prove mathematical theorems about its capabilities and limitations. Finite automata, Turing machines, formal languages, computability, uncomputability, computational complexity, and the P vs. NP question.
*Note:* Students may not receive credit for both CS 121 and CS 125.
*Prerequisite:* experience in formal mathematics at the level of Computer Science 20.

**Computer Science 124. Data Structures and Algorithms**
Catalog Number: 5207
*Jelani Nelson*
*Half course (spring term). Tu., Th., 11:30–1. EXAM GROUP: 15*
Design and analysis of efficient algorithms and data structures. Algorithm design methods, graph algorithms, approximation algorithms, and randomized algorithms are covered.
*Note:* Starting in the spring of 2013, Computer Science 124 will assume background from Computer Science 20. Students will not receive credit for both CS 124 and CS 125.

*Prerequisite:* Computer Science 50 or equivalent; Computer Science 51 is helpful. Some exposure to discrete applied mathematics, such as Applied Mathematics 106 or 107 or Computer Science 121 or Statistics 110, is also helpful.

**Computer Science 125. Algorithms and Complexity - *(New Course)***
Catalog Number: 35335
*Michael D. Mitzenmacher and Salil P. Vadhan*
*Half course (fall term). Tu., Th., 10–11:30. EXAM GROUP: 12*
An accelerated introduction to theoretical computer science for students with strong mathematical preparation, to be taken in place of both Computer Science 121 and 124. Algorithm design methods, including graph algorithms, approximation algorithms, and randomized algorithms. Models of computation, computability theory, and computational complexity, including the P vs. NP question.
*Note:* Students may not receive credit for both CS 125 and either CS 121 or CS 124.
*Prerequisite:* Comfort with reading and writing mathematical proofs, at the level of Math 25 or 55 (which may be taken concurrently).

**[Computer Science 127. Introduction to Cryptography]**
Catalog Number: 23635
*Instructor to be determined*
*Half course (fall term). Tu., Th., 10–11:30.*
Algorithms to guarantee privacy and authenticity of data during communication and computation. 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, zero-knowledge proofs, fully homomorphic encryption, and the role of cryptography in network and systems security.
*Prerequisite:* Computer Science 121 or Computer Science 124.

**Computer Science 141. Computing Hardware**
Catalog Number: 4357
*David M. Brooks*
*Half course (fall term). M., W., 1-2:30, and a two-hour weekly laboratory. EXAM GROUP: 1*
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:* Programming experience required.

**Computer Science 143. Computer Networks**
Catalog Number: 6401
*H. T. Kung*
*Half course (fall term). M., W., 1–2:30. EXAM GROUP: 1*
Networking has enabled the emergence of mobile and cloud computing, creating the most important technological paradigm shift in computing of the past decade. Further advancements in networking are expected to similarly transform the technological landscape over the next decade through the emergence of the Internet of Things and gigabit wireless networks. In order to play a role in this era of new network-powered advancements, students must have a thorough

understanding of emerging networking topics. Rather than teaching the basic networking protocols, which have become very mature and can be treated as a black box, in CS 143, we will teach the new issues and topics of interest which will power important emerging applications. This focus on upcoming applications is the motivation for CS 143 this semester. The class will be organized into the following nine modules: Basic Networking Concepts: Protocol Layering; Internet of Things: All-service Bluetooth Low Energy (BLE); Data Center Networking: Software Defined Networking; Web-scale Networking: Distributed Cloud Computing and Virtual Machine Migration; Content Networks: Video Streaming; Network Security: Defense Against Protocol Exploitation; Wireless Networking: Wireless Mesh, Geographic Routing; Machine Learning Assisted Networking: End-to-end Application Adaptive Protocols; Cyber-physical Networks: Vehicular Networking. Students will have the opportunity to implement the concepts learned in the course through programming assignments, read and discuss the latest networking literature, and design and implement a final project.
*Prerequisite:* Strong interest in the subject matter and programming experience (CS50 should be fine). Lab sessions will be provided to give extra support.

**[Computer Science 144r. Networks Design Projects]**
Catalog Number: 5415
*Instructor to be determined*
*Half course (spring term). M., W., 2:30–4.*
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: 99684
*David M. Brooks*
*Half course (spring term). M., W., 1–2:30. EXAM GROUP: 8*
Review of the fundamental structures in modern processor design. Topics include computer organization, memory system design, pipelining, and other techniques to exploit parallelism. Emphasis on a quantitative evaluation of design alternatives and an understanding of timing issues.
*Prerequisite:* Computer Science 141.

**[*Computer Science 148. Design of VLSI Circuits and Systems]**
Catalog Number: 1772 Enrollment: Limited to 16.
*Instructor to be determined*
*Half course (spring term). Tu., Th., 11:30–1.*
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. Programming Languages**
Catalog Number: 6841
*Stephen N. Chong*
*Half course (spring term). Tu., Th., 10–11:30. EXAM GROUP: 11*
Comprehensive introduction to the principal features and overall design of both traditional and modern programming languages, including syntax, formal semantics, abstraction mechanisms, modularity, type systems, naming, polymorphism, closures, continuations, and concurrency. Provides the intellectual tools needed to design, evaluate, choose, and use programming languages.
*Prerequisite:* Computer Science 51; Computer Science 121 is recommended. Students must have good programming skills, be comfortable with recursion, basic mathematical ideas and notations.

**[Computer Science 153. Compilers]**
Catalog Number: 2842
*Instructor to be determined*
*Half course (fall term). M., W., F., at 11.*
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 or 61.

**Computer Science 161. Operating Systems**
Catalog Number: 4347
*Margo I. Seltzer*
*Half course (spring term). Tu., Th., 1–2:30. EXAM GROUP: 1*
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 and 61.

**[Computer Science 164. Software Engineering]**
Catalog Number: 7295
*Instructor to be determined*
*Half course (spring term). M., W., 2:30–4:30.*
Introduction to principles of software engineering and best practices, including code reviews, source control, and unit tests. Topics include Ajax, database schemas, event handling, HTTP, MVC, object-oriented design, and user experience. Projects include web apps with front-end UIs

(mobile and desktop) and back-end APIs. Languages include JavaScript and PHP.
*Note:* Students will work on projects in groups. Enrollment may be limited. In 2014, this course will focus on the design and implementation of web apps.
*Prerequisite:* Any four CS courses numbered 50 or higher (or, with instructor's permission, multiple years of programming experience).

**[Computer Science 165. Data Systems]**
Catalog Number: 0560
*Instructor to be determined*
*Half course (spring term). M., W., 1–2:30.*
We are in the big data era and data systems sit in the critical path of everything we do, i.e., in businesses, in sciences, as well as in everyday life. This course will be a comprehensive introduction to modern data systems. The primary focus of the course will be on modern trends that are shaping the data management industry right now such as column-store and hybrid systems, shared nothing architectures, cache conscious algorithms, hardware/software codesign, main memory systems, adaptive indexing, stream processing, scientific data management, and key value stores. We will also study the history of data systems and traditional and seminal concepts and ideas such as the relational model, row-store database systems, optimization, indexing, concurrency control, recovery and SQL in order to understand both how data systems have evolved over the years and why, as well as how these concepts apply today and how data systems might evolve in the future.
*Prerequisite:* Computer Science 51 and Computer Science 61.

**Computer Science 171. Visualization**
Catalog Number: 8877
*Alexander Lex*
*Half course (spring term). Tu., Th., 2:30–4, and a weekly section to be arranged. EXAM GROUP: 11*
An introduction to key design principles and techniques for visualizing data. Covers design practices, data and image models, visual perception, interaction principles, visualization tools, and applications. Introduces programming of web-based interactive visualizations.
*Note:* This course, when taken for a letter grade, meets the General Education requirement for Empirical and Mathematical Reasoning.
*Prerequisite:* Students are expected to have basic programming experience (e.g., Computer Science 50).

**Computer Science 175. Computer Graphics**
Catalog Number: 3771
*Steven J. Gortler*
*Half course (fall term). M., W., 2:30–4. EXAM GROUP: 7*
This course covers the fundamentals of 3D computer graphics using a modern shader-based version of OpenGL. Main topics include: geometric coordinate systems and transformations, keyframe animation and interpolation, camera simulation, triangle rasterization, material simulation, texture mapping, image sampling and color theory. The course also touches on ray tracing, geometric modeling and simulation-based animation.
*Prerequisite:* Computer Science 51, Applied Mathematics 21b or Mathematics 21b.



***Computer Science 179. Design of Usable Interactive Systems**
Catalog Number: 4052 Enrollment: Limited to 48.
*Krzysztof Z. Gajos*
*Half course (spring term). M., W., 11:30–1. EXAM GROUP: 14*
Usability and design as keys to successful technology. Covers user observation techniques, needs assessment, low and high fidelity prototyping, usability testing methods, as well as theory of human perception and performance, and design best practices. Focuses on understanding and applying the lessons of human interaction to the design of usable systems; will also look at lessons to be learned from less usable systems. The course includes several small and one large project.

**Computer Science 181. Machine Learning**
Catalog Number: 6454
*Ryan Prescott Adams*
*Half course (spring term). M., W., 1–2:30. EXAM GROUP: 8*
Introduction to machine learning, providing a probabilistic view on artificial intelligence and reasoning under uncertainty. Topics include: supervised learning, ensemble methods and boosting, neural networks, support vector machines, kernel methods, clustering and unsupervised learning, maximum likelihood, graphical models, hidden Markov models, inference methods, and computational learning theory. Students should feel comfortable with multivariate calculus, linear algebra, probability theory, and complexity theory. Students will be required to produce non-trivial programs in Python.
*Prerequisite:* Computer Science 51, Computer Science 121, Statistics 110, Math 21a and 21b (or equivalent).

**Computer Science 182. Intelligent Machines: Reasoning, Actions, and Plans**
Catalog Number: 0134
*Barbara J. Grosz*
*Half course (fall term). Tu., Th., 1–2:30. EXAM GROUP: 8*
Introduction to AI focused on problems in reasoning about action and rational decision making, covering search, knowledge representation and planning. Search: heuristics, informed search and optimization; constraint satisfaction; game playing. Knowledge representation: logics, efficient logical inference, reasoning about categories. Planning: action representations and planning algorithms, hierarchical task networks, sequential decision making. Applications to multi-agent systems, robotics and natural-language processing. Discussion of relevant work in philosophy, economics, and decision theory.
*Prerequisite:* Computer Science 51; Computer Science 121 (may be taken concurrently).

**Computer Science 186. Economics and Computation**
Catalog Number: 87282
*David C. Parkes*
*Half course (spring term). M., W., 11:30–1. EXAM GROUP: 14*
The interplay between economic thinking and computational thinking as it relates to electronic commerce, social networks, collective intelligence and networked systems. Topics covered include: game theory, peer production, reputation and recommender systems, prediction markets, crowd sourcing, network influence and dynamics, auctions and mechanisms, privacy and

security, matching and allocation problems, computational social choice and behavioral game theory. Emphasis will be given to core methodologies, with students engaged in theoretical, computational and empirical exercises.
*Prerequisite:* Applied Math 21b, Computer Science 51 or equivalent, Statistics 110, and one of Computer Science 181, Computer Science 182, Economics 1011a, Economics 1052, or Economics 1056.

**Computer Science 187. Computational Linguistics**
Catalog Number: 0249
*Stuart M. Shieber*
*Half course (fall term). M., W., 2:30–4. EXAM GROUP: 7*
Watson is the world Jeopardy champion. Siri responds accurately to "Should I bring an umbrella tomorrow?". How do they work? This course provides an introduction to the field of 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 such as those deployed in Watson and Siri, and covers pertinent ideas from linguistics, logic programming, and statistical modeling. The course will include an experimental practicum component covering skills in technical writing and editing that should be of general use as well.
*Prerequisite:* Computer Science 51 and Computer Science 121 or permission of the instructor.

***Computer Science 189r. Autonomous Multi-Robot Systems**
Catalog Number: 36932 Enrollment: Limited to 15.
*Radhika Nagpal*
*Half course (fall term). F., 1–4. EXAM GROUP: 1*
Building autonomous robotic systems requires understanding how to make robots that observe, reason, and act. Each component uses many engineering principles: how to fuse, multiple, noisy sensors; how to balance short-term versus long-term goals; how to control one's actions and how to coordinate with others. This year, we will study these questions in the context of a project to develop autonomous robot soccer teams. The class format will mix seminar and lab formats.
*Note:* Preference will be given to students with experience in AI (e.g. CS181 or CS182 and/or robotics ES159).

***Primarily for Graduates***

**Computer Science 205. Computing Foundations for Computational Science**
Catalog Number: 85368
*H. T. Kung*
*Half course (spring term). M., W., 2:30-4 pm, and a weekly section to be arranged. EXAM GROUP: 18*
An applications course highlighting the use of computers in solving scientific problems. Students will be exposed to fundamental computer science concepts such as computer architectures, data structures, algorithms, and parallel computing. Fundamentals of scientific computing including abstract thinking, algorithmic development, and assessment of computational approaches. Students will learn to use open source tools and libraries and apply them to data analysis, modeling, and visualization of real scientific problems. Emphasizes parallel programming and "parallel thinking."

*Prerequisite:* Students are expected to have basic programming experience (e.g., Computer Science 50).

**Computer Science 207. Systems Development for Computational Science**
Catalog Number: 33846
*Cristopher R. Cecka and Ray Jones*
*Half course (fall term). M., W., F., at 11. EXAM GROUP: 18*
This is a project-based course emphasizing designing, building, testing, maintaining and modifying software for scientific computing. Students will work in groups on a number of projects, ranging from small data-transformation utilities to large-scale systems. Students will learn to use a variety of tools and languages, as well as various techniques for organizing teams. Most important, students will learn to fit tools and approaches to the problem being solved.
*Prerequisite:* Students are expected to have basic programming experience (Computer Science 50).

**[Computer Science 221. Computational Complexity]**
Catalog Number: 5812
*Instructor to be determined*
*Half course (spring term). Tu., Th., 10–11:30.*
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.
*Prerequisite:* Computer Science 121 or equivalent.

**[Computer Science 222. Algorithms at the Ends of the Wire]**
Catalog Number: 2493
*Instructor to be determined*
*Half course (fall term). Tu., Th., 11:30–1.*
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.
*Prerequisite:* Computer Science 124.

**Computer Science 223. Probabilistic Analysis and Algorithms**
Catalog Number: 4740
*Michael D. Mitzenmacher*
*Half course (spring term). Tu., Th., 11:30–1. EXAM GROUP: 15*
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 224 (formerly Computer Science 226). Advanced Algorithms - *(New Course)***
Catalog Number: 32918



*Jelani Nelson*
*Half course (fall term). Tu., Th., 2:30–4. EXAM GROUP: 14*
Advanced algorithm design, including but not limited to amortization, randomization, online algorithms, graph algorithms, approximation algorithms, linear programming, and data structures.
*Prerequisite:* CS 124 and probability.

**Computer Science 225. Pseudorandomness**
Catalog Number: 4869
*Salil P. Vadhan*
*Half course (spring term). Tu., Th., 10–11:30. EXAM GROUP: 12*
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.
*Prerequisite:* Exposure to randomized algorithms (as in Computer Science 124 or 125), computational complexity (as in Computer Science 121 or 125), and algebra (as in Applied Mathematics 106 or Mathematics 123).

**Computer Science 227r (formerly Computer Science 220r). Topics in Cryptography and Privacy**
Catalog Number: 1637
*Kobbi Nissim and Or Sheffet*
*Half course (fall term). Tu., Th., 11:30–1. EXAM GROUP: 15*
Topics in cryptography and data privacy drawn from the theoretical computer science research literature. Focus for 2014-15: Differential Privacy -- a mathematical framework for privacy-preserving analysis of datasets, which enables aggregate computations while preventing the leakage of individual-level information.
*Prerequisite:* Computer Science 124, 125, or 127.

**[Computer Science 228. Computational Learning Theory]**
Catalog Number: 0364
*Instructor to be determined*
*Half course (spring term). Tu., Th., 2:30–4.*
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. Applications to Boolean functions, automata and geometric functions.
*Prerequisite:* Computer Science 121 or equivalent.

**Computer Science 229r. Topics in the Theory of Computation: Biology and Complexity**
Catalog Number: 3730
*Leslie G. Valiant*
*Half course (spring term). Tu., Th., 2:30–4. EXAM GROUP: 11*
Biology abounds with step by step processes, whether in evolution, neural activity, development, or protein circuits. In many of these neither the actual steps taken nor the outcomes are well understood. Computer science is the study of step by step processes and offers an approach to



understanding them as they occur in biology. Students will read, present, and critically evaluate research papers in this area.
*Prerequisite:* Computer Science 121 or equivalent.

**[Computer Science 244r. Networks Design Projects]**
Catalog Number: 3018
*Instructor to be determined*
*Half course (spring term). M., W., 2:30–4.*
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 246. Advanced Computer Architecture**
Catalog Number: 0979
*David M. Brooks*
*Half course (spring term). M., W., 1–2:30. EXAM GROUP: 8*
The contents and course requirements are similar to those of Computer Science 146, with the exception that students enrolled in Computer Science 246 are expected to undertake a substantial course project.
*Prerequisite:* Computer Science 141.

**[Computer Science 247r. Advanced Topics in Computer Architecture ]**
Catalog Number: 48162
*Instructor to be determined*
*Half course (fall term). M., W., F., 10–11:30.*
Seminar course exploring recent research in computer architecture. Topics vary from year to year and will include subjects such as multi-core architectures, energy-efficient computing, reliable computing, and the interactions of these issues with system software. Students read and present research papers, undertake a research project.
*Prerequisite:* Computer Science 146 or 246 or permission of the instructor.

**[*Computer Science 248. Advanced Design of VLSI Circuits and Systems]**
Catalog Number: 7191 Enrollment: Limited to 16.
*Instructor to be determined*
*Half course (spring term). Tu., Th., 11:30–1.*
The contents and course requirements are similar to those of Computer Science 148, with the exception that students enrolled in Computer Science 248 are expected to do a substantial design project and paper discussions on advanced topics.
*Prerequisite:* Computer Science 141 or permission of instructor.

**Computer Science 250. Software Foundations - *(New Course)***
Catalog Number: 82359
*John G. Morrisett*

*Half course (fall term). Tu., Th., 10–11:30. EXAM GROUP: 12*
This course introduces concepts and techniques in the foundational study of programming languages, as well as their formal logical underpinnings. The central theme is the view of programs and languages as formal mathematical objects about which precise claims may be made and proved. Particular topics include operational techniques for formal definition of language features, type systems, and program logics. The models and proofs are formalized using mechanical theorem provers.
*Prerequisite:* Computer Science 51

**Computer Science 252r. Advanced Topics in Programming Languages**
Catalog Number: 1986
*Stephen N. Chong*
*Half course (fall term). Tu., Th., 2:30–4. EXAM GROUP: 14*
Seminar course exploring recent research in programming languages. Topics vary from year to year. Students read and present research papers, undertake a research project.
*Prerequisite:* Computer Science 152 or permission of the instructor.

***Computer Science 260r. Projects and Close Readings in Software Systems**
Catalog Number: 49684
*Edward W. Kohler*
*Half course (spring term). M., W., 1–2:30. EXAM GROUP: 8*
Modern software systems construction and analysis. Distributed systems; operating systems; networks; data centers; big data; emerging systems deployments. Close, careful reading of research papers and code, coupled with programming projects. Readability and programmability. Topic focus will change each offering. May be repeated for credit with instructor permission.
*Prerequisite:* Programming experience and instructor permission.

**Computer Science 261. Research Topics in Operating Systems**
Catalog Number: 6706
*Margo I. Seltzer*
*Half course (fall term). Tu., Th., 1–2:30. EXAM GROUP: 8*
An introduction to operating systems research. Paper-based seminar course that introduces students to the state of the art in systems research through historical and quantitative lenses. Students will read and discuss research papers and complete a final research project.
*Prerequisite:* Computer Science 161, or equivalent.

**[Computer Science 262. Introduction to Distributed Computing]**
Catalog Number: 7949
*Instructor to be determined*
*Half course (spring term). Tu., Th., 2:30–4.*
An examination of the special problems associated with distributed computing such as partial failure, lack of global knowledge, asynchrony and coordination of time, and protocols that function in the face of these problems. Emphasis on both the theory that grounds thinking about these systems and in the ways to design and build such systems.
*Prerequisite:* Computer Science 161 or permission of instructor.

**Computer Science 265 (formerly Database Systems). Big Data Systems**
Catalog Number: 2083
*Stratos Idreos*
*Half course (fall term). W., F., 1–2:30. EXAM GROUP: 1*
Big data is everywhere. A fundamental goal across numerous modern businesses and sciences is to be able to exploit as many machines as possible, to consume as much information as possible and as fast as possible. The big challenge is "how to turn data into useful knowledge". This is far from a simple task and a moving target as both the underlying hardware and our ability to collect data evolve. In this class, we will discuss how to design data systems and algorithms that can "scale up" and "scale out". Scale up refers to the ability to use a single machine to all its potential, i.e., to exploit properly the memory hierarchy and the multiple CPU and GPU cores. Scale out refers to the ability to use more than 1 machines (typically 100s or 1000s) effectively. This is a research oriented class. Every week we will read two modern research papers; one from the scale up area and one from the scale out area. We will use examples from several areas, including relational systems and distributed databases, graph processing systems (i.e., for social networks), key value stores, noSQL and newSQL systems as well as mobile computing. Each student will work on a semester long data systems research project (in groups of 2-4 students) which can be in any of the above areas and will be based on an open research problem.
*Prerequisite:* CS 165 or permission of instructor.

**Computer Science 277. Geometric Modeling in Computer Graphics**
Catalog Number: 3067
*Steven J. Gortler*
*Half course (spring term). Tu., Th., 1–2:30. EXAM GROUP: 1*
Advanced seminar in computer graphics focusing on geometric representations and processing. Topics include: subdivision surfaces, surface parametrization, vector fields over surfaces, shape editing, shape matching and surface reconstruction.
*Prerequisite:* Computer Science 175.

**[Computer Science 278. Rendering and Image Processing in Computer Graphics]**
Catalog Number: 4883
*Instructor to be determined*
*Half course (spring term). Tu., Th., 1–2:30.*
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 279. Research Topics in Human-Computer Interaction**
Catalog Number: 1435
*Krzysztof Z. Gajos*
*Half course (fall term). Tu., Th., 10–11:30. EXAM GROUP: 12*
The course covers major areas of inquiry and core research methods in Human-Computer Interaction including experimental design, statistical data analysis, and qualitative methods. Activities will include discussion of primary literature, a small number of lectures, assignments (design, execution and analysis of both lab-based and on-line experiments), and a research



project. Special focus this year is on social computing and crowd-powered systems. Specifically, we will look at the design and analysis of systems, in which crowds of intrinsically motivated volunteers contribute to meaningful and non-trivial human computation tasks as a byproduct of doing something that they are motivated to do anyway.
*Note:* Designed for first year grads from all areas. Advanced undergraduates welcome, particularly those who wish to do research (or write a thesis) in an area related to Human-Computer Interaction.
*Prerequisite:* None for graduate students; for undergrads CS 179 or CS 171 is strongly recommended and permission of the instructor is required. Basic web hacking is required to implement and deploy web-based experiments.

**Computer Science 280r. Advanced Topics in Artificial Intelligence**
Catalog Number: 11199
*Barbara J. Grosz*
*Half course (spring term). Tu., Th., 1–2:30. EXAM GROUP: 1*
Seminar course exploring research directions in artificial intelligence (AI), typically combining two or more of such areas as multi-agent systems, natural-language processing, machine learning, reasoning under uncertainty, representation systems. Topic for Spring 2015: Multi-agent systems teamwork and plan management.
*Prerequisite:* Computer Science 181 or 182, or equivalent; or permission of instructor.

**[Computer Science 281. Advanced Machine Learning]**
Catalog Number: 97848 Enrollment: Limited to 60.
*Instructor to be determined*
*Half course (fall term). M., W., 1–2:30.*
Advanced statistical machine learning and probabilistic data analysis. Topics include: Markov chain Monte Carlo, variational inference, Bayesian nonparametrics, text topic modeling, unsupervised learning, dimensionality reduction and visualization. Requires a major final project.
*Prerequisite:* Students should feel comfortable with basic linear algebra and probability theory. Students will be expected to implement algorithms in a programming language such as Matlab, Python or R.

**Computer Science 282r. Decision-Making Under Uncertainty - *(New Course)***
Catalog Number: 90117
*Finale Doshi-Velez*
*Half course (spring term). Tu., Th., 10–11:30. EXAM GROUP: 12*
The focus of the Spring 2015 course will be reinforcement learning, a framework for solving problems involving a sequence of decisions with uncertain outcomes. This course will cover the fundamental theory through readings of classic papers and build practical intuition through coding assignments. Topics will include Markov decision process and partially observable Markov decision processes, planning under uncertainty, model-free and model-based reinforcement learning, function approximation in reinforcement learning, and batch reinforcement learning.
*Prerequisite:* Students should be familiar with basic linear algebra, probability, and algorithms;

courses such as Stat 110, AM 21b, and CS 124 may be helpful. Students will be expected to implement algorithms in programming languages such as Matlab, Python, and Java (e.g. CS 51).

**Computer Science 283. Computer Vision**
Catalog Number: 4475
*Todd Zickler*
*Half course (fall term). Tu., Th., 11:30–1. EXAM GROUP: 15*
Vision as an ill-posed inverse problem: image formation, two-dimensional signal processing; feature analysis; image segmentation; color, texture, and shading; multiple-view geometry; object and scene recognition; and applications.
*Note:* Offered jointly with the Design School as SCI-06275.

**Computer Science 284r. Topics on Computation in Networks and Crowds**
Catalog Number: 74473
*Yaron Singer*
*Half course (fall term). M., W., 10–11:30. EXAM GROUP: 5*
Topics on the design and analysis of algorithms, processes, and systems related to crowds and social networks. Readings in AI, theoretical CS, machine learning, social science theory, economic theory, and operations research.
*Prerequisite:* Mathematics 21b, Applied Mathematics 21b, or equivalent; Computer Science 124, and 181 or 182, or equivalents; or permission of instructor.

**[Computer Science 285. Multi-Agent Systems]**
Catalog Number: 1060
*Instructor to be determined*
*Half course (fall term). M., W., 11:30–1.*
Algorithmic, game-theoretic and logical foundations of multi-agent systems, including distributed optimization and problem solving, non-cooperative game theory, learning and teaching, communication, social choice, mechanism design, auctions, negotiation, coalitional game theory, logics of knowledge and belief, collaborative plans and social systems.
*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.
*Instructor to be determined*
*Half course (spring term). F., 9–12.*
Interplay between computation and economics. Topics in electronic commerce, computational social choice, computational mechanism design, peer production, prediction markets and reputation systems. Readings in AI, theoretical CS, multi-agent systems, economic theory, and operations research.
*Prerequisite:* Mathematics 21b, Applied Mathematics 21b, or equivalent; Economics 1011a, or equivalent; or permission of instructor.

**Computer Science 287r. Topics in Computational Linguistics and Natural Language Processing**

Catalog Number: 3306
*Stuart M. Shieber*
*Half course (spring term). M., F., 3–5. EXAM GROUP: 17*
In-depth investigation of topics in computational linguistics and natural-language processing. Students discuss research papers and undertake a significant research project. This term, the course will focus on synchronous grammars and their use for formal modeling of the semantics of natural language, including background on Montague grammar, pertinent logic, lambda calculus, applications to machine translation and other language-processing problems.
*Note:* The course is being offered jointly with the linguistics program as Linguistics 287, with computationally-oriented final projects. Students may not take both Computer Science 287r and Linguistics 287 for credit.
*Prerequisite:* Computer Science 187 or Linguistics 116 or permission of instructor.

**[Computer Science 288r. Advanced Topics in Computer Vision]**
Catalog Number: 62802
*Instructor to be determined*
*Half course (spring term). Tu., Th., 11:30–1.*
Seminar course exploring recent research in computer vision. Topics vary from year to year, typically including object recognition; activity recognition; and visual surveillance. Students read and present research papers and undertake a research project.
*Prerequisite:* Applied Math 21b or equivalent; Statistics 110 or equivalent; or permission of the instructor.

***Computer Science 289 (formerly *Computer Science 266). Biologically-inspired Multi-agent Systems**
Catalog Number: 0766 Enrollment: Limited to 20.
*Radhika Nagpal*
*Half course (spring term). Tu., Th., 11:30–1. EXAM GROUP: 15*
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:* Experience with algorithms (e.g. Computer Science 124) and programming (e.g. Computer Science 51).

**Computer Science 299r. Special Topics in Computer Science**
Catalog Number: 4592
*David C. Parkes*
*Half course (fall term; repeated spring term). Hours to be arranged. EXAM GROUP: 10*
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 School of Engineering and Applied Sciences. 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 303,304. Statistical Machine Learning**
Catalog Number: 46531,61638
*Ryan Prescott Adams 3022*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 17; Spring: 13*

***Computer Science 305,306. Readable, Extensible, High-Performance Software Systems**
Catalog Number: 15739,17423
*Edward W. Kohler 1996*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 7; Spring: 18*

***Computer Science 307,308. Biologically-Inspired Multi-Agent Systems, Distributed Systems, and Computational Biology**
Catalog Number: 8289,8308
*Radhika Nagpal 5068*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 18; Spring: 14*

***Computer Science 309,310. Computational Mechanism Design, Electronic Marketplaces, and Multi-Agent Systems**
Catalog Number: 8764,0931
*David C. Parkes 4202*
*Half course (fall term; repeated spring term). . EXAM GROUP: 5*

***Computer Science 311,312. Collaborative Systems, AI Planning, and Natural Language Processing**
Catalog Number: 4677,6223
*Barbara J. Grosz 1599*
*Half course (fall term; repeated spring term). . EXAM GROUP: 5*

***Computer Science 313,314. Visual Computing**
Catalog Number: 4273,1628
*Hanspeter Pfister 5882 (on leave 2014-15)*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 14; Spring: 11*

***Computer Science 315,316. Social Computing: Computation and Economics**
Catalog Number: 2892,2433
*Yiling Chen 6187*
*Half course (fall term; repeated spring term). . EXAM GROUP: 4*



***Computer Science 319,320. Data Systems Design - *(New Course)***
Catalog Number: 13059,92828
*Stratos Idreos 7489*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 14; Spring: 11*

***Computer Science 321,322. Databases, Operating System, and Software Design**
Catalog Number: 4085,4086
*Margo I. Seltzer 3371*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 1; Spring: 8*

***Computer Science 323,324. Human-Computer Communication through Natural, Graphical, and Artificial Languages**
Catalog Number: 2450,2453
*Stuart M. Shieber 2456*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 9; Spring: 2*

***Computer Science 325,326. Intelligent Interactive Systems and Human-Computer**
Catalog Number: 15849,82478
*Krzysztof Z. Gajos 6339*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 9; Spring: 2*

***Computer Science 327,328. Mathematical Logic, Theory of Computation**
Catalog Number: 1160,3576
*Harry R. Lewis 4455*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 14; Spring: 11*

***Computer Science 343,344. Computer Architecture: Modeling and Design**
Catalog Number: 3932,9266
*David M. Brooks 4222*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 16; Spring: 6*

***Computer Science 345,346. High-Performance Computer Systems**
Catalog Number: 6154,6156
*Michael D. Smith 3372*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 9; Spring: 2*

***Computer Science 347,348. Computer Vision**
Catalog Number: 1882,8831
*Todd Zickler 5143*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 16; Spring: 6*

***Computer Science 351,352. Cryptography: Unbreakable Codes and Financial Cryptography**
Catalog Number: 0218,0255
*Michael O. Rabin 7003*
*Half course (fall term; repeated spring term). . EXAM GROUP: 10*



***Computer Science 355,356. Computational Complexity, Parallel Computation, Computational Learning, Neural Computation**
Catalog Number: 0345,0346
*Leslie G. Valiant 7396*
*Half course (fall term; repeated spring term). . EXAM GROUP: 15*

***Computer Science 357,358. Computational Complexity, Cryptography, and Pseudorandomness**
Catalog Number: 3485,8641
*Salil P. Vadhan 3833*
*Half course (fall term; repeated spring term). . EXAM GROUP: 4*

***Computer Science 359,360. On-line Algorithms and Randomized Algorithms**
Catalog Number: 2104,1477
*Michael D. Mitzenmacher 7748*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 1; Spring: 8*

***Computer Science 361,362. Programming Languages and Semantics**
Catalog Number: 8672,8366
*John G. Morrisett 4853*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 11; Spring: 7*

***Computer Science 363,364. Programming Languages and Security**
Catalog Number: 52264,67371
*Stephen N. Chong 6340*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 6; Spring: 17*

***Computer Science 365. SEAS Teaching Practicum**
Catalog Number: 8195
*John A. Girash 6894*
*Half course (spring term). Tu., 2:30–4:30. EXAM GROUP: 16*
Gain effective skills for teaching applied sciences. Topics: presentation and communication, grading and giving feedback on assignments, cognition and learning, classroom practice and student interactions. Seminar style with an emphasis on observation, practice, feedback, discussion, and reflection.

***Computer Science 375,376. Computer Graphics**
Catalog Number: 6832,7313
*Steven J. Gortler 2824*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 8; Spring: 1*

***Computer Science 377,378. Sketching Algorithms for Massive Data**
Catalog Number: 11014,73819
*Jelani Nelson 7260*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 7; Spring: 18*

***Computer Science 379,380. Algorithms for Social Data**
Catalog Number: 37287,20798
*Yaron Singer 7269*
*Half course (fall term; repeated spring term). . EXAM GROUP: Fall: 17; Spring: 13*

***Cross-listed Courses***

**Linguistics 287. Topics in Computational Linguistics and Natural Language Processing - *(New Course)***
**MCB 131. Computational Neuroscience**
**Statistics 221. Statistical Computing and Learning**
***Statistics 385 (formerly *Statistics 285r). Statistical Machine Learning**

# Dramatic Arts

## AN HISTORICAL EDITION OF FAS COURSES OF INSTRUCTION

***Faculty of the Committee on Dramatics***

Martin Puchner, Byron and Anita Wien Professor of Drama and of English and Comparative Literature *(Chair)*
Daniel Albright, Ernest Bernbaum Professor of Literature
Robin M. Bernstein, Professor of African and African American Studies and of Women, Gender and Sexuality *(on leave 2014-15)*
Sara L. Brown, Lecturer on Dramatic Arts
Julie A. Buckler, Professor of Slavic Languages and Literatures and of Comparative Literature, Harvard College Professor, and Director of the Humanities Program at the Radcliffe Institute for Advanced Studies
Glenda R. Carpio, Professor of English and of African and African American Studies
Deborah D. Foster, Senior Lecturer on Folklore and Mythology
Marjorie Garber, William R. Kenan, Jr. Professor of English and of Visual and Environmental Studies
Sylvaine Guyot, Roy G. Clouse Associate Professor of Romance Languages and Literatures *(on leave 2014-15)*
Jill Johnson, Senior Lecturer on Music and Director of Dance
Daria Khitrova, Assistant Professor of Slavic Languages and Literatures
Ju Yon Kim, Assistant Professor of English
David Marcel Levine, Visiting Professor of Dramatic Arts
Ryan Scott McKittrick, Lecturer on Dramatics Arts and Director of Artistic Programs/Dramaturg