# COMPUTER SCIENCE

*Emeriti: (Professors)* Tom Binford, George B. Dantzig, Edward Feigenbaum,* Robert W. Floyd, John G. Herriot, Donald E. Knuth,* John McCarthy,* William F. Miller, Nils J. Nilsson,* Joseph Oliger, Vaughan Pratt,* Gio Weiderhold*

*Chair*: Hector Garcia-Molina

*Associate Chair for Education:* Russell Shackelford

*Professors:* David Cheriton, Bill Dally, David Dill, Hector Garcia-Molina, Gene H. Golub, Leonidas J. Guibas, Patrick Hanrahan, John Hennessy, Mark A. Horowitz, Oussama Khatib, Monica Lam, Jean-Claude Latombe, Zohar Manna, Edward J. McCluskey, John Mitchell, Rajeev Motwani, Jeffrey D. Ullman, Terry Winograd, Mihalis Yannakakkis

*Associate Professors:* Michael Genesereth, Daphne Koller, Marc Levoy, Nick McKeown, Serge A. Plotkin, Mendel Rosenblum, Yoav Shoham, Jennifer Widom

*Assistant Professors:* Mary G. Baker, Serafim Batzoglou, Dan Boneh, Dawson Engler, Ronald P. Fedkiw, Armando Fox, Christoforos Kozyrakis, Michael Leventon, Christopher Manning, Andrew Ng, Balaji Prabhakar

*Professors (Research):* Richard Fikes, John K. Salisbury

*Professors (Teaching):* Eric S. Roberts, Russell Shackelford

*Courtesy Professors*: Giovanni De Micheli, Bernd Girod, Michael Levitt, Teresa Meng, Grigori Mints, Fouad A. Tobagi

*Courtesy Associate Professors:* Russ Altman, Martin Fischer, John Gill, David Heeger, Oyekunle Olukotun

*Courtesy Assistant Professors:* Joshua B. Tenenbaum, Benjamin Van Roy

*Senior Lecturer:* Margaret Johnson

*Lecturers:* Gerald Cain, Nicholas J. Parlante, Robert Plummer, Mehran Sahami, Patrick Young, Julie Zelenski

*Acting Assistant Professor:* Jan Borchers

*Consulting Professors:* Kurt Konolige, Prabhakar Raghavan

*Visiting Professor:* Nir Friedman

*Visiting Associate Professors:* Peyman Milanfar, Moshe Tennenholz

*Visiting Assistant Professors:* Cynthia Thompson, Sebastian Thrun

*Consulting Assistant Professor:* Brian Jeffrey Fogg

* Recalled to active duty.

*Mail Code:* 94305-9025

*Department Phone:* (650) 723-2273

*Web site:* http://www.cs.stanford.edu/

Courses given in Computer Science have the subject code CS. For a complete list of subject codes, see Appendix B.

The Department of Computer Science (CS) operates and supports computing facilities for departmental education, research, and administration needs.

All CS students have access to a departmental student machine, a Multi-CPU SUN Enterprise 3000, as well as a cluster of public workstations located in the Gates Building. In addition, most students have access to systems associated with their research areas.

Each research group in Computer Science has systems specific to its research needs. These systems range from PC clones and Macs to high-end Multi-CPU compute and file servers. Servers and workstations manufactured by SUN, HP, SGI, Intel, Dell, Apple, and IBM are commonplace.

Support for course work and instruction is provided on systems available through Information Technology Systems and Services (ITSS).

## UNDERGRADUATE PROGRAMS

The department offers both a major and a minor in Computer Science. The requirements for these programs are outlined in the "School of Engineering" section of this bulletin and described in more detail in the *Handbook for Undergraduate Engineering Programs* published by the School of Engineering. The department has an honors program, which is described in the following section.

**This file has been excerpted from the *Stanford Bulletin, 2002-03*, pages 145-160. Every effort has been made to insure accuracy; late changes (after print publication of the bulletin) may have been made here. Contact the editor of the *Stanford Bulletin* via email at arod@stanford.edu with changes, corrections, updates, etc.**

In addition to Computer Science itself, Stanford offers several interdisciplinary degrees with a substantial computer science component. The Computer Systems Engineering major (also in Engineering) allows the study of issues of both computer hardware and software, bridging the gap between traditional CS and Electrical Engineering majors. The Symbolic Systems major (in the School of Humanities and Sciences) offers a chance to explore computer science and its relation to linguistics, philosophy, and psychology. Finally, the Mathematical and Computational Sciences major (also Humanities and Sciences) allows students to explore computer science along with more mathematics, statistics, and operations research.

### HONORS

The Department of Computer Science (CS) offers an honors program for selected undergraduates whose academic records and personal initiative indicate that they have the necessary skills to undertake high-quality research in computer science. Admission to the program is by application only. To apply for the honors program, students must be majoring in Computer Science, have a grade point average (GPA) of at least 3.5 in courses that count toward the major, and achieve senior standing (135 or more units) by the end of the academic year in which they apply. Coterminal master's students are eligible to apply as long as they have not already received their undergraduate degree. Beyond these requirements, students who apply for the honors program must also find a Computer Science faculty member who agrees to serve as the thesis adviser for the project. Thesis advisers must be members of Stanford's Academic Council.

Students who meet the eligibility requirements and wish to be considered for the honors program must submit a written application to the CS undergraduate program office by May 1 of the year preceding the honors work. The application must include a letter describing the research project, a letter of endorsement from the faculty sponsor, and a transcript of courses taken at Stanford. Each year, a faculty review committee will select the successful candidates for honors from the pool of qualified applicants.

In order to receive departmental honors, students admitted to the honors program must, in addition to satisfying the standard requirements for the undergraduate degree, do the following:

1. Complete at least 9 units of CS 191 or 191W under the direction of their project sponsor.
2. Attend a weekly honors seminar Spring Quarter.
3. Complete an honors thesis deemed acceptable by the thesis adviser and at least one additional faculty member.
4. Present the thesis at a public colloquium sponsored by the department.
5. Maintain the 3.5 GPA required for admission to the honors program.

## GRADUATE PROGRAMS

The University's basic requirements for the M.S. and Ph.D. degrees are discussed in the "Graduate Degrees" section of this bulletin.

### MASTER OF SCIENCE

In general, the M.S. degree in Computer Science is intended as a terminal professional degree and does not lead to the Ph.D. degree. Most students planning to obtain the Ph.D. degree should apply directly for admission to the Ph.D. program. Some students, however, may wish to complete the master's program before deciding whether to pursue the Ph.D. To give such students a greater opportunity to become familiar with research, the department has instituted a program leading to a master's degree with distinction in research. This degree is described in more detail in a subsequent section.

Applications for admission to the M.S. program, and all of the required supporting documents, must be received before December 1, 2002 for international applicants and December 15, 2002 for domestic applicants. Exceptions are made for applicants who are already students at Stanford and are applying to the coterminal program. Information on these deadlines is available from the department.

### REQUIREMENTS

A candidate is required to complete a program of 45 units. At least 36 of these must be graded units, passed with an average 3.0 (B) grade point average (GPA) or better. The 45 units may include no more than 21 units of courses from those listed below in Requirements 1 and 2. Thus, students needing to take more than seven of the courses listed in Requirements 1 and 2 actually complete more than 45 units of course work in this program. Only extremely well-prepared students may expect to finish the program in one year; most complete the program in six quarters. Students hoping to complete the program with 45 units should already have a substantial background in computer science, including course work or experience equivalent to all of Requirement 1 and some of the courses in Requirement 2.

*Requirement 1*—The following courses may be needed as prerequisites for other courses in the program: CS 103X, 107, 108; EE 182; MATH 109 or 120.

*Requirement 2*—Students must demonstrate breadth of knowledge in the field by completing the following courses:

1. Area A: Mathematical and Theoretical Foundations:
   a) Required:
      1) Statistics (STATS 116 *or* MS&E 220)
      2) Algorithms (CS 161)
      3) Automata (CS 154)
   b) Choose one of:
      1) Numerical Analysis (CS 137 or 237A)
      2) Logic (CS 157, 257, 258, *or* PHIL 160A)
      3) Mathematical Methods (CS 205)
2. Area B: Computer Systems:
   a) Required: Architecture (EE 182 or 282)
   b) Choose two of:
      1) Operating Systems (CS 140)
      2) Compilers (CS 143)
      3) Introduction to Computer Networks (CS 244A *or* EE 284)
3. Area C: AI and Applications:
   a) Choose two of the following, with at least one 200-level course:
      1) AI (CS 121 or 221)
      2) Databases (CS 145 or 245)
      3) Graphics (CS 148 or 248)

Individual specializations are free to narrow the set of choices in specific areas of the breadth requirement; see the individual specialization sheets in the department office for details. Breadth courses are waived only if evidence is provided that similar or more advanced courses have been taken, either at Stanford or another institution. Courses that are waived rather than taken may not be counted toward the M.S. degree. Breadth courses may be taken on a Satisfactory/No Credit basis provided that a minimum of 36 graded units is presented within the 45-unit program.

*Requirement 3*—At least 1 but no more than 3 units of 500-level seminars must be taken.

*Requirement 4*—A program of 21 units in an area of specialization must be completed. All courses in this area must be taken for letter grades. Eight approved programs are listed below. Students may propose to the M.S. program committee other coherent programs that meet their goals and satisfy the basic requirements.

1. Numerical Analysis/Scientific Computation
   a) CS 237A, 237B, 237C
   b) At least two of: CS 260, MS&E 121, MATH 131, 132, 220A, 220B, 220C, STATS 200
   c) At least two of: CS 223A, 238, 326A, 327A, 328, 336, 337, 339, AA 214A, 214B, STATS 227
2. Systems
   a) CS 240, 242
   b) At least three of: CS 243, 244A, 245, 248, 348B, EE 271, 275
   c) At least 6 more units selected from '2b' and from the following: CS 194, 241, 244B, 244C, 246, 249, 255, 315A, 315B, 342, 343, 344, 345, 346, 347, 348A, 348C, 349, 448, EE 183, 281, 318, 319, 374, 384A, 384B, 384S, 384X, 384Y, 482A, 482B, 487, 488, 489, PSYCH 267

3. Software Theory
   a) CS 242, 243, 256, 258
   b) At least one of: CS 244A, 245, 342, 343, 345
   c) At least one course from the following: CS 255, 261, 351, 355, 356, 361A, 361B, 365, 368
   d) At least one additional course selected from '3b,' '3c,' CS 346
4. Theoretical Computer Science
   a) CS 256, 258, 261 (361A, 361B, or 365 may be substituted for 261)
   b) At least 12 more units from CS 228, 255, 345, 351, 352, 353, 355, 356, 357, 358, 359,* 361A, 361B, 365, 367A, 367B, 368, 369*, MS&E 310
5. Artificial Intelligence
   a) At least four of: CS 222, 223A, 223B, 224M, 224N, 227, 228, 229, 326A
   b) A total of 21 units from the above and from the following: CS 205, 206, 225A, 225B, 226, 246, 256, 257, 262, 270, 274, 276A, 276B, 277, 315A, 323, 327A, 328, 329, 354, 374, 377*, 379*, 426, ECON 286, EE263, 376A, ENGR 205, 209A; LING 235, 238, 239MT; MAN. SCI. 251, 252, 339, 351, 352, 353, PHIL 160B, 169, 298, PSYCH 202, 203, STAT 202, 315B, SYMB SYS 115, 239A, 239B
6. Database
   a) CS 245
   b) Two of: CS 345, 346, 347
   c) Four additional courses selected from '6b' and from the following: CS 222, 240, 242, 243, 244A, 244B, 244C, 246, 249, 255, 262, 270, 271, 276A, 276B, 315A, 315B, 344, 374, 395
7. Human-Computer Interaction
   a) CS 147, 247A, 247B
   b) At least 6 units from: CS 148 or 248, 377 (may be taken repeatedly), 378, 447
   c) A total of 21 units from the above and from the following: COMM 269, 272, CS 249, 270, 271, 348A, 348B, 448, ENGR 145, MS&E 234, 273, 280, 284, LINGUIST 238, ME 101, 115, 313, PSYCH 203, 221, 267, SYMBSYS 115
8. Real-World Computing
   a) At least two of: CS 223A, 223B, 248
   b) At least three of: CS 205, 237A, 237B, 237C, 249, 262, 277, 326A, 348A, 348B, 348C, 36, 374
   c) A total of 21 units from the above and from the following: CS 225A, 225B, 228, 229, 247A, 270, 271, 274, 327A, 328, 336, 448, PSYCH 267

* With consent of Specialization chair.

*Requirement 5*—Additional elective units must be technical courses (numbered 100 or above) related to the degree program and approved by the adviser. Elective courses may be taken on a satisfactory/no credit basis provided that a minimum of 36 graded units are presented within the 45-unit program.

## MASTER OF SCIENCE WITH DISTINCTION IN RESEARCH

A student who wishes to pursue the M.S./CS with distinction in research must first identify a faculty adviser who agrees to supervise and support the research work. The research adviser must be a member of the Academic Council and must hold an appointment in Computer Science. The student and principal adviser must also identify another faculty member, who need not be in the Department of Computer Science, to serve as a secondary adviser and reader for the research report. In addition, the student must complete the following requirements beyond those for the regular M.S./CS degree:

1. *Research Experience:* the program must include significant research experience at the level of a half-time commitment over the course of three academic quarters. In any given quarter, the half-time research commitment may be satisfied by a 50 percent appointment to a departmentally supported research assistantship, 6 units of independent study (CS 393, 395, or 399), or a prorated combination of the two (such as a 25 percent research assistantship supplemented by 3 units of independent study). This research must be carried out under the direction of the primary or secondary adviser.
2. *Supervised Writing and Research:* in addition to the research experience outlined in the previous requirement, students must enroll in at least 3 units of independent research (CS 393, 395, or 399) under the direction of their primary or secondary adviser. These units should be closely related to the research described in the first requirement, but focused more directly on the preparation of the research report described in the next section. Note that the writing and research units described in parts (1) and (2) must be taken in addition to the 21 units required for the specialization, although they do count toward the 45 units required for the degree.
3. *Research Report:* students must complete a significant report describing their research and its conclusions. The research report represents work that is publishable in a journal or at a high-quality conference, although it is presumably longer and more expansive in scope than a typical conference paper. Two copies of the research report must be submitted to the Student Services office in the department three weeks before the beginning of the examination period in the student's final quarter. Both the primary and secondary adviser must approve the research report before the "distinction in research" designation can be conferred.

## DOCTOR OF PHILOSOPHY

Applications to the Ph.D. program and all supporting documents must be received before December 1, 2002 for international applicants and December 15, 2002 for domestic applicants. The following are department requirements (see the Computer Science Ph.D. administrator for further details, or visit http://cs.stanford.edu/degrees/phd.html):

1. A student should plan and successfully complete a coherent program of study covering the basic areas of computer science and related disciplines. The student's adviser has primary responsibility for the adequacy of the program, which is subject to review by the Ph.D. program committee.
2. Each student, to remain in the Ph.D. program, must satisfy the breadth requirement covering introductory level graduate material in major areas of computer science. Once a student fulfills six of eight whole areas of the breadth requirement, he or she may apply for admission to candidacy for the Ph.D. This is typically done by the end of the second year in the program. The student must completely satisfy the breadth requirement by the end of nine quarters (excluding summers), and must pass a qualifying exam in the general area of the expected dissertation.
3. As part of the training for the Ph.D., the student is required to complete at least 4 units (a unit is 10 hours per week for one quarter) as a teaching assistant or instructor for courses in Computer Science numbered 100 or above.
4. The most important requirement is the dissertation. After passing the qualifying examination, each student must secure the agreement of a member of the department faculty to act as the dissertation adviser. (In some cases, the dissertation adviser may be in another department.)
5. The student must pass a University oral examination in the form of a defense of the dissertation. It is usually held after all or a substantial portion of the dissertation research has been completed.
6. The student is expected to demonstrate the ability to present scholarly material orally, both in the dissertation defense and by a lecture in a department seminar.
7. The dissertation must be accepted by a reading committee composed of the principal dissertation adviser, a second member from within the department, and a third member chosen from within the University. The principal adviser and at least one of the other committee members must be Academic Council members.

### Ph.D. MINOR

For a minor in Computer Science, a candidate must complete 20 unduplicated units of computer science course work numbered 200 or above. At least three of the courses have to be master's core courses to provide breadth and one course numbered 300 or above to provide depth. One of the courses taken must include a significant programming project to demonstrate programming efficiency. All courses must be taken for a letter grade of 'B' or higher.

## TEACHING AND RESEARCH ASSISTANTSHIPS

Graduate student assistantships are available. Half-time assistants receive a tuition scholarship for 8, 9, or 10 units per quarter during the academic year, and in addition receive a monthly stipend.

Duties for half-time assistants during the academic year involve approximately 20 hours of work per week. Teaching assistants (TAs) help an instructor teach a course by conducting discussion sections, consulting with students, grading examinations, and so on. Research assistants (RAs) help faculty and senior staff members with research in computer science. Most teaching and research assistantships are held by Ph.D. students in the Department of Computer Science. If there is an insufficient number of Ph.D. students to staff teaching and research assistantships, then these positions are open to a limited number of master's students in the department. However, master's students should not plan on being appointed to an assistantship.

Students with fellowships may have the opportunity to supplement their stipends by serving as graduate student assistants.

# COURSES

(WIM) indicates that the course satisfies the Writing in the Major requirement.

(AU) indicates that the course is subject to the University Activity Unit limitations (8 units maximum).

## GUIDE TO SELECTING INTRODUCTORY COURSES

Students arriving at Stanford have widely differing backgrounds and goals, but most find that the ability to use computers effectively is beneficial to their education. The department offers many introductory courses to meet the needs of these students.

For students whose principal interest is an exposure to the fundamental ideas behind computer science and programming, CS 105 is the most appropriate course. It is intended for students in nontechnical disciplines who expect to make some use of computers, but who do not expect to go on to more advanced courses. CS 105 meets the Area 2b General Education Requirement and includes an introduction to programming, and the use of modern Internet-based technologies. Students interested in learning to use the computer should consider CS 1C (Introduction to Computing at Stanford) or 1U (Introduction to Unix).

Students who intend to pursue a serious course of study in computer science may enter the program at a variety of levels, depending on their background. Students with little prior experience or those who wish to take more time to study the fundamentals of programming should take 106A followed by 106B. Students in 106A need not have prior programming experience. Students with significant prior exposure to programming or those who want an intensive introduction to the field should take 106X, which covers most of the material in 106A and B in a single quarter. CS106A uses ANSI C as its programming language. Beginning in 2002-03, CS106B and X use C++. No prior knowledge of these languages is assumed, and the prior programming experience required for 106X may be in any language. In all cases, students are encouraged to discuss their background with the instructors responsible for these courses.

After the introductory sequence, Computer Science majors and those who need a significant background in computer science for related majors in engineering should take 103, 107 and 108. CS 103 offers an introduction to the mathematical and theoretical foundations of computer science. CS 107 exposes students to a variety of programming paradigms that illustrate critical strategies used in systems development; 108 builds on this material, focusing on the development of large interactive programs based on the object-oriented programming paradigm.

In summary:

For exposure—1C or 1U
For nontechnical use—105
For scientific use—106A
For a technical introduction—106A
For significant use—106A,B or 106X, along with 103, 107, and 108

## NUMBERING SYSTEM

The first digit of a CS course number indicates its general level of sophistication:

| | |
|---|---|
| 1- 99 | service courses for nontechnical majors |
| 100-199 | other service courses, basic undergraduate |
| 200-299 | advanced undergraduate/beginning graduate |
| 300-399 | advanced graduate |
| 400-499 | experimental |
| 500-599 | graduate seminars |

The tens digit indicates the area of Computer Science it addresses:

| | |
|---|---|
| 0- 9 | Introductory, miscellaneous |
| 10-19 | Hardware Systems |
| 20-29 | Artificial Intelligence |
| 30-39 | Numerical Analysis |
| 40-49 | Software Systems |
| 50-59 | Mathematical Foundations of Computing |
| 60-69 | Analysis of Algorithms |
| 70-79 | Typography and Computational Models of Language |
| 90-99 | Independent Study and Practicum |

## NONMAJOR

**CS 1C. Introduction to Computing at Stanford**—For those with limited experience on computers. Introduction to the basics of computing, and a variety of programs, encouraging individual exploration of the programs covered. Topics: word processing, spread sheets, using the WWW and the Internet, and computing resources at Stanford. Macintosh and PC systems. One-hour lecture/demonstration in dormitory clusters prepared and administered weekly by the Resident Computer Coordinator (RCC). Weekly short assignments and final project. Not a programming course.

*1 unit, Aut (Staff)*

**CS 1U. Introduction to Unix**—Tutorial on using the Unix operating system. Topics: text editors, the file system, the C shell, standard Unix utilities, PERL. Includes simple shell programming, but is not a programming course and assumes no prior exposure to programming.

*1 unit, Spr (Staff)*

**CS 12N. The Coming Revolution in Computer Architecture**—Stanford Introductory Seminar. Preference to freshmen. Staying on the historical microprocessor performance curve requires new architectures that exploit the characteristics of emerging technology. This motivates a revolution in computer architecture where what has reigned for the last 50 years will be replaced by one or more radically new architectures. Topics: classic architecture and technology trends that have driven its performance growth, and factors that are leading to the end of this growth; characteristics of technology that can be exploited for future growth and driving application areas; exploration of some alternative organizations and programming systems that are candidates to replace the status quo.

*3 units (Dally) not given 2002-03*

**CS 20N. Internet Auctions and Other Games of Computer Science and Economics**—Stanford Introductory Seminar. Preference to freshmen. A decade ago, auctions were familiar to a small set of people, and the concept of an online auction did not exist. Today millions of people participate in eBay and other online auctions. What is the difference between an English, Dutch, and a Japanese auction? What is the difference between eBay's auction format and that of Amazon? What does game theory have to do with this? What does TCP/IP have to do with either auctions or game theory? How computer science and economics interact with one another. No prior knowledge assumed, but basic mathematical reasoning required.

*3 units, Aut (Shoham)*

**CS 22N. Smart Computers and Other Technological Opportunities**—Stanford Introductory Seminar. Preference to freshmen. How smart are computers now? How smart can we make them, and how soon, and what good will they be? Of the current smart things, how smart are they and what genuine benefits are offered? Outside of computers, what are the technological opportunities for humanity? To what extent is humanity in difficulty with natural resources? What significant threats are there to humanity?

*3 units, Spr (McCarthy)*

**CS 24Q. Digital Actors**—Stanford Introductory Seminar. Preference to sophomores. Digital actors are an emerging field, with applications to video games, movies, simulation and training, manufacturing, and animated web pages. Introduces the computational techniques used to create and animate robotics, geometric computing, computer vision, and graphics. The problem of creating/animating digital actors, technical subproblems. Prerequisite: knowledge of elementary geometry. Recommended: some programming experience.

*3 units (Latombe) not given 2002-03*

**CS 48N. The Science of Art**—Stanford Introductory Seminar. Preference to freshmen. The interwoven histories of science and Western art from the Renaissance to the 19th century. Emphasis is on the revolutions in science and mathematics that have inspired parallel revolutions in the visual arts such as Brunelleschi's invention of linear perspective, Newton's discoveries in geometric optics, and the theories of color vision proposed by Goethe, Young, and Helmholtz. The scientific principles behind image making, including a brief survey of digital image synthesis or computer graphics. However, this is not a course in computer graphics. No programming experience required. Limited enrollment. GER:2b

*3 units, Win (Levoy)*

**CS 50. Problem Solving With Mathematica**—For engineers, physicists, mathematicians, and others who need to solve mathematical or quantitative problems. Comprehensive introduction to Mathematica, an interactive mathematical software package that includes a high-level programming language. Symbolic, numerical, graphical, animation, and programming capabilities, including use of Mathematica to manipulate expressions, find roots, solve differential equations, visualize functions and data, import and export data in arbitrary formats, work with expressions in standard mathematical notation, and perform statistical analyses.

*2 units (Williams) not given 2002-03*

**CS 51. Introduction to Quantum Computing and Quantum Information Theory**—For computer scientists, physicists, mathematicians, engineers, and others who want to learn the capabilities of quantum computers and the necessary quantum mechanics and complexity theory. Topics: quantum algorithms (including Shor's polynomial time algorithm for integer factorization, Grover's database search algorithm, quantum tree search, quantum wavelets), quantum information theory, quantum cryptography, breaking the RSA cryptosystem, quantum teleportation, circuit design, quantum error correction, and examples of prototype quantum computers. Prerequisites: familiarity with elementary matrix algebra and complex numbers.

*2 units, Win (Williams)*

**CS 53N. Open Source Project**—Stanford Introductory Seminar. Preference to freshmen. Recent work has shown how meta compilation can find thousands of errors in systems such as Linux and OpenBSD. Students apply the approach to open source projects, patch the errors they find, and analyze different source bases for relative quality. Working understanding of complex systems. Prerequisite: some programming experience.

*3 units (Engler) not given 2002-03*

**CS 55N. Computer and Information Security**—Stanford Introductory Seminar. Preference to freshmen. Topics: computer security, including the damage caused by break-ins, common holes in computer systems, technological solutions for preventing attacks, cryptography, and legal issues in computer security.

*3 units (Boneh) not given 2002-03*

**CS 55Q. Computer Security in the Electronic Age**—Stanford Introductory Seminar. Preference to sophomores. Based on readings and discussions of current issues in computer security. Topics: the history of codes and ciphers and a summary of basic mathematics used in current cryptography; causes of computer vulnerabilities, including program errors, design flaws, and inherent network and browser limitations; policies and practices that restrict or monitor access to information.

*3 units (Mitchell) not given 2002-03*

**CS 59N. Great Ideas in Computer Science**—Stanford Introductory Seminar. Preference to freshmen. The intellectual tradition of computer science focusing on a set of creative ideas reflecting milestones in the history of the discipline. Topics stressing the interplay between theory and practice: limits of computation, algorithmic efficiency, cryptography, language translation, artificial intelligence, and networking. Prerequisite: introductory computer science at the level of CS106A.

*3 units (Roberts) not given 2002-03*

**CS 68N. Technological Visions of Utopia**—Stanford Introductory Seminar. Preference to Freshmen. The role of computers and other technologies in literary visions of utopian and anti-utopian societies. Readings include classical utopian texts including More's *Utopia* and Bellamy's *Looking Backward*, along with recent books and films in which technology plays a more central role.

*3 units, Aut (Roberts)*

**CS 68Q. Ruler, Compass, and Computer: Computational Representations of Geometry**—Stanford Introductory Seminar. Preference to sophomores. Introduction to some of the mathematical ideas behind commonly used representations and algorithms for geometric objects, focusing on intuitive understanding as opposed to formal development. Such representations and algorithms play an important role in computer graphics, computer vision, and robotics, and all parts of computer science dealing with the physical world. Prerequisite: an introductory CS class such as 106A, B, or X. Recommended: general background knowledge in mathematics and physics.

*3 units (Guibas) not given 2002-03*

**CS 69N. The Two Cultures: Bridging the Gap**—Stanford Introductory Seminar. (Same as ENGLISH 69N.) Preference to freshmen. In 1959, the British physicist and novelist C. P. Snow delivered a lecture at Cambridge University in which he argued that "the intellectual life of the whole of western society is increasingly being split into two polar groups." In Snow's view, these groups, which can be characterized roughly as humanists and scientists, exist as separate cultures that have "almost ceased to communicate at all." Professors in Computer Science and English collaborate to examine the nature of this split, reflected at Stanford by the tendency to divide the campus community into techies and fuzzies, and explore ways to bridge this cultural gap.

*3 units (Roberts, Saldívar) not given 2002-03*

**CS 71N. The Downside of Computing Systems**—Stanford Introductory Seminar. Preference to freshmen. Computers are critical components of our world in such tasks as surgery, air traffic control, and international banking. How computing systems fail, how such failures may affect our society in the future, and how to build and maintain systems to avoid failures. Case studies of computer-related disasters, including the Therac-025 accidents, the Internet worm, and the Ariane 5 crash. Topics: computer security, robust distributed systems, fault-tolerant architectures, organizational behavior.

*3 units (Baker) not given 2002-03*

**CS 73N. Business on the Information Highways**—Stanford Introductory Seminar. Preference to freshmen. Understanding the capabilities of the Internet and its services. Writing for the web. The effect on commerce, education, and health care. Technical and business alternatives. Who will be hurt and who will benefit from the changes occurring? The central project develops sections of a web publication.

*3 units, Win (Wiederhold)*

**CS 74N. Digital Dilemmas**—Stanford Introductory Seminar. Preference to freshmen. The history and evolution of computing and communication technology, and how it affects the fabric of society. Topics: the military-academic-industrial research complex, the Cold War, and the public good; intellectual property and the Internet; the balance between individual privacy and freedom and the security and stability of the state, and the effect of strong cryptography on this balance. Readings, discussion, debates, guest speakers, one or more field trips.

*3 units (Fox) not given 2002-03*

## UNDERGRADUATE

**CS 103A. Discrete Mathematics for Computer Science**—The fundamental mathematical foundations required for computer science. Topics: logic, relations, functions, basic set theory, proof techniques, combinatorics, recursion, and recurrence relations. Corequisite: 106A or X. GER:2c

*3 units, Aut, Win (Johnson)*

**CS 103B. Discrete Structures**—Continuation of 103A. Topics: analysis of algorithms, mathematical formulations of basic data models (linear models, trees, graphs, and sets), regular expressions, grammars. Corequisite: 106B or 106X.

*3 units, Win, Spr (Sahami)*

**CS 103X. Discrete Structures (Accelerated)**—Covers the material in 103A and B in a single quarter. Students who take 103X should feel comfortable with mathematical formalism. GER:2c

*3-4 units, Win (Cain)*

**CS 105. Introduction to Computers**—For non technical majors. What computers are and how they work. Practical experience in programming. Construction of computer programs and basic design techniques. A survey of Internet technology and the basics of computer hardware. Students in technical fields and students looking to acquire programming skills should take 106A or 106X. Students with prior computer science experience at the level of CS106 or above must receive consent of instructor. Prerequisite: minimal math skills. GER:2b

*3-5 units, Aut, Spr (Young)*

**CS 106A. Programming Methodology**—For students in technical disciplines. Broad introduction to the engineering of computer applications, emphasizing software engineering principles: design, decomposition, information hiding, procedural abstraction, testing, and reusable software components. Uses the programming language C, and concentrates on the development of good programming style and on understanding the basic facilities provided by the language. Alternatives: 105, 106X. GER:2b

*3-5 units, Aut, Spr (Sahami), Win (Plummer)*

**CS 106B. Programming Abstractions**—Abstraction and its relation to programming. The software engineering principles of data abstraction, modules, certain fundamental data structures (e.g., stacks and queues), and data-directed design. Recursion and recursive data structures (linked lists and binary trees). Brief introduction to time and space complexity analysis. Uses the programming language C++. Prerequisite: 106A or consent of the instructor. GER:2b

*3-5 units, Win, Spr (Zelenski)*

**CS 106X. Programming Methodology and Abstractions (Accelerated)**—Covers most of the material in 106A,B. Students are expected to have previous programming experience at a level that allows them to understand the concepts presented in 106A. First three weeks focus on understanding how basic programming concepts are expressed in C++. 106B material is covered for the balance. Students who complete 106A should enroll in 106B. 106X can be taken after 106A only with consent of instructor. GER:2b

*3-5 units, Aut, Spr (Zelenski)*

**CS 107. Programming Paradigms**—Intense introduction to a variety of programming paradigms, programming languages, and language implementations. Topics: the compile-time languages, using advanced memory management features of imperative and multithreaded C; the functional paradigm, using LISP; the object-oriented paradigm, using Java; and the generic programming paradigm, introducing C++ and templates. Substantial programming projects. Prerequisite: 106B, 106X, or equivalent.

*3-5 units, Aut, Spr (Cain)*

**CS 108. Object-Oriented Systems Design**—Software design and construction in the context of large OOP libraries. Taught in Java. Topics: review of OOP, the structure of Graphical User Interface (GUI) OOP libraries, GUI application design and construction, OOP software engineering strategies, approaches to programming in teams. Prerequisite: 107.

*3-4 units, Aut, Win (Parlante)*

**CS 110. Introduction to Computer Systems and Assembly Language Programming**—Basic organization of digital computers, buses, registers, processors, I/O, memory systems. Data representation, data structures, and computer arithmetic. Instruction sets and execution; addressing modes. Assembly language programming, including subroutines, interrupts, and traps. Operating systems issues; combines general principles and practice in implementations. Prerequisite: 106B or 106X.

*3-4 units, Spr (Staff)*

**CS 112. Computer Organization and Design**—(Enroll in EE 182.)

*3-4 units, Aut (Fox), Spr (Kozyrakis)*

**CS 118. Introduction to Mechatronics**—(Enroll in EE 118.)

*4 units, Win (Carryer)*

**CS 121. Introduction to Artificial Intelligence**—(Only one of 121/221 counts towards any CS degree program.) Introduction to the key concepts, representations, and techniques used in building practical computational systems (agents) that appear to display artificial intelligence (AI), through the use of sophisticated adaptive information processing algorithms. Topics: history of AI, reactive systems, heuristic search, planning, constraint satisfaction, knowledge representation and uncertain reasoning, machine learning, classification, applications to language, and vision. Prerequisites: 103B or 103X and basic facility with differential calculus, vector algebra, and probability theory.

*3 units, Win (Latombe)*

**CS 137. Introduction to Scientific Computing**—The fundamental issues of numerical computation for the mathematical, computational, and physical sciences, and engineering. Emphasis is from the perspective of the computer scientist. Use of numerical algorithms in engineering practice. Problems of accurately computing solutions in the presence of rounding errors and of computing discrete approximations of solutions which are defined on the continuum. The taxonomy of problem classes with methods for their solution and principles useful for analysis of performance and algorithmic development. Topics: error analysis, the solution of linear and nonlinear equations, interpolation and numerical differentiation, the approximation of integrals, and the solution of differential equations. Prerequisites: 106A; MATH 103 or 113 or equivalents.

*3-4 units, Win (Golub)*

**CS 138. Matlab and Maple for Science and Engineering Applications**—Introduction to use of Matlab and Maple in engineering applications. Emphasis is on the use of software to solve real problems. How the algorithms work, primarily so user may understand their possible limitations. How to use packages to solve a variety of introductory but important problems in: linear systems, eigenvalue problems, ordinary differential equations, elementary statistics, elementary signal processing (Fourier transforms, wavelets), computer algebra, graphical interfaces. Applications for the engineering and physical sciences. Prerequisites: undergraduate linear algebra and a willingness to program.

*3-4 units, Win (Moler)*

**CS 140. Operating Systems and Systems Programming**—The fundamentals of operating systems design and implementation. Basic structure; synchronization and communication mechanisms; implementation of processes, process management, scheduling, and protection; memory organization and management, including virtual memory; I/O device management, secondary storage, and file systems. Prerequisite: 108. Recommended: EE 182.

*3-4 units, Aut, Win (Rosenblum)*

**CS 143. Compilers**—Principles and practices for design and implementation of compilers and interpreters. Topics: lexical analysis; parsing theory (LL, LR, and LALR parsing); symbol tables; type systems; scoping, semantic analysis; intermediate representations, runtime environments; and code generation. Students construct a compiler for a simple object-oriented language during course programming projects. Prerequisites: 103B or X, 107.

*3-4 units, Aut (Back), Win (Adams)*

**CS 145. Introduction to Databases**—A comprehensive introduction to database design and the use of database management systems for applications. The relational model and XML, including the XML query languages XPath and XQuery, and SQL, the standard language for creating, querying, and modifying relational and object-relational databases. Relational algebra, relational design principles based on functional dependencies and normal forms, and the entity-relationship and object-oriented approaches to database design. Other issues include indexes, views, transactions, authorization, integrity constraints, and triggers. Several advanced topics drawn from data warehousing, data mining, OQL, Datalog, temporal databases, middleware, or peer-to-peer systems. Prerequisites: 103B or X, 107.

*3-4 units, Aut (Ullman, Rajaraman), Spr (Widom)*

**CS 147. Introduction to Human-Computer Interaction Design**—Usability and affordances, direct manipulation, systematic design methods, user conceptual models and interface metaphors, design languages and genres, human cognitive and physical ergonomics, information and interactivity structures, design tools and environments. Structured around a set of case studies in which notable interface designs and/or projects are analyzed as illustrative of underlying principles. Discussions of cases and interface analysis and design exercises.

*3-4 units, Aut (Borchers)*

**CS 148. Introductory Computer Graphics**—(For undergraduates; M.S. students or students with a strong interest in continuing in graphics should take 248. Only one of 148 or 248 counts towards any CS degree program.) Introduction to two- and three-dimensional computer graphics. Topics: fundamentals of input and display devices, scan conversion of geometric primitives, two- and three-dimensional transformations and clipping, windowing techniques, curves and curved surfaces, three-dimensional viewing and perspective, hidden surface removal, illumination and color models, OpenGL, and 3-D modeling tools. Emphasis is on the development of practical skills in using graphics libraries and tools. Programming using C/C++ and OpenGL, with demos in SoftImage. Prerequisites: 107, MATH 103.

*3 units, Win (Johnson)*

**CS 154. Introduction to Automata and Complexity Theory**—Regular sets: finite automata, regular expressions, equivalences among notations, methods of proving a language not to be regular. Context-free languages: grammars, pushdown automata, normal forms for grammars, proving languages non-context-free. Turing machines: equivalent forms, undecidability. Nondeterministic Turing machines: properties, the class NP, complete problems for NP, Cook's theorem, reducibilities among problems. Prerequisites: 103B or X.

*3-4 units, Win (Batzoglou), Spr (Motwani)*

**CS 154N. Introduction to NP Completeness**—Turing machines: equivalent forms, undecidability. Nondeterministic Turing machines: properties, the class NP, complete problems for NP, Cook's theorem, reducibilities among problems. Students participate in approximately the last half of 154. Prerequisite: a knowledge of formal languages and automata as in the first part of 154.

*2 units, Win (Batzoglou), Spr (Motwani)*

**CS 155. Computer and Network Security**—For seniors and first-year graduate students. Principles of computer systems security. Attack techniques and how to defend against them. Topics include: network attacks and defenses, operating system holes, application security (web, email, databases), viruses, social engineering attacks, privacy, and digital rights management. Course projects focus on building reliable code. Prerequisite: 140.

*3 units, Spr (Boneh, Mitchell)*

**CS 156. Introduction to Verification and Concurrency**—A taste of logic: propositional, predicate, temporal. Specification and verification of sequential programs: correctness and termination. Concurrent programming: communication and synchronization, principles and algorithms. Specification of concurrent programs: safety and progress. Verification of safety properties: invariants. Prerequisite: 103B or X.

*3 units (Manna) not given 2002-03*

**CS 157. Logic and Automated Reasoning**—Introduction to logic for computer scientists. An elementary exposition from a computational point of view, of propositional logic, predicate logic, axiomatic theories, and theories with equality and induction. Interpretations, models, validity, proof. Automated deduction: polarity, skolemization, unification, resolution, equality. Strategies. Applications. Prerequisite: 103B or X.

*3-4 units, Aut (Genesereth), Spr (Sanchez)*

**CS 157L. Logic and Automated Reasoning Laboratory**

*by arrangment*

**CS 161. Design and Analysis of Algorithms**—Efficient algorithms for sorting, searching, and selection. Algorithm analysis: worst and average case analysis. Recurrences and asymptotics. Data structures: balanced trees, heaps, etc. Algorithm design techniques: divide-and-conquer, dynamic programming, greedy algorithms, amortized analysis. Algorithms for fundamental graph problems, e.g., depth-first search, connected components, topological sort, shortest paths. Possible topics: network flow, string searching, parallel computation. Prerequisite: 103B or X; STATS 116.

*3-4 units, Aut (Cain)*

**CS 162. Introduction to Combinatorics and its Applications**—(Enroll in MATH 108.)

*3 units, Aut (Maclagan)*

**CS 163. Symmetric Functions and Algebraic Combinatorics**—(Enroll in MATH 112.)

*3 units, not given 2002-03*

**CS 191. Senior Project**—Restricted to Computer Science students. Group or individual projects under faculty direction. Register using the section number associated with the instructor.

*1-6 units, any quarter (Staff)*

**CS 191W. Writing Intensive Senior Project**—Restricted to Computer Science students. Group or individual projects under faculty direction. Register using the section number of an Academic Council member. (WIM)

*3-6 units, Aut, Win, Spr (Staff)*

**CS 192. Programming Service Project**—Restricted to Computer Science students. Appropriate academic credit (without financial support) is given for volunteer computer programming work of public benefit and educational value.

*1-4 units, any quarter (Staff)*

**CS 193C. Client-Side Internet Technologies**—JavaScript, Document Object Model, Flash, HTML, Cascading Style Sheets, and XML. Prerequisites: programming experience at the level of CS106A.

*3 units, Win (Young)*

**CS 193D. C++ and Object-Oriented Programming**—C++ programming language and object-oriented programming paradigm. The major

features of C++ 3.0 and the object design principles which apply generally in Object Oriented Languages. Intensive programming assignments. Prerequisites: knowledge of C and basic programming methodology as developed in 106B or 106X.
*3 units, Win (Cain)*

**CS 193I. Internet Technologies**—Programmer-oriented survey of the authoring, distributing, and browsing technologies. The role, use, and implementation of current Internet tools. Topics: TCP/IP; namespace, connections, and protocols. Client/server structures. Web/HTTP/HTML techniques for text, images, links, and forms. Server side programming, CGI scripts. Security and privacy issues. Programming projects on client- and server-side may be in Perl or Java. Languages are introduced as needed. Emphasis is on understanding, exploiting, and extending Internet technologies. Prerequisites: programming fundamentals at the level of 106B or 106X, and the ability to build and debug programs in a Unix environment.
*3 units, Spr (Parlante)*

**CS 193J. Programming in Java**—Hands-on experience to gain practical Java programming skills. Topics: object-oriented programming (classes, objects, messaging, inheritance), Java language features (interfaces, exceptions, packages, concurrency, garbage collection), use of the built-in packages (lang, util, io, networking, awt, swing), applets and servlets, security and verification, Java implementation and the virtual machine. Intensive programming assignments. Prerequisite: knowledge of C language and programming experience at the level of 106B, 106X.
*3 units, Win (Parlante)*

**CS 193N. C# and the .NET Platform**—The object-oriented in the language C#. Software development for the .NET platform including: Windows forms, graphics using GDI+, building custom controls, data access with ADO.NET, developing software for the Internet. Intensive programming assignments. Prerequisite: 106B or 106X.
*3 units, Spr (Plummer)*

**CS 193W. Microsoft Windows Programming**—The fundamentals of programming on the Microsoft Windows platform, focusing on the use of Microsoft Foundation Class (MFC) framework. Other aspects of Windows programming including Microsoft's COM and OLE object models. Requires a significant amount of programming. Prerequisite: knowledge of C++ at level of 108 or 193D.
*3 units (Young) alternate years, given 2003-04*

**CS 194. Software Project**—Student teams complete a significant programming project through the phases of design, specification, coding, and testing under faculty supervision. Lectures on software engineering methodologies. Implementation; well written proposal, specification, and software design document; demonstration of a prototype design and the final product. Prerequisite: 108. (WIM)
*3 units, Win (Young), Spr (Plummer)*

**CS 196. Microcomputer Consulting**—Overview of computer consulting, focusing on Macintosh and Intel-compatible systems. Topics: operating systems, networks, troubleshooting, and consulting methodology. Biweekly lectures emphasize on-campus computing environments. Students work as consultants in campus computer clusters and in residences. Prerequisite: 1C.
*2 units, Aut, Spr (Staff)*

**CS 197. Mainframe and Workstation Computer Consulting**—Computer consulting in a workstation and server environment, focusing on the UNIX operating system under the SUN, HP, and SGI hardware systems. Topics: UNIX fundamentals, consulting tips, networking, and systems administration. Students work as on-duty consultants at the Sweet Hall computer cluster. Pre- or corequisite: 1U.
*2 units, Win, Spr (Staff)*

**CS 198. Teaching Computer Science**—Teach a small discussion section of 106A while learning the fundamentals of teaching a programming language at the introductory level. Two workshops and one meeting weekly focus on teaching skills, techniques and course specifics. Application and interview required; email cs198@cs for information. Prerequisite: 106B or 106X.
*4 units, Aut, Win, Spr (Roberts, Irani, Belknap)*

**CS 199. Independent Work**—Special study under faculty direction, usually leading to a written report. Letter grade given; if this is not appropriate, enroll in 199P. Register using the section number associated with the instructor.
*1-6 units, any quarter (Staff)*

**CS 199P. Independent Work**—Like 199, but graded Satisfactory/No Credit.
*1-6 units, any quarter (Staff)*

## UNDERGRADUATE AND GRADUATE

**CS 200. Undergraduate Colloquium**—Strongly recommended for junior-year CS majors as a way to build contacts with faculty. Weekly presentations by faculty and people from industry informally describing their views of computer science as a field and their experience as computer scientists. (AU)
*1 unit, Win (Staff)*

**CS 201. Computers, Ethics, and Social Responsibility**—Primarily for majors entering computer-related fields. Analysis of the ethical and social issues related to the development and use of computer technology. Background in ethical theory, and social, political, and legal considerations. Scenarios in problem areas: privacy, reliability and risks of complex systems, and the responsibility of professionals for the applications and consequences of their work. Prerequisite: 106B or 106X. (WIM)
*3-4 units, Spr (Johnson)*

**CS 205. Mathematical Methods for Robotics, Vision, and Graphics**—Overview of some of the continuous mathematics background necessary for research in robotics, vision, and graphics. Possible topics: linear algebra; ordinary and partial differential equations; dynamic systems and stochastic estimation (Kalman filtering); vector and tensor calculus; calculus of variations. Prerequisites: 106B or 106X; MATH 51 and 113; or equivalents.
*3 units, Aut (Fedkiw)*

**CS 206. Technical Foundations of Electronic Commerce**—The explosion of electronic commerce poses new problems for computer science and puts a new spin on old ones. Topics include: searching hyperlinked structures; data mining; online auctions and other trading mechanisms; safe exchange; copyright protection and security; online payment mechanisms; web software infrastructure; personalization and user tracking; integration of catalogs and other trading information. Prerequisites: understanding of basic computer science concepts covered in 103 and 107, or equivalent set of courses; ability to follow simple combinatorial, probabilistic, and algorithmic arguments.
*3 units, Spr (Ullman)*

**CS 211. Logic Design**—(Enroll in EE 275.)
*3 units, Aut, Win (McCluskey)*

**CS 212. Computer Architecture and Organization**—(Enroll in EE 282.)
*3 units, Aut (Kozyrakis), Win (Olukotun)*

**CS 221. Artificial Intelligence: Principles and Techniques**—(Only one of 121 or 221 counts towards any CS degree program.) Broad technical introduction to core concepts and techniques in artificial intelligence. Topics: search, planning, constraint satisfaction, knowledge representation, probabilistic models, machine learning, neural networks, vision, robotics, and natural language understanding. Prerequisites: 103B or 103X or PHIL 160A, 106B, or 106X, and exposure to basic concepts in probability. Recommended: 107 and facility with basic differential calculus.
*3-4 units, Aut (Ng)*

**CS 222. Knowledge Representation**—Declarative knowledge representation methods in artificial intelligence. Topics: time and action, defaults, compositional modeling, object-oriented representation, inheritance, ontologies, knowledge on the Web, knowledge servers, multiple views, qualitative modeling. Prerequisite: basic familiarity with logic. Recommended: prior exposure to artificial intelligence as in 121/221.
*3 units, Win (Fikes)*

**CS 223A. Introduction to Robotics**—Topics: robotics foundations in kinematics, dynamics, control, motion planning, trajectory generation, programming and design. Recommended: knowledge of matrix algebra.
*3 units, Win (Khatib)*

**CS 223B. Introduction to Computer Vision**—Fundamental issues and techniques of computer vision. Image formation, edge detection and image segmentation, stereo, motion, shape representation, recognition. Project or final. Prerequisite: 205 or equivalent.
*3 units, Spr (Leventon)*

**CS 224M. Multi-Agent Systems**—For advanced undergraduates, M.S. students, and beginning Ph.D. students. Topics: logics of knowledge and belief, other logics of mental state, theories of belief change, multi-agent probabilities, essentials of game theory, social choice and mechanism design, multi-agent learning, communication. Applications are discussed where appropriate, but the emphasis is on conceptual matters and theoretical foundations. Prerequisities: knowledge of basic probability theory and first-order logic.
*3 units, Win (Shoham)*

**CS 224N. Natural Language Processing**—(Same as LINGUIST 237.) Develops an in-depth understanding of the algorithms available for the processing of linguistic information and the underlying computational properties of natural languages. Morphological, syntactic, and semantic processing from a linguistic and an algorithmic perspective. Focus is on modern quantitative techniques in NLP: using large corpora, statistical models for acquisition, representative systems. Prerequisites: LINGUIST 138/238 or CS 121/221, and programming experience. Recommended: basic familiarity with logic and probability.
*3-4 units, Spr (Manning)*

**CS 225A. Experimental Robotics**—Hands-on experience with robotic systems. Topics: kinematic and dynamic control of motion, compliant motion and force control, sensor-based collision avoidance, motion planning, dynamic skills, and robot-human interfaces. Limited enrollment. Prerequisite: 223A.
*3 units, Spr (Khatib)*

**CS 225B. Robot Programming Laboratory**—Hands-on introduction to the techniques of robot programming for robotics and non-robotics students. Series of guided exercises in which students program mobile robots to exhibit increasingly complex behavior (simple dead reckoning and reactivity, goal-directed motion, localization, complex tasks). Topics: basics of motor control and sensor characteristics; sensor fusion, model construction, and robust estimation; control regimes (subsumption, potential fields); active perception; reactive planning architectures; various topics in sensor-based control, including vision-guided navigation. Student programmed robot contest. Programming is in C++ on Unix or Windows machines, done in teams. Prerequisites: basic programming skills at the level of 106B, 106X, 205, or equivalent.
*3-4 units, Aut (Konolige)*

**CS 227. Reasoning Methods in Artificial Intelligence**—Technical presentation of algorithmic techniques for problem solving in AI. Combines formal algorithmic analysis with a description of recent applications. Topics: propositional satisfiability, constraint satisfaction, planning and scheduling, diagnosis and repair. Focus is on recent results. Prerequisites: familiarity with the basic notions in data structures and design and with techniques in the design and analysis of algorithms. Recommended: previous or concurrent course in AI.
*3 units, Spr (Nayak)*

**CS 228. Probabilistic Models in Artificial Intelligence**—Probabilistic modeling languages suitable for representing complex domains, algorithms for reasoning and decision making using these representations, and learning these representations from data. Focus is on graphical modeling languages such as Bayesian belief networks, extensions to temporal modeling using hidden Markov models and dynamic Bayesian networks, and extensions to decision making using influence diagrams and Markov decision processes. Recent applications to domains (speech recognition, medical diagnosis, data mining, statistical text modeling, and robot motion planning). Prerequisites: understanding of basic concepts in probability theory and in design and analysis.
*3 units (Staff) not given 2002-03*

**CS 229. Statistical Learning and Pattern Classification**—Topics: foundations of statistical pattern recognition, parametric and non-parametric density estimation, linear and nonlinear classifiers, decision trees, Bayesian and neural networks, reinforcement learning, learning theory, and recent trends such as boosting and support vector machines. Focus is on the underlying concepts and their application to various problems in vision, speech, language processing, animation, control, etc. Prerequisites: background in linear algebra, basic probability theory, and statistics; ability to write computer programs in one or more commonly used languages.
*3 units, Win (Ng)*

**CS 237. Advanced Numerical Analysis**—Three quarter graduate sequence designed to acquaint students in mathematical and physical sciences and engineering with the fundamental theory of numerical analysis. Examples from applications.

**CS 237A. Numerical Linear Algebra**—First in a three quarter graduate sequence. Solution of systems of linear equations: direct methods, error analysis, structured matrices; iterative methods and least squares. Parallel techniques. Prerequisites: 137, MATH 103 or 113.
*3 units, Aut (Golub)*

**CS 237B. Numerical Solution of Partial Differential Equations I**—Linear multistep methods and Runge-Kutta methods for ordinary differential equations: zero-stability, A-stability, and convergence. Elliptic partial differential equations: finite difference, finite element, and spectral methods. Parabolic partial differential equations: stability, convergence, and qualitative properties. Hyperbolic partial differential equations: stability convergence and qualitative properties. Prerequisites: MATH 130, 131.
*3 units, Win (Fedkiw)*

**CS 237C. Numerical Solution of Partial Differential Equations II**—Advanced numerical methods for partial differential equations. Nonlinear hyperbolic equations and systems. Combined solution methods from elliptic, parabolic, and hyperbolic problems. Examples include: Burgers equation, Euler equations for compressible flow, Navier Stokes equations for incompressible flow. Prerequisites: 237A, 237B, MATH 130, 131.
*3 units, Win (Staff)*

**CS 238. Parallel Methods in Numerical Analysis**—Recent developments in parallel computer technology have made it necessary to reformulate numerical algorithms to exploit the full potential of this technology. Emphasis is on the different techniques for obtaining maximum parallelism in various numerical algorithms, especially those occurring when solving matrix problems and partial differential equations, and the subsequent mapping onto the computer. Implementation issues on parallel computers. Topics: parallel architecture, programming models, matrix computations, FFT, fast multiple methods, domain decomposition, graph partitioning. Prerequisite: 237A or ME 200A, or consent of instructor. Recommended: familiarity with differential equations, and experience in advanced programming language such as F90, C, C++.
*3 units (Staff) not given 2002-03*

**CS 240. Advanced Topics in Operating Systems**—Advanced study in OS topics and exposure to recent developments in OS research. Readings/lectures on classic and new papers. Topics: virtual memory management, synchronization and communication, file systems, protection and security, operating system extension techniques, fault tolerance, and the history and experience of systems programming. Prerequisite: 140 or equivalent.

*3 units, Win, Spr (Engler)*

**CS 241. Internet Technologies and Systems**—Architecture, design, and implementation of Internet-scale services and applications. Scalability, high availability, fault tolerance, and robustness for Internet services. Cluster-based runtime systems for Internet workloads, implementation and deployment challenges, economics of deploying and operating a service. Extending Internet services to mobile, wireless, and post-PC computing devices. Service-centric view of the Internet, including composition of services and mass customization. Research agenda for Internet-scale services. Significant programming assignments involve building and deploying Internet service prototypes. Limited enrollment. Prerequisites: 193I or equivalent experience; 240 and 244A.

*3 units, Spr (Fox)*

**CS 242. Programming Languages**—The basic elements of programming languages and programming paradigms: functional, imperative, and object-oriented. Introduction to formal semantic methods. Modern type systems, higher-order functions and closure, exceptions and continuations. Runtime support for different language features. Emphasis is on separating the different elements of programming languages and styles. First half uses Lisp and ML to illustrate concepts; second half a selection of object-oriented languages. Prerequisite: 107, or experience with Lisp, C and some object-oriented language.

*3 units, Aut (Mitchell)*

**CS 243. Advanced Compiling Techniques**—The theoretical and practical aspects of building modem compilers. Topics: intermediate representations, basic blocks and flow-graphs, data flow analysis, register allocation, global code optimizations, and interprocedural analysis. Prerequisite: 143 or equivalent.

*3-4 units, Win (Lam)*

**CS 244A. Introduction to Computer Networks**—(Autumn Quarter, enroll in EE 284.) Structure and components of computer networks; functions and services; packet switching; layered architectures; ISO's Open Systems Interconnections (OSI) reference model; physical layer; data link layer; error control; window flow control; media access control protocols used in local area networks (Ethernet, Token Ring, FDDI) and satellite networks; network layer (datagram service, virtual circuit service, routing, congestion control, IP); transport layer (UDP, TCP); application layer.

*3-4 units, Aut (Tobagi), Win (McKeown)*

**CS 244B. Distributed Systems**—Distributed operating systems and applications issues, emphasizing high-level protocols and distributed state sharing as the key technologies. Topics: distributed shared memory, object-oriented distributed system design, distributed directory services, atomic transactions and time synchronization, file access, process scheduling, process migration and remote procedure call focusing on distribution, scale, robustness in the face of failure, and security. Prerequisites: 240, 244A.

*3 units, Spr (Cheriton)*

**CS 244C. Distributed Systems Project**—Companion project option for students taking 244B. Corequisite: 244B.

*3-6 units, Spr (Cheriton)*

**CS 245. Database System Principles**—File organization and access, buffer management, performance analysis, and storage management. Database system architecture, query optimization, transaction management, recovery, concurrency control. Reliability, protection, and integrity. Design and management issues. Prerequisites: 145, 161.

*3 units, Win (Garcia-Molina)*

**CS 246. Information Integration**—In recent years, there has been a marked increase in the amount of structured data available on the Internet. Accessing this information in an integrated way is complicated by conceptual heterogeneity among the data sources, or differences in their structure and vocabulary. Approaches to solving this problem. Topics: notations and models for structured data such as XML and RDF, information integration techniques, standard schemas and vocabularies, data structuring technology, and applications such as in corporate logistics, e-commerce, civil engineering, and health care. Large-scale datawebs and prospects for building a fully integrated semantic web, essentially a world wide web for databases. Prerequisites: 145, 157.

*3 units, Spr (Genesereth)*

**CS 247A. Human-Computer Interaction: Interaction Design Studio**—Intended as preparation for project-based courses such as 377, 447, or ME 293. Systematic presentation and experience with the methods used in interaction design, including needs analysis, user observation, idea sketching, concept generation, scenario building, storyboards, user character stereotypes, usability analysis, and market strategies. Prerequisite: 147 or ME 101.

*3-4 units, Win (Staff)*

**CS 247B. Contextual and Organizational Issues in Human-Computer Interaction**—(Same as MS&E 430.) Focus is on the contextual issues associated with designing and using computer interfaces and technology, providing insights into, experience with, and ways of understanding issues in work and consumer settings that influence the design of computer interfaces. Student team projects develop skills in: observing individuals and groups of people in context, using models of work and other activity to extend their design capabilities, identifying constraints and tradeoffs on designs within the context of use, and observing and working with people in interdisciplinary design groups. Enrollment limited. Prerequisite: 247A, or consent of instructor.

*3-4 units, Spr (Hinds)*

**CS 248. Introduction to Computer Graphics**—(Only one of 148 or 248 counts towards any CS degree program.) The fundamentals of input and display devices, scan conversion of geometric primitives, 2D and 3D geometric transformations, clipping and windowing, scene modeling and animation, algorithms for visible surface determination, introduction to local and global shading models, color, and real-time rendering methods. Written assignments and programming projects. Prerequisites: 107, MATH 103 or equivalent.

*3-5 units, Aut (Levoy)*

**CS 249. Object-Oriented Programming from a Modeling and Simulation Perspective**—Object-oriented programming techniques and issues, emphasizing programming as modeling and simulation. Topics: large-scale software development approaches, encapsulation, use of inheritance and dynamic dispatch, design of interfaces and interface/implementation separation, exception handling, design patterns, minimalizing dependencies and value-oriented programming. The role of programming conventions/style/restrictions in surviving object-oriented programming for class libraries, frameworks, and programming-in-the-large; general techniques for object-oriented programming. Prerequisites: knowledge of C and basic programming methodology as developed in 106B or 106X, 107, basic knowledge of C++ (may be taken concurrently). Recommended: 193D.

*3-5 units, Win (Cheriton)*

**CS 255. Introduction to Cryptography**—Intended for advanced undergraduates and graduate students. Introduction to the basic theory and practice of cryptographic techniques used in computer security. Topics: encryption (single and double key), digital signatures, pseudo-random bit generation, authentication, electronic commerce (anonymous cash, micropayments), key management, PKI, zero-knowledge protocols. Prerequisite: basic understanding of probability theory.

*3 units, Win (Boneh)*

**CS 256. Formal Methods for Reactive Systems**—Formal methods for specification, verification, and development of concurrent and reactive programs. Reactive systems: syntax and semantics, fairness requirements. Specification language: temporal formulas (state, future, and past) and w-automata. Hierarchy of program properties: safety, guarantee, obligation, response, persistence, and reactivity. Deductive verification of programs: verification diagrams and rules, completeness. Modularity. Parameterized programs. Algorithmic verification of finite-state programs. Prerequisite: 157 or PHIL 160A, or equivalent.
*3 units (Manna) not given 2002-03*

**CS 256L. Formal Methods for Reactive Systems Laboratory**
*2 units (Manna) not given 2002-03*

**CS 257. Automated Deduction and Its Applications**—Proving theorems and extracting information from proofs. Uses in software engineering (program specification, synthesis, and verification) and artificial intelligence (commonsense and robotic planning, natural-language understanding). The foundations of logic programming. Deductive tableaux, nonclausal resolution, skolemization, building theories into unification and inference rules, term rewriting, inductive theorem proving. The design of theorem provers. Prerequisite: 157.
*3 units (Staff) not given 2002-03*

**CS 258. Introduction to Programming Language Theory**—Syntactic, operational, and semantic issues in the mathematical analysis of programming languages. Type systems and non-context-free syntax. Universal algebra and algebraic data types. Operational semantics given by rewrite rules; confluence and termination. Scott-semantics for languages with higher-type functions and recursion. Treatment of side-effects. Prerequisites: 154, 157 or PHIL 160A.
*3 units, Win (Mitchell)*

**CS 260. Concrete Mathematics**—Mathematics for the analysis of algorithms: recurrences, summations, generating functions, asymptotics. Elementary combinatorics, discrete probability, and number theory. Prerequisites: 103B or 103X, MATH 42, or equivalent.
*3 units (Staff) not given 2002-03*

**CS 261. Optimization and Algorithmic Paradigms**—Algorithms for network optimization: max-flow, min-cost flow, matching, assignment, and min-cut problems. Introduction to linear programming. Use of LP duality for design and analysis of algorithms. Approximation algorithms for NP-complete problems such as Steiner Trees, Traveling Salesman, and scheduling problems. Randomized algorithms. Introduction to on-line algorithms.
*3 units (Plotkin) not given 2002-03*

**CS 262. Computational Genomics**—For graduate students. Introduction to the applications of computer science to genomics research and to basic concepts in genomics from a computer science point of view. Topics: algorithms for sequence analysis and their applications to the most current genomics research, sequence alignments, hidden Markov models, multiple alignment algorithms and heuristics such as Gibbs sampling, and the probabilistic interpretation of alignments. Applications of these tools to sequence analysis: DNA sequencing and assembly, genomic annotation of repeats, genes, and regulatory sequences, microarrays and gene expression, and comparative genomics. Prerequisites: 161 or familiarity with basic algorithmic concepts. Basic knowledge of genetics helpful, but not required.
*3 units, Spr (Batzoglou)*

**CS 270. Introduction to Biomedical Informatics: Fundamental Methods**—(Same as BIOMEDIN 210.) Issues in the modeling, design, and implementation of computational systems for use in biomedicine. Topics: basic knowledge representation, controlled terminologies in medicine and biological science, fundamental algorithms, information dissemination and retrieval, knowledge acquisition, and ontologies. Emphasis is on the principles of modeling data and knowledge in biomedicine and on translation of resulting models into useful automated systems. Recommended: understanding of the basic principles of object-oriented systems (at the level of Computer Science 107).
*3 units, Aut (Staff)*

**CS 271. Introduction to Clinical Systems**—(Same as BIOMEDIN 211.) Survey of the major applications in clinical informatics, including imaging systems, information systems, and decision-support technology. Emphasis is on the system requirements, relevant data, standards, algorithms, and implementation issues in each area. Prerequisite: 210.
*3 units, Win (Staff)*

**CS 274. Representations and Algorithms for Computational Molecular Biology**—(Same as BIOMEDIN 214.) Introduction to the basic computational issues and methods used in bioinformatics, including access and use of biological data sources on the Internet. Topics: basic algorithms for alignment of biological sequences and structures, computing with strings, phylogenetic tree construction, hidden Markov models, computing with networks of genes, basic structural computations on proteins, protein structure prediction, protein threading techniques, homology modeling, molecular dynamics and energy minimization, statistical analysis of 3D biological data, integration of diverse data sources, knowledge representation and controlled terminologies for molecular biology, graphical display of biological data, genetic algorithms and genetic programming applied to biological problems. See instructor for unit options. Prerequisites: programming skills and understanding of matrix algebra.
*1-5 units, Spr (Altman)*

**CS 275A. Musical Information: An Introduction**—(Enroll in MUSIC 253.)
*1-4 units, Win (Selfridge-Field)*

**CS 275B. Applications of Musical Information: Query, Analysis, and Style Simulation**—(Enroll in MUSIC 254.)
*1-4 units, Spr (Selfridge-Field)*

**CS 276A. Text Information Retrieval, Mining, and Exploitation: Basic Concepts**—Basic and advanced techniques for text-based information systems: efficient text indexing; Boolean, vector space, and probabilistic retrieval models; evaluation and interface issues; web search including crawling, link-based algorithms, and web metadata; peer-to-peer search. Implementation of major elements of a search engine.
*3 units, Aut (Manning, Raghavan, Schuetze)*

**CS 276B. Text Information Retrieval, Mining, and Exploitation: Advanced Topics**—Document clustering, classification, routing, and recommendation systems. Machine learning methods. Information extraction methods: terminologies and ontology acquisition, named entity recognition, coreference resolution, web wrappers, and web agents. Natural language processing techniques: summarization, cross-lingual retrieval, event tracking, question answering, and text mining. Multimedia information retrieval. Biomedical text: special constraints, knowledge discovery, improved performance from integrating textual information.
*3 units, Win (Manning, Raghavan, Schuetze)*

**CS 277. Experimental Haptics**—Introduction to the essential elements of haptics as it relates to creating touch feedback in simulated or virtualized environments. Goal is to provide students with the skills to develop virtual reality haptic simulators and applications. Theoretical topics: psychophysical issues, performance and design of haptic interfaces, haptic rendering methods for 3-D virtual environments, and haptic simulation and rendering of rigid and deformable solids. Applied topics: introduction to C++Builder, Open GL and basic haptic library, implementation of haptic rendering algorithms, collision detection in 3-D environments, design of real-time models for soft tissue simulation. Guest speakers. Directed lab/programming exercises; a more open-ended final project. Enrollment limited to 20. Prerequisites: 145. Recommended: 223.
*3 units, Spr (Salisbury)*

SCHOOL OF ENGINEERING

**CS 298. Seminar on Teaching Introductory Computer Science**—Opportunity for faculty and undergraduate and graduate students who are interested in teaching to discuss specific topics raised by teaching computer science at the introductory level. Prerequisite: consent of instructor.

*1-3 units (Roberts) not given 2002-03*

## PRIMARILY FOR GRADUATE STUDENTS

**CS 300. Departmental Lecture Series**—For first-year Computer Science Ph.D. students. Presentations by members of the department faculty, each describing informally his or her current research interests and views of computer science as a whole. (AU)

*1 unit, Aut (Staff)*

**CS 315A. Parallel Computer Architecture and Programming**—The principles and tradeoffs in the design of parallel architectures. Emphasis is on naming, latency, bandwidth, and synchronization in parallel machines. Case studies on shared memory, message passing, data flow, and data parallel machines illustrate techniques. Architectural studies and lectures on techniques for programming parallel computers. Programming assignments on one or more commercial multiprocessors. Prerequisites: EE 282, and reasonable programming experience.

*3 units (Staff) not given 2002-03*

**CS 315B. Parallel Programming Project**—Continuation of 315A. A significant parallel programming project is required using shared memory, message passing, or data parallel machines. Parallel programming languages and their implementation, performance debugging or parallel programs, parallel data structures and algorithms. Prerequisite: 315A, or consent of instructor.

*3 units (Staff) not given 2002-03*

**CS 316A. Logic Synthesis of VLSI Circuits**—(Enroll in EE 318.)

*3 units, Win (DeMicheli)*

**CS 316B. Computer-Aided System Design Laboratory**—(Enroll in EE 319.)

*3 units, Spr (De Micheli)*

**CS 317. Fault Tolerant Computing Systems**—(Enroll in EE 489.)

*3 units (McCluskey) alternate years, given 2003-04*

**CS 319. Topics in Digital Systems**—Advanced material is often taught for the first time as a topics course, perhaps by a faculty member visiting from another institution. Students may therefore enroll repeatedly in a course with this number. See *Time Schedule* for topics currently being offered.

*3 units, by arrangement*

**CS 323. Common Sense Reasoning in Logic**—Formalizing common sense knowledge and reasoning using situation calculus with nonmonotonic logics, especially circumscription. Variations of situation calculus. Formalizing context. Formalizing facts about knowledge. Prerequisite: basic knowledge of logic such as 157, or PHIL 160A.

*3 units (McCarthy) not given 2002-03*

**CS 326A. Motion Planning**—For students interested in computer graphics, geometrical computing, robotics, and/or artificial intelligence. Computing object motions is central to many application domains such as design, manufacturing, robotics, animated graphics, medical surgery, drug design. Basic path planning methods generate collision-free paths among static obstacles. Extensions include uncertainty, mobile obstacles, manipulating movable objects, and maneuvering with kinematic constraints. Configuration space is a unifying concept, geometric arrangements are a basic combinatorial structure. Theoretical methods with applications in various domains: assembly planning, radiosurgery, graphic animation of human figures.

*3 units, Spr (Latombe)*

**CS 327A. Advanced Robotics**—Emerging areas of human-centered robotics and interactive haptic simulation of virtual environments. Topics: redundancy; task-oriented dynamics and control, whole-body control-task and posture decomposition, cooperative robots, haptics and simulation, haptically augmented teleoperation, human-friendly robot design. Prerequisites: 223A or equivalent.

*3 units, Spr (Khatib)*

**CS 328. Topics in Computer Vision**—Fundamental issues of, and mathematical models for, computer vision. Sample topics: camera calibration, texture, stereo, motion, shape representation, image retrieval, experimental techniques. Student papers and project. Prerequisites: 205, 223B, or equivalents.

*3 units (Staff) not given 2002-03*

**CS 329. Topics in Artificial Intelligence**—Advanced material is often taught for the first time as a topics course, perhaps by a faculty member visiting from another institution. Students may therefore enroll repeatedly in a course with this number.

*by arrangement*

**CS 336. Advanced Methods in Matrix Computation**—Eigenvalue problems: perturbation theory, Lanczos method, Jacobi method. Parallel implementation. Singular value problems. Generalized eigenvalue problems. Polynomial equations. Prerequisite: 237A.

*3 units (Staff) not given 2002-03*

**CS 337. Numerical Methods for Initial Boundary Value Problems**—Initial boundary value problems are solved in different areas of engineering and science modeling phenomena including wave propagation and vibration, and fluid flow. Numerical techniques for such simulations in the context of applications. Emphasis is on stability and convergence theory for methods for hyperbolic and parabolic initial boundary value problems, and the development of efficient methods for these problems.

*3 units (Staff) not given 2002-03*

**CS 339. Topics in Numerical Analysis**—Advanced material is often taught for the first time as a topics course, perhaps by a faculty member visiting from another institution. Students may therefore enroll repeatedly in a course with this number. See *Time Schedule* for current topics.

**CS 339. Sparse Matrix Algorithms**—Many problems in science and engineering require fundamental matrix computations for their solution, and these matrices are often mostly zero. Topics: direct methods, but with some application to iterative methods; sparse matrix-vector multiply; matrix-matrix multiply and transpose; forward/backsolve; LU and Cholesky factorization; singular value decomposition; reordering methods including the use of graph partitioning methods; and updating/downdating a Cholesky factorization. Prerequisites: numerical and linear algebra, graph theory, data structures and algorithms, and MATLAB. Includes programming assignments in C or Fortran, and MATLAB.

*3 units, Win (Davis)*

**CS 340. Level Set Methods**—Introduction to modeling surfaces with implicit functions. Focus is on the mathematical and computational techniques required to treat dynamic implicit surfaces. Level set methods can be used for a variety of applications including multiphase flow (such as bubbles and drops), image processing, computer vision, and graphics.

*3 units (Fedkiw) not given 2002-03*

**CS 342. Programming Language Design**—Problems of programming language design and comparison of traditional solutions. Possible topics: formal semantics, implementation considerations, extensibility, very high level languages, evaluation of language designs, the innovative features of a variety of modern programming languages. Prerequisites: 242, 243.

*3 units (Mitchell) not given 2002-03*

**CS 343. Topics in Compilers**—Advanced topics in compilers. Topics change every quarter; course may be taken repeatedly for credit. Prerequisite: 243.
*3 units, Spr (Lam)*

**CS 344. Projects in Computer Networks**—For students with a strong interest in computer networks from novel applications to physical layer coding schemes; software to hardware; theory to design-and-build. Teams of two or more complete a small research project of sufficient quality and interest to merit presentation at a conference, or to form the basis of a new business, e.g., studies of network traces, network traffic visualization tools, home-networking, analysis of performance of cable-modems, novel web applications, or novel router architecture. Enrollment limited to 20. Prerequisites: 244A or EE 284. Recommended: EE 384A, 384X, 384Y.
*3 units (McKeown) not given 2002-03*

**CS 345. Advanced Topics in Database Systems**—Advanced topics in the area of database and information systems. Content differs in each offering; may be taken multiple times for credit with instructor approval. Prerequisites: 145, 245 for some offerings.
*3 units, Spr (Staff)*

**CS 346. Database System Implementation**—A major database system implementation project realizes the principles and techniques covered in earlier courses. Students independently build a complete database management system, from file structures through query processing, with a personally designed feature or extension. Lectures on project details and advanced techniques in database system implementation, focusing on query processing and optimization. Guest speakers from industry on commercial DBMS implementation techniques. Prerequisites: 145, 245, programming experience in C++.
*3-5 units, Aut (Widom)*

**CS 347. Transaction Processing and Distributed Databases**—The principles and system organization of distributed databases. Data fragmentation and distribution, distributed database design, query processing and optimization, distributed concurrency control, reliability and commit protocols, and replicated data management. Distributed algorithms for data management: clocks, deadlock detection, and mutual exclusion. Heterogeneous and federated distributed database systems. Overview of commercial systems and research prototypes. Prerequisites: 145, 245.
*3 units, Spr (Staff)*

**CS 348A. Computer Graphics: Geometric Modeling**—The mathematical tools needed for the geometrical aspects of computer graphics and especially modeling smooth shapes. Fundamentals: homogeneous coordinates, transformations, and perspective. Theory of parametric and implicit curve and surface models: polar forms, Bezier arcs and de Casteljau subdivision, continuity constraints, B-splines, tensor product, and triangular patch surfaces. Subdivision surfaces and multiresolution representations of geometry. Representations of solids and conversions among them. Mesh generation, simplification, and compressions. Prerequisites: linear algebra and discrete algorithms.
*3-4 units, Win (Guibas)*

**CS 348B. Computer Graphics: Image Synthesis Techniques**—Intermediate level, emphasizing the sampling, shading, and display aspects of computer graphics. Topics: local and global illumination methods including radiosity and distributed ray tracing, texture generation and rendering, volume rendering, strategies for anti-aliasing and photo-realism, human vision and color science as they relate to computer displays, and high-performance architectures for graphics. Written assignments and programming projects. Prerequisite: 248 or equivalent. Recommended: exposure to Fourier analysis or digital signal processing.
*3-4 units, Spr (Hanrahan)*

**CS 348C. Computer Graphics: Animation Techniques**—Overview of computer animation techniques. Topics: traditional principles of animation, physical simulation, procedural methods, and motion capture based animation. Focus is on computer science aspects of animation. The basics including kinematic and dynamic modeling techniques; exploration of current research topics such as motion re-targeting, learning movements and behaviors, and video based modeling and animation. Hands-on animation experience through class projects.
*3-4 units (Staff) not given 2002-03*

**CS 348D. Vision and Image Processing**—(Enroll in PSYCH 267.)
*1-3 units (Heeger) not given 2002-03*

**CS 349. Topics in Programming Systems**—Advanced material is often taught for the first time as a topics course, perhaps by a faculty member visiting from another institution. Students may enroll repeatedly in a course with this number. See *Time Schedule* for topics currently being offered.
*by arrangement*

**CS 351. Topics in Complexity Theory and Lower Bounds**—Focus is on one of basic machine models and complexity measures, their properties and relationships; complexity classes and their properties; reductions and complete problems; concrete representative problems from important complexity classes. Techniques for establishing limits on the possible efficiency of algorithms, and concrete lower bounds based on the following models of computation: decision trees, straight line programs, communication games, branching programs, PRAMs, boolean circuits. Approximation algorithms and the complexity of approximations. Pseudo-randomness and cryptography. Prerequisite: 154, or equivalent.
*3 units, Aut (Sivakumar)*

**CS 353. Algebraic Logic**—Algebraic methods relevant to computer science. Lattice theory: partial orders, monoids, closure systems, topologies, fixpoint theorems. Universal algebra: HSP, free algebras, equational theories, Birkhoff's theorem, completeness of equational logic. Algebras for logic: Boolean, Heyting, cylindric. Categories: limits, adjunctions, algebraic theories, monads. In 2002-03 only, Chu spaces, linear logic, and higher dimensional automata. Prerequisites: 157; or PHIL 160A, 161; or equivalents.
*3 units, Aut (Pratt)*

**CS 354. Probabilistic Reasoning in Computing**—The basics of Bayesian probability theory as applied to computing and intelligence systems. Emphasis is on working through applications and understanding relevant theory. Relevant probability theory and techniques: interpretations, graphical and network models, information theory, decision theory, inference, and alternative approaches. Probabilistic aspects of computational problems in learning, search, data analysis, neural, and dynamic systems. Some topics by guest lecturers. Prerequisites: 106B or 106X, 221, a knowledge of basic statistical measures as in PSYCH 60, and basic math.
*3 units (Staff) not given 2002-03*

**CS 355. Advanced Topics in Cryptography**—For graduate students. Topics: pseudo-random generation, zero knowledge protocols, elliptic curve systems, threshold cryptography, security analysis using random oracles, lower and upper bounds on factoring and discrete log. Prerequisite: 255.
*3 units, Aut (Boneh)*

**CS 356. Automatic Formal Verification Techniques**—Automatic methods for formally verifying hardware, protocol, and software system designs. Topics: state graph and automata models of system behavior. Automata on infinite strings. Linear and branching-time temporal logic. Model-checking. Modeling real-time systems. Analysis methods based on Boolean formulas, and other ways of coping with the state explosion problem. Exploiting abstractions. Applications to circuits, algorithms, and protocols. Case studies use a variety of verification tools. Prerequisite: 154 or 254. Recommended: good understanding of basic automata and complexity theory, and undergraduate-level background in computer science.
*3 units (Dill) alternate years, given 2003-04*

**CS 357. Topics in Formal Methods**—Formal methods for the specification, verification, analysis, and systematic development of real-time and hybrid systems. Hybrid systems involve continuous changes and discrete transitions. Computational models: timed and phase transition systems, timed and hybrid automata. Specification: timed and hybrid statecharts, metric and hybrid temporal logics, duration calculus. Statecharts. Structured specification. Verification rules and diagrams. Refinement techniques. Algorithmic verification of finite-state systems. Advanced research topics. Prerequisite: 256 or equivalent.
*3-5 units (Manna, Sipma) not given 2002-03*

**CS 357C. Automata on Infinite Objects**—Topics: automata on infinite words (Buchi, Muller, Rabin automata); automata on infinite trees; alternating automata on infinite words and trees; relationship with linear-time and branching time logics; relationship with monadic second-order logics; complexity and decidability issues; applications to verification and synthesis of reactive programs; applications to game theory.
*3 units (Manna, Sipma) not given 2002-03*

**CS 358. Topics in Programming Language Theory**—Possible topics of current research interest in the mathematical analysis of programming languages: structured operational semantics, domain theory, semantics of concurrency, rich type disciplines, problems of representation independence, and full abstraction. May be repeated for credit. Prerequisites: 154, 157, 258, or equivalents.
*3 units (Mitchell) not given 2002-03*

**CS 359. Topics in Theory of Computation**—Advanced material is often taught for the first time as a topics course, perhaps by a faculty member visiting from another institution. Students may therefore enroll repeatedly in a course with this number. See *Time Schedule* for topics currently being offered.
*by arrangement*

**CS 361A. Advanced Algorithms**—Advanced data structures: union-find, self-adjusting data structures and amortized analysis, dynamic trees, Fibonacci heaps, universal hash function and sparse hash tables, persistent data structures. Advanced combinatorial algorithms: algebraic (matrix and polynomial) algorithms, number theoretic algorithms, group theoretic algorithms and graph isomorphism, online algorithms and competitive analysis, strings and pattern matching, heuristic and probabilistic analysis (TSP, satisfiability, cliques, colorings), local search algorithms. Prerequisite: 161 or 261, or equivalents.
*3 units, Aut (Motwani)*

**CS 361B. Advanced Algorithms**—Topics: fundamental techniques used in the development of exact and approximate algorithms for combinational optimization problems, e.g., generalized flow, multicommodity flow, sparsest cuts, generalized Steiner trees, load balancing, and scheduling. Using linear programming, emphasis is on LP duality for design and analysis of approximation algorithms; interior point methods for LP. Techniques for development of strongly polynomial algorithms.
*3 units (Plotkin) not given 2002-03*

**CS 365. Randomized Algorithms**—Design and analysis of algorithms that use randomness to guide their computations. Basic tools, from probability theory and probabilistic analysis, that are recurrent in algorithmic applications. Randomized complexity theory and game-theoretic techniques. Algebraic techniques. Probability amplification and derandomization. Applications: sorting and searching, data structures, combinatorial optimization and graph algorithms, geometric algorithms and linear programming, approximation and counting problems, parallel and distributed algorithms, online algorithms, number-theoretic algorithms. Prerequisites: 161 or 261, STATS 116, or equivalents.
*3 units, Win (Motwani)*

**CS 367A. Parallel Computation**—Introduction to theoretical issues in parallel computation. Properties of parallel computation models and algorithm design techniques specific to each model, including systolic arrays, mesh-connected computers, hypercube-related networks, and PRAM. Topics: algorithms for sorting, connected components, shortest paths, and other basic problems. Upper and lower bounds for randomized and deterministic routing on hypercube and related networks. Techniques for reducing the processor-time product for PRAM algorithms.
*3 units (Plotkin) not given 2002-03*

**CS 367B. Parallel Computation**—Advanced parallel algorithms. Focus is on developing techniques for the design of parallel algorithms on the PRAM model of computation and its derivatives. Possible topics: efficient randomized parallel algorithms for symmetry-breaking and related problems. Derandomization techniques. Parallel sorting. Deterministic and randomized parallel algorithms for flows and related problems; assignment problem, matching in general graphs. Evaluation of straight-line code, P-complete problems.
*3 units (Plotkin) not given 2002-03*

**CS 368. Geometric Algorithms**—Graduate-level introduction to the basic techniques used in the design and analysis of efficient geometric algorithms including: convexity, triangulation, sweeping, partitioning, and point location. Voronoi and Delaunay diagrams. Arrangements and convex polytopes. Intersection and visibility problems. Geometric searching and optimization. Random sampling methods. Impact of numerical issues in geometric computation. Example applications to motion planning, visibility preprocessing, model-based recognition, and GIS. Prerequisite: 161.
*3 units, Spr (Guibas)*

**CS 369. Topics In Analysis of Algorithms**—Advanced material is often taught for the first time as a topics course, perhaps by a faculty member visiting from another institution. Students may therefore enroll repeatedly in a course with this number. See Time Schedule for topics currently being offered.
*by arrangement*

**CS 374. Algorithms in Biology**—Algorithms and computational models applied to molecular biology and genetics. Current, exciting algorithms from a variety of biological areas. Topics: protein folding, phylogeny and haplotypes, genome rearrangements, sequence comparison, DNA computing, and self-assembly. May be repeated for credit. Prerequisites: 161, 262 or 274, or BIOCHEM 218, or equivalents.
*3 units, Aut (Batzoglou)*

**CS 377. Topics in Human-Computer Interaction**—Topics of current research interest in human-computer interaction. Contents change each quarter. May be repeated for credit. See http://hci.stanford.edu/courses/ for specific offerings.
*3 units, Aut, Win, Spr (Staff)*

**CS 378. Phenomenological Foundations of Cognition, Language, and Computation**—Critical analysis of theoretical foundations of the cognitive approach to language, thought, and computation. Contrasts of the rationalistic assumptions of current linguistics and artificial intelligence with alternatives from phenomenology, theoretical biology, critical literary theory, and socially-oriented speech act theory. Emphasis is on the relevance of theoretical orientation to the design, implementation, and impact of computer systems as it affects human-computer interaction.
*3-4 units (Winograd) not given 2002-03*

**CS 379. Interdisciplinary Topics**—Advanced material that relates computer science to other disciplines is often taught for the first time as a topics course, perhaps by a faculty member visiting from another institution. Students may therefore enroll repeatedly in a course with this number. See *Time Schedule* for topics being currently offered.
*by arrangement*

**CS 390A,B,C. Curricular Practical Training**—Provides educational opportunities in high technology research and development labs in the computing industry. Qualified computer science students engage in internship work and integrate that work into their academic program.

Students register during the quarter they are employed and must complete a research report outlining their work activity, problems investigated, key results, and any follow-on projects they expect to perform. Meets the requirements for Curricular Practical Training for students on F-1 visas. 390 A, B and C may each be taken only once.

*1 unit, any quarter (Motwani)*

**CS 393. Computer Laboratory**—For CS graduate students. A substantial computer program is designed and implemented; written report required. Recommended as a preparation for dissertation research. Register using the section number associated with the instructor. Prerequisite: consent of instructor.

*1-9 units, any quarter (Staff)*

**CS 394. Business Management for Computer Scientists and Electrical Engineers**—Focus is on the functional areas necessary for making successful business decisions. Topics: corporate strategy, new product development, marketing, sales, distribution, customer service, and financial accounting. How to identify and analyze issues in each of these areas in a rapidly changing world. A framework and tool set is developed for formulating, evaluating, and recommending action from the general manager point of view and for communicating and defending ideas in a team environment. Enrollment limited to 60. See http://www.stanford.edu/class/cs394/. Prerequisite: graduate student in Computer Science or Electrical Engineering.

*3-4 units, not given 2002-03*

**CS 395. Independent Database Project**—For graduate students in Computer Science. Use of database management or file systems for a substantial application or implementation of components of database management system. Written analysis and evaluation required. Register using the section number associated with the instructor. Prerequisite: consent of instructor.

*1-6 units, any quarter (Staff)*

**CS 399. Independent Project**—Letter grade only.

*1-9 units, any quarter (Staff)*

**CS 399P. Independent Project**—Graded satisfactory/no credit.

*1-9 units, any quarter (Staff)*

## EXPERIMENTAL

**CS 426. Genetic Algorithms and Genetic Programming**—Genetic algorithm is a domain-independent algorithm for search, optimization, and machine learning. Genetic programming is a domain-independent automatic programming technique that extends the genetic algorithm to the breeding of populations of computer programs capable of producing human-competitive results. Topics: mathematical basis for genetic algorithms; implementation on parallel computers and field-programmable gate arrays; problems of system identification, control, classification, analysis of genome and protein sequences; automatic synthesis of the design of topology, sizing, placement, and routing of analog electrical circuits; automatic synthesis of controllers and antennas and other complex network structures and reverse engineering of metabolic pathways and networks of chemical reactions.

*3 units (Koza) alternate years, given 2003-04*

**CS 428. Information Processing for Sensor Networks**—Introduction to the literature on sensor networks and the fundamental issues in designing and analyzing sensor network information processing applications. Localization and tracking as canonical examples to expose important constraints in scaling and deploying these sensor networks. Topics include querying, data routing, power management, and network self-organization. How these base capabilities can support high-level information processing tasks.

*2 units, Spr (Guibas)*

**CS 444A. Recovery Oriented Computing**—Exploration of techniques for designing, building, and measuring dependable systems, with an emphasis on fault recovery and fault avoidance. Overview of techniques for failure avoidance and recovery from Internet systems, software engineering, classic fault tolerance, and formal methods; guest speakers with experience across broad range of these areas; modeling and dealing with human operator error and designer errors. Small-group interactive workshop format with readings from contemporary literature; mandatory research project culminating in publishable-quality article; demonstration of a prototype; and public presentation at a poster session.

*2-4 units (Fox) not given 2002-03*

**CS 444N. Mobile and Wireless Networks and Applications**—How mobility affects networks, systems, and applications. Mobility of devices and end-users has behavioral implications at all layers of the traditional Internet protocol stack, from the MAC layer up through the application layer. Topics: how mobility affects network protocols and how it affects different functional aspects of systems including security, privacy, file systems, resource discovery, resource management (including energy usage), personal online identities, and other areas. Student groups design and implement mobile applications and system features of their choosing. Prerequisites: 240, 244A, 244B, or equivalents.

*3 units, Spr (Baker)*

**CS 447. Interdisciplinary Interaction Design**—(Same as ME 325.) (Formerly ME 293.) Small teams develop innovative technology prototypes that combine product and interaction design. Focus is on software and hardware interfaces, interaction, design aesthetics, and some underpinnings of successful design: a reflective, interactive design process, group dynamics of effective interdisciplinary teamwork, and working with users. Prerequisite: 247A.

*3-4 units (Kelley, Winograd) not given 2002-03*

**CS 448. Topics in Computer Graphics**—In-depth study of an active research topic in computer graphics. Topic changes each quarter, e.g., exotic input and display technologies, graphics architectures, topics in modeling shape and motion, experiments in digital television, interactive workplaces, introduction to hand-drawn cartoon animation. Readings from literature and a project. May be taken repeatedly for credit. Prerequisite: 248 or consent of instructor. Check http://graphics.stanford.edu/courses/ for specific offerings.

**CS 448. Physics Based Animation for Modeling Virtual Humans**—A tour through physics based animation methods with a focus on modeling virtual humans. Rigid body dynamics for modeling bones, articulated rigid bones and constraints for modeling joints, 3D deformable models and mesh generation for tissue (fat, muscle, etc.) including anisotropic constituitive modeling and contractile forces for muscles, 2D deformable models for skin and clothing, 1D models for hair, etc. Emphasis will be on algorithms, e.g. ordinary differential equations, collision detection and modeling, etc. The class is open to all students for 1 unit, and may be taken for 3 units by those students submitting a final project.

*1-3 units, Spr (Fedkiw)*

**CS 468. Topics in Geometric Algorithms**—Advanced seminar covering different topics related to geometric computing. Recent offerings: shape matching, proximity and nearest-neighbor problems, visibility and motion planning, and collision detection. Readings from the literature and a presentation or a project may be required. May be taken multiple times for credit. Prerequisite: 368, or consent of instructor.

*2 units, Aut, Win (Guibas)*

**CS 471. E-Business Process Foundry**—Project-based course. Cross-disciplinary teams from the Computer Science Department and the Graduate School of Business produce prototypes of operating business processes that utilize web technologies. Small project teams make regular presentations and progress reports to the class and industry experts. Each team includes individuals with technical knowledge of databases, scripting languages, and web services, as well as business knowledge of the non-virtual processes that provide real goods and services. Past teams have successfully implemented prototype systems in daily use by existing nonprofit organizations.

*4 units (Fox, Patell) not given 2002-03*

**CS 499. Advanced Reading and Research**—For CS graduate students. Register using the section number associated with the instructor. Prerequisite: consent of instructor.
*1-15 units, any quarter (Staff)*

## GRADUATE SEMINARS

**CS 510. Digital Systems Reliability Seminar**—(Enroll in EE 385A.)
*1-4 units, Aut, Win, Spr (McCluskey)*

**CS 525. Seminar on Knowledge Acquisition for Expert Systems**—(Same as BIOMEDIN 230.) For graduate students. Discussion of experimental approaches to the construction of expert-system knowledge bases. Topics: interviewing techniques, formal and informal approaches to modeling expert knowledge, and automated tools that facilitate knowledge acquisition. Enrollment limited to 20. Prerequisite: one course in artificial intelligence.
*1-2 units (Musen) alternate years, given 2003-04*

**CS 528. Broad Area Colloquium for Artificial Intelligence, Geometry, Graphics, Robotics and Vision**—Weekly series of informal research talks on topics related to perceiving, modeling, manipulating, and displaying the physical world. The computational models and numerical methods underlying these topics. Brings together faculty and students in these five closely related areas. (AU)
*1 unit, Aut, Win, Spr (Levoy)*

**CS 530. Applied Mathematics: Scientific Computing Seminar**
*1 unit (Staff) not given 2002-03*

**CS 531. Numerical Analysis: Scientific Computing Seminar**
*1 unit, Aut, Win, Spr (Golub)*

**CS 540. Seminar on Computer Systems**—(Enroll in EE 380.)
*1 unit, Aut, Win, Spr (Allison)*

**CS 545. Database Research Seminar**—Presentations of current research and industrial innovation in the area of database and information systems.
*1 unit, Win (Widom)*

**CS 545G. Genome Database Seminar**—Reading, demonstrations, and analysis of databases used to encode molecular-biology information, including DNA sequences, protein sequence and structure, gene expression, and other functional-genomics data. Emphasis is on representation and integration of the many diverse data sources, and their presentation for biomedical and pharmaceutical researchers. Issues: data structures and ontologies, cross referencing, quality control and error detection, search processes, suitability of different DBMSs, data provenance, and privacy protection for patient-derived information. Introduction and discussion by the instructors and presentations by experts from commercial and research organizations. May be combined with a 395 project. (AU)
*1 unit, Spr (Wiederhold, Karp)*

**CS 547. Human-Computer Interaction Seminar**—Weekly speakers on topics related to human-computer interaction design. (AU)
*1 unit, Aut, Win, Spr (Winograd)*

**CS 548. Distributed Systems Research Seminar**—Recent research in distributed operating systems, computer communications, parallel machines, parallel programming, and distributed applications. Invited speakers from Stanford and elsewhere present topics and results of current interest. (AU)
*1 unit, Spr (Cheriton)*

**CS 559. Seminar on Mathematical Theory of Computation**—Topics vary each year and may include: logic and its relation to computation, programming language analysis and design, specification and verification of software and hardware systems, theories of concurrency, approaches to static analysis and program state. Invited speakers present recent results and summaries of articles from the current literature. (AU)
*1 unit, by arrangement (Mitchell) not given 2002-03*

**CS 579. Frontiers in Interdisciplinary Biosciences**—(Crosslisted in multiple departments in the schools of Humanities and Sciences, Engineering, and Medicine; students should enroll directly through their affiliated department, otherwise enroll in CHEMENG 459.) An introduction to cutting-edge research involving interdisciplinary approaches to bioscience and biotechnology; for specialists and non-specialists. Organized and sponsored by the Stanford BioX Program. Three seminars each quarter address a broad set of scientific and technical themes related to interdisciplinary approaches to important issues in bioengineering, medicine, and the chemical, physical, and biological sciences. Leading investigators from Stanford and throughout the world present the latest breakthroughs and endeavors that cut broadly across many core disciplines. Pre-seminars introduce basic concepts and provide background for non-experts. Registered students attend all pre-seminars in advance of the primary seminars, others welcome. Prerequisite: keen interest in all of science, engineering, and medicine with particular interest in life itself. Recommended: basic knowledge of mathematics, biology, chemistry, and physics.
*1 unit, Aut, Win, Spr (Robertson)*

## OVERSEAS STUDIES

Courses approved for the Computer Science major and taught overseas can be found in the "Overseas Studies" section of this bulletin, or in the Overseas Studies office, 126 Sweet Hall.

### KYOTO

**CS 112K. Computer Organization and Design**—(Same as EE 182.)
*3-4 units, Aut (Fox), Spr (Kozyrakis)*