



Candidates for the doctoral degree in Comparative Literature may pursue advanced studies under the individual supervision of these instructors.

Note: Permission of the instructor and the Chairman of the Department required.

Cross-listed Courses

***African and African American Studies 301. Graduate Proseminar**

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
David M. Brooks, Gordon McKay Professor of Computer Science (*on leave spring term*)
Cristopher R. Cecka, Lecturer on Computational Science
Yiling Chen, Associate Professor of Computer Science (*on leave fall term*)
Stephen N. Chong, Associate Professor of Computer Science
David Cox, Assistant Professor of Molecular and Cellular Biology and of Computer Science
Krzysztof Z. Gajos, Associate Professor of Computer Science
John Andrew Girash, Lecturer on Engineering and Applied Sciences
Steven J. Gortler, Robert I. Goldman Professor of Computer Science
Barbara J. Grosz, Higgins Professor of Natural Sciences
Nicole Immorlica, Visiting Assistant Professor in Computer Science
Edward W. Kohler, Associate Professor of Computer Science
H. T. Kung, William H. Gates Professor of Computer Science and Electrical Engineering (*on leave fall term*)
Henry H. Leitner, Senior Lecturer on Computer Science
Harry R. Lewis, Gordon McKay Professor of Computer Science (*Director of Undergraduate Studies in Computer Science*)
David J. Malan, Senior Lecturer on Computer Science
Michael D. Mitzenmacher, Thomas J. Watson, Sr. Professor of Computer Science (*on leave 2013-14*)
John G. Morrisett, Allen B. Cutting Professor of Computer Science (*on leave spring term*)
Cherry Murray, John A. and Elizabeth S. Armstrong Professor of Engineering and Applied Sciences, Professor of Physics, and Dean of the School of Engineering and Applied Sciences



Radhika Nagpal, Fred Kavli Professor of Computer Science
Jelani Nelson, Assistant Professor of 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
Margo I. Seltzer, Herchel Smith Professor of Computer Science
Stuart M. Shieber, James O. Welch, Jr. and Virginia B. Welch Professor of Computer Science
(on leave 2013-14)
Yaron Singer, Assistant Professor of Computer Science
Michael D. Smith, John H. Finley, Jr. Professor of Engineering and Applied Sciences, and Dean
of the Faculty of Arts and Sciences
Jesse Adam Tov, Lecturer on Computer Science
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
Gu-Yeon Wei, Gordon McKay Professor of Electrical Engineering and Computer Science
Woodward Yang, Gordon McKay Professor of Electrical Engineering and 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

Other Faculty Offering Instruction in Computer Science

Joseph K. Blitzstein, Professor of the Practice in Statistics
Lawrence Lessig, Roy L. Furman Professor of Law (*Law School*)
Michael O. Rabin, Thomas J. Watson, Sr. Professor of Computer Science, Emeritus

The School of Engineering and Applied Sciences (www.seas.harvard.edu) offers undergraduate and graduate courses in Applied Mathematics, Applied Physics, Computer Science, Earth and Planetary Sciences, 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, 13



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 or the Core area requirement for Quantitative Reasoning.

***Computer Science 2. Digital Platforms - (New Course)**

Catalog Number: 31335 Enrollment: Limited to 5.

Lawrence Lessig (Law) and Jonathan L. Zittrain

Half course (spring term). Th., 5–7 p.m. EXAM GROUP: 18

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: 3

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. Students planning to take one of these computer science theory courses in the academic year 2013-2014 should consider taking Computer Science 20 in the spring of 2013. This course, when taken for a letter grade, meets the General Education requirement for Empirical and Mathematical Reasoning or the Core area requirement for Quantitative Reasoning.

[*Computer Science 42. Controlling Cyberspace]

Catalog Number: 37293 Enrollment: Limited to 10. Course application due December 9, 2011.

Instructor to be determined

Half course (spring term). M., 5:30–7:30 p.m. EXAM GROUP: 9



Why does the Internet environment exist in the form it does today? What does its future, and the future of online life in general, look like? To what extent is this future malleable? Governments, corporate intermediaries, and hackers are empowered to different degrees by the space, and their interests and strengths are often in tension. This class uses academic as well as non-traditional texts to engender a broader understanding of Internet culture and technology, with an eye towards mapping informed choices about the future.

Note: Permission of instructor required after submission of an [application form](#). Offered jointly with the Law School as LAW 2433.

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: 6, 7

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 9/6 and Fri 10/18. 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: 6, 7

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 9/6 and Fri 10/18. Students with conflicts may watch those lectures online.



Computer Science 51. Introduction to Computer Science II

Catalog Number: 3411

Henry H. Leitner and Jesse Adam Tov

Half course (spring term). Tu., Th., 1-2:30, and an additional 90-minute section to be arranged.

EXAM GROUP: 15, 16

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: 16, 17

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 91r. Supervised Reading and Research**

Catalog Number: 0361

Harry R. Lewis

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 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: 16*

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 or the Core area requirement Social Analysis.

Computer Science 109. Data Science - (New Course)

Catalog Number: 70866

*Hanspeter Pfister (Computer Science) and Joseph K. Blitzstein (Statistics)**Half course (fall term). Tu., Th., 2:30–4, and a weekly section.*

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, 13*

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.

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: 13, 14*



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.

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 127. Introduction to Cryptography - (New Course)

Catalog Number: 23635

Salil P. Vadhan

Half course (fall term). Tu., Th., 10–11:30. EXAM GROUP: 12, 13

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

Instructor to be determined

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: Programming experience required.

[Computer Science 143. Computer Networks]

Catalog Number: 6401

Instructor to be determined

Half course (fall term). M., W., 1–2:30. EXAM GROUP: 6, 7

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

Half course (spring term). M., W., 2:30–4. EXAM GROUP: 7, 8

In recent years we have witnessed a wave of new Internet of Things (IoT) devices in consumer markets. These include wearables such as wrist watch computers and ear phones for personal entertainment, work assistance and bio-metric monitoring. Enabled with energy-efficient



computing cores and sensors, these devices can be programmed to perform a variety of personalized or context-specific tasks at extremely low power consumption. Many believe that IoT will play a key role in the next frontier of computing. In this course we will study IoT and related devices and services. We will focus on new opportunities and challenges, including security on tiny devices and use of machine learning technology to enhance their usage. The course will consist of these five components: Overview of IoT Technology and Services, Networking for IoT, Security and Privacy for IoT, Machine Learning in Small-Data for the Local Context and for Wearables in IoT, and course projects formulation and execution. Industrial partners will help facilitate course projects. Project devices could include: Depth sensing (Kinect and depth cameras), Samsung Galaxy and watch, Raspberry Pi, Pebble watches, Google glass, Biometric sensors (Fitbits, Nike Fuel), Intel Galileo board, NFC (Near Field Communication)/RFID (Radio Frequency Identification), and Chromecast.

Note: Preference given to upper-class undergraduates or graduate students in computer science or in business.

Prerequisite: Programming experience.

[Computer Science 146. Computer Architecture]

Catalog Number: 99684

Instructor to be determined

Half course (spring term). M., W., 1–2:30. EXAM GROUP: 6, 7

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. 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. Programming Languages

Catalog Number: 6841

Stephen N. Chong

Half course (spring term). Tu., Th., 10–11:30. EXAM GROUP: 12, 13

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

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 or 61.

Computer Science 161. Operating Systems

Catalog Number: 4347

Margo I. Seltzer

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 and 61.

***Computer Science 164. Software Engineering**

Catalog Number: 7295 Enrollment: Permission of the instructor is required.

David J. Malan

Half course (spring term). M., W., 2:30–4:30. EXAM GROUP: 7, 8, 9

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. 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

Stratos Idreos

Half course (spring term). M., W., 1–2:30. EXAM GROUP: 6, 7

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

Hanspeter Pfister

Half course (spring term). Tu., Th., 2:30–4, and a weekly section to be arranged.

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 or the Core area requirement for Quantitative 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, 8

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 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: 4, 5

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.*

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: 15, 16*

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 186. Economics and Computation

Catalog Number: 87282

*David C. Parkes**Half course (spring term). M., W., 11:30–1. EXAM GROUP: 4, 5*

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

*Instructor to be determined**Half course (spring term). M., W., 2:30–4. EXAM GROUP: 7, 8*

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.

Prerequisite: 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: 6, 7, 8

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 201. Cyberlaw and Intellectual Property: Advanced Problem Solving Workshop]

Catalog Number: 15116 Enrollment: Limited to 10.

Instructor to be determined

Half course (spring term). M., Tu., 3:20–4:50.

Students in this experimental capstone course will work in teams to solve a series of hard problems-based cases related to cyberlaw and intellectual property.

Note: Offered jointly with the Law School as 2315, and will meet at the Law School.

[*Computer Science 203hf. A Better Internet: Policy and Practice]

Catalog Number: 81745 Enrollment: Students will be selected via an application process.

Instructor to be determined

Half course (throughout the year). Fall: W., 5–7 p.m. EXAM GROUP: Fall: 9

Students will propose ideas for a better Internet, test them with stakeholders, prototype them for industries and organizations, and attack difficult implementation problems in a problem-solving capstone.

Note: Half course through the year, meeting in January at Stanford University. Offered jointly offered at Law School as LAW 2055.

Computer Science 205. Computing Foundations for Computational Science

Catalog Number: 85368

Cristopher R. Cecka

Half course (fall term). M., W., F., at 11, and a weekly section to be arranged. EXAM GROUP: 4

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

Half course (spring term). M., W., F., at 11. EXAM GROUP: 4

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) and have completed Computer Science 205.

[Computer Science 220r. Cryptography: Trust and Adversity]

Catalog Number: 1637

Instructor to be determined

Half course (fall term). Tu., Th., 10–11:30. EXAM GROUP: 12, 13

Modern cryptography. Mathematical tools. Public-key encryptions, digital signatures, key exchanges, zero-knowledge proofs, authentication, oblivious transfer, practical zero-knowledge proofs and financial cryptography, secure multi-party computation, provably secure encryptions. Foundations: Probabilistic encryption and semantic security. Attacks and countermeasures.

Computer Science 221. Computational Complexity

Catalog Number: 5812

Salil P. Vadhan

Half course (spring term). Tu., Th., 10–11:30. EXAM GROUP: 12, 13

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. EXAM GROUP: 13, 14

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

Instructor to be determined

Half course (fall term). Tu., Th., 11:30–1. EXAM GROUP: 13, 14

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 225. Pseudorandomness]

Catalog Number: 4869

Instructor to be determined

Half course (spring term). M., W., 2:30–4. EXAM GROUP: 7, 8

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), 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

Instructor to be determined

Half course (fall term). Tu., Th., 11:30–1.

Important algorithms and their real life applications. Topics include combinatorics, string matching, wavelets, FFT, computational algebra number theory and geometry, randomized algorithms, search engines, page rankings, maximal flows, error correcting codes, cryptography, parallel algorithms.

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. Applications to Boolean functions, automata and geometric functions.

Prerequisite: Computer Science 121 or equivalent.

Computer Science 229r. Topics in the Theory of Computation

Catalog Number: 3730

Jelani Nelson

Half course (fall term). Tu., Th., 11:30–1. EXAM GROUP: 13, 14

Topic for 2013-14: Algorithms for Big Data. This course will cover rigorous methods, based on solid theoretical foundations, for processing massive amounts of data. Topics discussed will include streaming and sketching algorithms, dimensionality reduction, and external memory



algorithms, to name a few.

Prerequisite: comfort with rigorous mathematics, discrete probability, and reasoning about algorithms (e.g. as in CS 124).

Computer Science 244r. Networks Design Projects

Catalog Number: 3018

H. T. Kung

Half course (spring term). M., W., 2:30–4. EXAM GROUP: 7, 8

In recent years we have witnessed a wave of new Internet of Things (IoT) devices in consumer markets. These include wearables such as wrist watch computers and ear phones for personal entertainment, work assistance and bio-metric monitoring. Enabled with energy-efficient computing cores and sensors, these devices can be programmed to perform a variety of personalized or context-specific tasks at extremely low power consumption. Many believe that IoT will play a key role in the next frontier of computing. In this course we will study IoT and related devices and services. We will focus on new opportunities and challenges, including security on tiny devices and use of machine learning technology to enhance their usage. The course will consist of these five components: Overview of IoT Technology and Services, Networking for IoT, Security and Privacy for IoT, Machine Learning in Small-Data for the Local Context and for Wearables in IoT, and course projects formulation and execution. Industrial partners will help facilitate course projects. Project devices could include: Depth sensing (Kinect and depth cameras), Samsung Galaxy and watch, Raspberry Pi, Pebble watches, Google glass, Biometric sensors (Fitbits, Nike Fuel), Intel Galileo board, NFC (Near Field Communication)/RFID (Radio Frequency Identification), and Chromecast.

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: Programming experience.

[Computer Science 246. Advanced Computer Architecture]

Catalog Number: 0979

Instructor to be determined

Half course (fall term). M., W., 1–2:30. EXAM GROUP: 6, 7

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

David M. Brooks

Half course (fall term). M., W., F., 10–11:30. EXAM GROUP: 3, 4

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. EXAM GROUP: 13, 14

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 252r. Advanced Topics in Programming Languages

Catalog Number: 1986

Stephen N. Chong

Half course (fall term). Tu., Th., 2:30–4. EXAM GROUP: 16, 17

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 253r. Virtual Machines]

Catalog Number: 2901

Instructor to be determined

Half course (fall term). M., F., 1–2:30. EXAM GROUP: 6, 7

Note: Preference given to graduate students or upper-class concentrators.

Prerequisite: Computer Science 153 or equivalent.

***Computer Science 260r. Projects and Close Readings in Software Systems - (New Course)**

Catalog Number: 49684

Edward W. Kohler

Half course (spring term). M., W., 1–2:30. EXAM GROUP: 6, 7

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.

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.

Prerequisite: Computer Science 161, or equivalent.

Computer Science 262. Introduction to Distributed Computing

Catalog Number: 7949

James H. Waldo



Half course (spring term). Tu., Th., 2:30–4. EXAM GROUP: 16, 17

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 264. Massively Parallel Computing]

Catalog Number: 37157

Instructor to be determined

Half course (spring term). M., W., F., 1–2:30. EXAM GROUP: 6, 7

This course is an introduction to several modern parallel computing approaches and languages. Covers programming models, hardware architectures, multi-threaded programming, GPU programming with CUDA, cluster computing with MPI, cloud computing, and map-reduce using Hadoop and Amazon's EC2. Students will complete readings, programming assignments, and a final project.

[Computer Science 265. Database Systems]

Catalog Number: 2083

Instructor to be determined

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.

Prerequisite: CS 165 or permission of instructor.

[Computer Science 277. Geometric Modeling in Computer Graphics]

Catalog Number: 3067

Instructor to be determined

Half course (spring term). Tu., Th., 1–2:30. EXAM GROUP: 15, 16

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.

Prerequisite: Computer Science 175.

Computer Science 278. Rendering and Image Processing in Computer Graphics

Catalog Number: 4883

Steven J. Gortler

Half course (spring term). Tu., Th., 1–2:30. EXAM GROUP: 15, 16

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 (formerly *Computer Science 279r). Research Topics in Human-Computer Interaction**

Catalog Number: 1435

Krzysztof Z. Gajos

Half course (fall term). Tu., Th., 10–11:30. EXAM GROUP: 12, 13

Current topics in HCI research, including intelligent interactive systems, usable privacy and security, interaction techniques, design methods, design for emerging markets. Special focus this year is on human computation and crowdsourcing. The course covers core research methods in HCI including experimental design, statistical data analysis, and qualitative methods. Activities will include discussion of primary literature, lectures, assignments, and a research project.

Designed for first year grads from all areas. Advanced undergrads welcome.

Prerequisite: None for graduate students; CS 179 strongly recommended for undergraduates.

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: 15, 16

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 2014: Models and algorithms for collaboration and coordination of multi-agent "mixed networks" comprising people and computer-system agents.

Prerequisite: Computer Science 181 or 182, or equivalents; or permission of instructor.

Computer Science 281. Advanced Machine Learning

Catalog Number: 97848 Enrollment: Limited to 60.

Ryan Prescott Adams

Half course (fall term). M., W., 1–2:30. EXAM GROUP: 6, 7

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 283. Computer Vision

Catalog Number: 4475

Todd Zickler

Half course (fall term). Tu., Th., 11:30–1. EXAM GROUP: 13, 14

Vision as an ill-posed inverse problem: image formation, two-dimensional signal processing; image enhancement and restoration; feature analysis; image segmentation; structure from



motion, texture, and shading; multiple view geometry; pattern classification; and applications.

Note: Offered jointly with the Design School as SCI-06275.

Computer Science 284r. Topics on Computation in Networks and Crowds - (New Course)

Catalog Number: 74473

Yaron Singer

Half course (fall term). M., W., 10–11:30. EXAM GROUP: 3, 4

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

David C. Parkes

Half course (fall term). Tu., Th., 10–11:30. EXAM GROUP: 12, 13

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.

Nicole Immorlica

Half course (spring term). F., 9–12. EXAM GROUP: 2, 3, 4

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. The topic for Spring 2014 is Matching and Market Design.

Prerequisite: Mathematics 21b, Applied Mathematics 21b, or equivalent; Economics 1011a, or equivalent; or permission of instructor.

[Computer Science 287r. Natural Language Processing]

Catalog Number: 3306

Instructor to be determined

Half course (spring term). Tu., Th., 2:30–4.

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 288r. Advanced Topics in Computer Vision - (New Course)

Catalog Number: 62802



Todd Zickler

Half course (spring term). Tu., Th., 11:30–1. EXAM GROUP: 13, 14

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: 13, 14

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.

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 301qc. iLaw: Internet Technology, Law, and Policy]

Catalog Number: 65515

Yochai Benkler (Law School), Terry Fisher (Law School), Urs Gasser (Law School), Lawrence Lessig (Law School), Phil Malone (Law School), Charles Nesson (Law School), John Palfrey (Law School), and Jonathan Zittrain

Quarter course (fall term). Hours to be arranged.

Internet Technology, Law, and Policy. Intensive four-day course on the current state of the



Internet, its problems, and its path. Privacy, property, speech, and governance. A rigorous survey course with a final work product required of students.

***Computer Science 303,304. Statistical Machine Learning**

Catalog Number: 46531,61638

Ryan Prescott Adams 3022

***Computer Science 305,306. Readable, Extensible, High-Performance Software Systems**

Catalog Number: 15739,17423

Edward W. Kohler 1996

***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 313,314. Visual Computing**

Catalog Number: 4273,1628

Hanspeter Pfister 5882

***Computer Science 315,316. Social Computing: Computation and Economics**

Catalog Number: 2892,2433

Yiling Chen 6187 (on leave fall term)

***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 (on leave 2013-14)

***Computer Science 325,326. Intelligent Interactive Systems and Human-Computer**

Catalog Number: 15849,82478

Krzysztof Z. Gajos 6339



***Computer Science 327,328. Mathematical Logic, Theory of Computation**

Catalog Number: 1160,3576

Harry R. Lewis 4455

***Computer Science 343,344. Computer Architecture: Modeling and Design**

Catalog Number: 3932,9266

David M. Brooks 4222 (on leave spring term)

***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. Cryptography: Unbreakable Codes and Financial Cryptography**

Catalog Number: 0218,0255

Michael O. Rabin 7003

***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. On-line Algorithms and Randomized Algorithms**

Catalog Number: 2104,1477

Michael D. Mitzenmacher 7748 (on leave 2013-14)

***Computer Science 361,362. Programming Languages and Semantics**

Catalog Number: 8672,8366

John G. Morrisett 4853 (on leave spring term)

***Computer Science 363,364. Programming Languages and Security**

Catalog Number: 52264,67371

Stephen N. Chong 6340

***Computer Science 365. SEAS Teaching Practicum**

Catalog Number: 8195

John A. Girash 6894



Half course (spring term). M., 2:30–4:30.

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

***Computer Science 377,378. Sketching Algorithms for Massive Data - (New Course)**

Catalog Number: 11014,73819

Jelani Nelson 7260

***Computer Science 379,380. Algorithms for Social Data - (New Course)**

Catalog Number: 37287,20798

Yaron Singer 7269

Cross-listed Courses

[MCB 131. Computational Neuroscience]

[Statistics 221. Statistical Computing and Learning](#)

[*Statistics 285r. Statistical Machine Learning]

Earth and Planetary Sciences

AN HISTORICAL EDITION OF FAS COURSES OF INSTRUCTION

Faculty of the Department of Earth and Planetary Sciences

John H. Shaw, Harry C. Dudley Professor of Structural and Economic Geology (*Chair*)

Jeremy Bloxham, Mallinckrodt Professor of Geophysics, Dean of Science

Kelly V. Chance, Lecturer in Earth and Planetary Sciences

Jennifer S. Cole, Preceptor in Earth and Planetary Sciences

Brian F. Farrell, Robert P. Burden Professor of Meteorology

Peter John Huybers, Professor of Earth and Planetary Sciences and of Environmental Science and Engineering

Miaki Ishii, Professor of Earth and Planetary Sciences