Yale University
Department of Computer Science

GRADUATE HANDBOOK

2006-2007 Edition

1. Introduction

The Yale Department of Computer Science was founded in 1969 as a small graduate program. It now includes 19 regular faculty members, 9 lecturers and affiliated visiting faculty members, 11 research scientists and postgraduate fellows, more than 65 graduate students, and more than 100 undergraduate majors. The Department offers more than 30 undergraduate and graduate courses annually as well as a number of reading and project courses and seminars.

The Ph.D. degree program in Computer Science stresses original research, by the student as an individual and as a member of the community of scholars. To this end, the course requirements are minimal and students normally begin research by the fall term of their second year of graduate study.

The terminal M.S. degree program in Computer Science brings students to the cutting edge of the field and provides them with a solid foundation on which to build their future careers. In 2001, Yale celebrated its 300th anniversary. The Computer Science

Department is almost an order of magnitude younger, reaching its 32nd birthday in the same year. But its influence on the University, within the research community, and on society as a whole, belies its young age. Computers are a dominant presence in almost every walk of life, and it is becoming increasingly difficult for people to answer the question, “How did we ever live without them?”

The Department of Computer Science was founded by people who had a vision. This vision was how computer science would fit into the unique spirit of Yale University, an institution oriented to an unusual degree around undergraduate education and close interdepartmental collaboration. The Department has always had close ties to mathematics and engineering, but has increasingly experienced collaborations with other disciplines important to Yale, including psychology, linguistics, economics, business, statistics, music, medicine, physics and more. It is through these collaborations that the importance of computer science in a broader sense is best appreciated.

DEPARTMENT CHAIR
Abraham Silberschatz

DIRECTOR OF GRADUATE STUDIES
Drew McDermott

PROFESSORS
Dana Angluin
James Aspnes (LOA Fall 2006)
Julie Dorsey
Stanley Eisenstat
Joan Feigenbaum
Michael Fischer
David Gelernter
Paul Hudak
Ravindran Kannan (LOA Fall 2006)
Drew McDermott
Vladimir Rokhlin
Holly Rushmeier (LOA Spring 2007)
Martin Schultz
Zhong Shao (LOA Spring 2007)
Abraham Silberschatz
Daniel Spielman
Steven Zucker

ASSOCIATE PROFESSORS
Richard Yang
Brian Scassellati

ADJUNCT PROFESSORS
Willard Miranker
Gil Kalai
Carsten Schuermann

SENIOR LECTURER
Robert Dunne

SENIOR RESEARCH SCIENTIST
Craig Douglas

RESEARCH SCIENTISTS
Robert Bjornson
Nicholas Carriero
Michael Godwin
Michael Hines

ASSOCIATE RESEARCH SCIENTISTS
Diana Resasco
Yosi Keller

POSTDOCTORAL ASSOCIATES
Fernando Esponda
Bodo Manthey

RESEARCH AFFILIATES
Stuart Feldman
Balaji Veerarahgaven

COURTESY APPOINTMENTS TO COMPUTER SCIENCE
PROFESSORS
Ronald Coifman
Stephen Morse
ASSOCIATE PROFESSOR
Mark Gerstein
ASSISTANT PROFESSORS
Kei Cheung
Yiorgos Makris
Edmund Yeh

2. Research in Computer Science

An academic research effort in any field is supposed to establish outer boundaries. Commercial efforts demonstrate what is profitable; academic efforts show what is possible. In physics, mathematics and the other
large, long-established areas, most research gains lead to relatively small changes in the character and perimeter of the field. Computer science is different. It's brand new. When a few researchers parachuted into this field in the 1940's, they had no territory to extend.

They invented computer science from scratch, more or less. They faced basic questions in science, algorithms, design, and philosophy; and by answering them, they established the intellectual basis of the field. What does “computable” mean? How do we classify computations, distinguish practical from impractical from impossible? How should programming languages be designed, and what difference do they make? Is it possible in principle to build a mind out of software? And so on.

When the Yale Department of Computer Science was founded in 1969, the fledgling field was viewed as consisting of four primary areas: Theory of Computation, Artificial Intelligence, Numerical Analysis, and Systems. Over the years, these areas have developed so quickly in both depth and breadth that each now consists of many sub-areas. Within Artificial Intelligence, areas such as machine learning, knowledge acquisition systems, robotics, computer vision, neural networks, and more have grown. Systems research now includes graphics, databases, operating systems, networking, and the entire sub-field of programming languages, which includes formal semantics, compilers, programming environments, software engineering, and object-oriented, functional, and logic programming. Similarly, Theory of Computation now includes computability, complexity theory, design and analysis of algorithms, cryptography, distributed computing, discrete mathematics, and more. Each of these many new areas now boast their own conferences, journals, federal research programs, and so on.

Each of these areas also has developed into both theoretical and experimental disciplines. For example, security theory establishes the correctness and robustness of idealized cryptographic protocols, but real systems need to be built and fielded to ensure that the protocols withstand realistic attacks. Type theory and category theory help establish the foundations of programming languages, but real languages need to be designed and built to ensure that programmers can use them productively. And learning and planning theory tells us what the limits of automated intelligence are, but robots need to be built to demonstrate the viability of these results. At Yale, both theoretical research and applied research are valued highly.

Computer Science has also grown beyond its own bounds to become a multi-disciplinary field that touches many other sciences as well as arts and humanities. Aside from the obvious overlaps with engineering and mathematics, there are natural connections with physics, economics, law, management, psychology, biology, medicine, music, philosophy, and linguistics. Indeed, members of the CS faculty have engaged in collaborations with each of these areas. These efforts have affected our curriculum through the establishment of cross-listed courses, interdisciplinary majors (such as Cognitive Science), special “tracks” and dual majors, and the creation of the Applied Math program. They have also led to interdisciplinary research centers, such as the Center for Scientific Computing (CS, Math), the Center for Computational Vision and Control (CS, EE, and Medicine) and the Center for Internet Studies (CS, SOM, Law).

Of course, with such diversity and depth, it has become increasingly difficult for any computer science department to cover all areas, and no one really tries. The many changes that are taking place in the field have led the Department to pursue a vision for

the future a bit different from the original structure upon which it was founded. At Yale, the current focus of Computer Science is in these specific areas: Algorithms and Complexity Theory, Distributed Computing, Machine Learning, Programming Languages and Compilers, Scientific Computing and Applied Math, Graphics, Vision and Robotics, and Security and Cryptography.

3. Graduate Programs

The Department offers two graduate programs: a Doctoral Program leading to a Doctor of Philosophy (Ph.D.) degree, and a terminal Master's Program leading to a Master of Science (M.S.) degree. The Doctoral Program is intended for students preparing for a career in teaching and/or research. The terminal Master's Degree Program is intended for students who want advanced study in computer science but do not intend to go on for the Ph.D.

A student may apply to either the Doctoral Program or to the terminal Master's Program. A student seeking the Ph.D. should apply directly to the Doctoral Program, even though he or she intends to obtain a Master's degree along the way. A student who has completed the Master's Program and decides so go on for a Ph.D. is not guaranteed admission to the Doctoral Program and must apply in the normal way.

3.1 The Doctoral Program

The Doctoral Program of graduate study leads to the Ph.D. degree and is normally completed in 4-5 years. The M.S. and the M.Phil degrees are granted to qualified students in the Ph.D. program who wish intermediate degrees. (See 3.1.6.)

3.1.1. A Brief Overview of the Doctoral Program

Here is how a Ph.D. student's career goes.

In the first year, the student is expected to take courses and familiarize himself or herself with the activities of the various research groups in the Department. Over the first two years, each student is required to take six courses that satisfy the Distribution Requirements enumerated n Section 3.1.2 below. In these constraints, we use a pattern notation to Indicate course classes. Every course has a three-digit number, Kdn. K must be greater then or equal to 5. n is arbitrary. The digit d is the most reliable Indicator of the subject area of the course.

After the first year, the student comes under the direction of a Supervisory Committee. This is a group of three or four faculty members who take responsibility for monitoring and mentoring the student's progress. The principal member of the committee is the student's advisor. Once a student has selected a Supervisory Committee, any changes to the Committee require consultation among the Director of Graduate Studies, the old Committee, and the proposed new one. Each student is expected to put together a supervisory team by the beginning of his or her third term, which means that there should be serious dialogue with the faculty starting well before the end of the second term.

In the second year, the student continues taking courses, completing a total of twelve courses by the end of the year. Two of these must be the CS-690 and CS-691 sequence, in which the student does a project under the direction of a faculty Advisor. The Advisor for the 690 Project should be chosen before the beginning of the second year. Each student must serve as a teaching assistant (TA) for two terms as a key part of academic training. First-year students are not normally eligible as TAs, so this requirement becomes relevant in the second year. However, It Is not necessary to satisfy the teaching requirement entirely in that year.

The end of the second year is the culmination of two years of study and research. The student passes an Area Exam demonstrating breadth of knowledge in the research area of the 690 Project, and finishes the 690 Project itself, presenting it to the Faculty in oral and written form.

By the start of the third year, the student begins dissertation research, usually under the direction of the same supervisory committee. However, if the student or the committee believes that the research is not going well, this is a good time to pick a new advisor and research topic.

By the end of the third year, the student should have satisfied all requirements for admission to candidacy, including writing a Thesis Prospectus that describes the general area and direction of the dissertation research. The student is then admitted to candidacy by vote of the Faculty. The Graduate School will not allow a student to register for a fourth year of study until this step has been completed.

After admission to candidacy, the student's position in the Department is secure, subject only to continued satisfactory progress toward completion of the Dissertation. When the Dissertation is complete, it is defended before the Faculty and approved by a Committee of Readers. The requirements for the Ph.D. have then been met and the degree is granted.

3.1.2. Requirements

The milestones along the way to the Ph.D. are described in detail below. The course requirements, examinations, and 690 Project should be completed by the end of the second year, and all requirements for admission to candidacy must be completed by the end of the third year. The Dissertation and Defense must be completed no later than the end of the sixth year. Exceptions to these rules require approval of the Director of Graduate Studies, the Faculty, and/or the Graduate School.

Course Requirements: Students are required to pass twelve courses,
satisfying the following constraints:

• Two of the courses must be CS-690 and CS-691, "Independent Project." These courses introduce the student to research under the guidance of the faculty advisor.

• Grade Requirement: CS-690 must be passed with a grade of SAT and CS-691 must be passed with a grade of HIGH PASS or HONORS.

Two of the remaining courses must be passed with a grade of HONORS, the rest must be passed with a grade of at least HIGH PASS.

• Depth requirement: By the end of the second year, each student must pass three advanced courses in a particular field. The student's Supervisory Committee will certify that this requirement has been
met.

The following rules govern which courses may be counted towards the course requirements:

CS Department courses:
• No course with number 499 or below can be used to satisfy a course requirement.

• Any 500-level course can be used to satisfy the 12-course requirement and the depth requirement. For example, if a theory student takes a 500-level theory course, it simultaneously counts toward both requirements.

• The CS-690 and CS-691 course sequence counts only
toward the 12-course and research requirements.

• A 600-, 700-, or 800-level course can count towards the 12-course and depth requirements, but only if it involves regular meetings with a faculty member and tangible written work that can be evaluated, and results in a regular grade of HONORS or HIGH PASS. In practice, some but not all 600-level courses will count, while 700- and 800-level courses normally will not count towards any requirements.

• Non-CS Department courses: A student can count a graduate-level course outside the Department towards the 12-course and depth requirements if it is relevant to the student's program of study. This determination is made by the Director of Graduate Studies, possibly
in consultation with the student's Advisor.

The constraints:
• Two of the six courses must be in theoretical computer science, whose numbers typically have d=6. Currently they include CS-557, CS-560, CS-561, CS564, CS-565, CS-567, and CS-569.

• Two are in programming languages and systems, whose numbers typically have d = 2 or 3. One of these courses must be "programming intensive," requiring participation in a multi-person project whose goal is produce several thousand lines of code. The programming-intensive courses currently are CS-521, CS-522, and CS-537. Other courses in this category are CS-524, CS-525, CS-529, CS-530, and CS-534.

• Two are in “applications,” typically having numbers with d = 4, 5, or 7. The list currently contains CS-540, CS-570, CS-572, CS-573, CS-575, CS-576, CS-577, CS-578, CS-579, CS-752.

• All the courses should contain substantial material beyond what the student has learned before coming to Yale. For instance, CS-570 ("Artificial Intelligence") is inappropriate for someone who took an AI course as an undergraduate.

• If a student is unable to find courses satisfying the requirements above, the DGS will usually accept courses from other departments that are in similar topic areas. Seminars may be acceptable, too. These are courses presented on an ad-hoc basis, that may consist mainly of paper presentation and discussion. (Their numbers usually have d=7.) The DGS will approve such a course to satisfy the distribution requirement if a student has already studied deeply in a research area, and is strongly motivated to explore it further.

Graduate Student

In order to keep the Faculty apprised of research progress, each student must give a public talk on the progress of his or her research twice during his career, once after completing a 690 Project, and once after filing a dissertation prospectus. These talks are known as Official Graduate Student Talks (OGSTs). They are open to the public, and all members of the department are welcome. The OGST must be scheduled in the first Fall term after being admitted to candidacy and writing a prospectus. The student should recruit one faculty member who is not on his or her supervisory committee or reader committee to attend the talk. If the student is unable to find such a person, the DGS will assign one.

The student must announce the talk at least a week in advance on the appropriate e-mail lists (currently grad-students@cs.yale.edu and faculty-cs@cs.yale.edu). The announcement should be repeated as the
date draws near. All OGST announcements should indicate the student’s year of study and the current 690 Project Advisor or Thesis Advisor. Students who have been admitted to candidacy should also list their reading committee and expected date of completion.

690 Project The student must submit a written report on his or her 690 Project to his or her Supervisory Committee, which grades it for (a) quality of the work, (b) quality of the technical writing, and (c) quality of the English. The grade and a one-page abstract must be transmitted to the Director of Graduate Studies.
Please note that the course grades for CS-690 and CS-691 are not the same as the grade for the 690 Report. The Advisor files a grade as of “SAT” or “UNSAT” for CS-690 and HONORS or HIGH Pass for CS-691, indicating whether the student is making satisfactory progress toward completing the research and the report. If not, the Supervisory Committee and Faculty should be notified.

Area Examination The student must pass an Area Examination by the end of the second year. The purpose of the Area Exam is to demonstrate proficiency in scholarship over a subject area that includes the area of the 690 Project, but is broader. The Exam is

formulated and administered by the student's Supervisory Committee. The Committee will decide whether the same exam should be given simultaneously to a cohort of students, as opposed to giving each student his or her own Exam.

The Exam typically includes either or both of the following:

• A written or oral test of in-depth knowledge.
• A test of the capacity to learn a topic from research literature (e.g., an extended oral presentation and critique of one or more research papers).

Thesis Advisor A regular faculty member must agree to direct the student's Dissertation, thereby certifying that the student is capable of doing original research. Meeting this requirement does not automatically follow from the student's receiving a grade of HONORS or HIGH PASS on the 690 Project. The advisor may be a ladder faculty member from another Yale department, if the student's Supervisory Committee and the DGS approve. It is generally unproductive for a student to attempt a dissertation in an area not covered by the Faculty's interests; faculty members will normally require a student to work in areas they care about. The Thesis Advisor must be chosen by the beginning of the third year.

Thesis Prospectus A Thesis Prospectus must be filed with the Director of Graduate Studies and the Graduate School, this being a written summary (about 3 or 4 pages long) of the nature and scope of the thesis research and a tentative title of the Dissertation. The Prospectus must also include a proposed Committee of Readers (see below) and be signed by the Advisor.

Admission to Candidacy The Faculty will vote to admit the student to Candidacy when all of the requirements described above have been satisfied: Course Requirements, 690 Project and Report, Area Examination, Thesis Advisor, and Thesis Prospectus. It is expected that a student will be admitted to Candidacy by the end of the third year. If the requirements have not been satisfied by the beginning of the fourth year, the graduate school will not allow the student to register for classes and the department will ask the student to withdraw.

The Dissertation The most important part of the Ph.D. program is research training, culminating in the writing of a Dissertation. The Dissertation should be concluded no later than the end of the student's sixth year. The Dissertation demonstrates the student's ability to perform original research. Thus, it must demonstrate technical mastery of the subject and must contain conclusions that modify or enlarge what has previously been known. Because Yale is a University, dedicated to the dissemination of knowledge, all results of research, including the Dissertation, must be made public. Access may not be restricted for any reason, commercial or governmental.

Thesis Defense The student must give an oral Defense of the thesis research when the student's Committee is satisfied that the work is complete and the student has a complete draft of the Dissertation ready to submit to the Graduate School. To ensure the latter, one

copy of the Dissertation must be given to the Departmental Registrar and made available to the Department Faculty at least one week before the Defense takes place.
The Defense consists of a one-hour public presentation of the results followed by a 15-minute question and discussion period, which is open to the entire Department and its guests. The Faculty and outside Readers then conduct an oral examination in closed session.

In order to give all interested faulty the opportunity to attend, the defense must be scheduled with the Director of Graduate Studies and announced to the Faculty at least one month in advance.

Dissertation Submission The Dissertation should be submitted to the Graduate School as soon as the Thesis Defense has been passed and any final corrections to the Dissertation have been made. This must be completed within one month of passing the Defense, or the student must defend again. A copy of the final draft must also be given to the Departmental Registrar. After the Dissertation is submitted, copies are sent to the members of the Reading Committee (see below), who each read the thesis and complete a Reader's Report form. When all Reader's Reports are in, they are made available to the Faculty, who then vote to recommend the degree at a special faculty meeting. The recommendation is then forwarded along with the Reader's Reports to the Graduate School, which reviews the recommendation. Finally, the entire Graduate Faculty votes to approve the degree.

In order to allow time for these steps to be completed in a timely fashion, the Graduate School requires that the Dissertation be submitted by October 1 for a December degree and by March 15 for a May degree. While these deadlines have not always been strictly enforced in the past, the Department and the Graduate School will feel under no obligation to complete the degree approval process in time for graduation if they are missed, and the actual award of the degree may be delayed half a year as a result.

Dissertation Readers The Dissertation must be read by a Committee of four Readers, which is a distinct entity from the Supervisory Committee (although it normally overlaps with it). Three Readers must be Internal
and one must be External. An Internal Reader may be any faculty-level person with a close affiliation to the Yale Department of Computer Science, including regular faculty, visiting faculty, research scientists, and associate research scientists. An External Reader may be any qualified person who is not closely affiliated with the Yale Department of Computer Science. In addition, the Reading Committee must conform to the following rules:

• At least two Internal Readers must be regular ladder faculty in the Yale Department of Computer Science;

• All Internal Readers are normally expected to attend the student's twice-a-year public talks and the Thesis Defense;

• At least three Readers must attend the Thesis Defense.

Exceptions to these rules require approval of the Director of Graduate Studies.

For the purposes of these regulations, “close affiliation” status is conferred by any extended visit in the Department or any kind of Departmental appointment or title, including affiliate and adjunct titles. Occasional short-term visits or research collaborations do not constitute close affiliation. Once conferred, the status of “close affiliation” persists for a period of two years after the affiliation terminates. Thus, a faculty member who takes a position elsewhere may continue to serve as an Internal Reader for two years after leaving and may not serve as an External Reader during that same period. The above notwithstanding, the Reading Committee must always include at least one current regular ladder faculty member in the Yale Computer Science Department. In addition, if the Advisor leaves Yale, the Graduate School may require that a current Yale faculty member serve as Acting Advisor.

The rules concerning the composition of the Reading Committee must be satisfied when the Committee is first formed, at the time of the Thesis Defense, and at any time that the Committee is changed.

3.1.3 Evaluation of Progress

Students must maintain a satisfactory rate of progress toward the Ph.D. in order to remain in good standing in the program. During the first year, progress is measured by formal course work. To remain in good standing, at least six courses must be completed with a grade of HIGH PASS or better.

After the first year, rate of progress is monitored by the student’s Supervisory Committee. The Committee looks at grade records, exam results, the 690 Report, and research progress. The Committee is also expected to attend the student’s semi-annual GST’s in order to see first-hand how the student is doing. Students beyond the first year receive written annual evaluations of their progress, drafted by the Supervisory Committee. A copy of this evaluation is placed in the student’s file. A decision that the student is not making satisfactory progress toward the Ph.D. may be made at any time by the Supervisory Committee.

Whenever a student is determined not to be in good standing, either by failing to achieve required milestones or by recommendation of his or her Supervisory Committee, the student and the Faculty will be notified. All information regarding the student, including course grades, research performance, and performance on exams, will be made available to the Faculty as a whole, which will then determine a course of action for the student. Possibilities at this stage can include continuation in the program with revised expectations, academic probation, or dismissal from the graduate program. The director of Graduate Studies will inform the student in writing of the Faculty’s determination and, in case continuation in the program is permitted, of conditions that must be fulfilled to return or remain in good standing.

In cases where the sole reason for the student's trouble is apparent inability to do research under the supervision of his or her current Committee, the usual expectation is that a new Committee will be formed and will give him or her an appropriate period of time (a term or a summer) to demonstrate ability to conduct a research project successfully. The Committee will report to the Faculty at the end of this period, so that a new decision can be made.

If the Committee determines that the student has not yet passed one of the designated requirements, then the Committee should report at that time, and as necessary in subsequent terms, on how the student is progressing towards satisfying the requirements, and what its recommendation is. The recommendation can range from "the student should be terminated" to "the student has satisfied all the requirements for admission to Candidacy." One possible recommendation is that the student change research area, under the direction of a new Supervisory
Committee. This recommendation is not routine, and should not be considered the normal consequence of failing a Area Exam.

The Supervisory Committee's evaluation is particularly crucial at the end of the second year, when the results of the 690 Project and Area Exam become available. At this time, the Supervisory Committee is expected to report in writing to the Faculty as a whole (as well as to the student) on the student's status. This notification should be given by the middle of May, and a special faculty meeting will be held toward the end of May to act on any recommendations.

3.1.4. Miscellany

In order to gain teaching experience, all graduate students are required to serve as a teaching assistant for two terms during their first three years of study. Teaching performed in order to meet the obligations of financial aid packages can also be used to satisfy this requirement. Students who perform teaching not required by a financial aid package may receive additional compensation. (See 6.2)

The Graduate School requires that a Ph.D. student spends a minimum of three years in residence and that full tuition be paid for four years. If the student graduates in fewer than four years, with no leaves of absence, then any additional tuition is waived.

Whether a student is in good standing is independent of whether there are funds to support him or her.

If a student's Advisor leaves Yale, then what happens depends on the student's state of progress toward a Ph.D. A student who has not completed the three-year residency requirement and been admitted to Candidacy will normally be expected to find a new Advisor or go with the departing faculty member and enroll in another Ph.D. program. An advanced student normally finishes his or her Dissertation while continuing under the technical supervision of the departed Advisor and receives a Yale degree. In this case, the Graduate School may require that a current Yale faculty member agree to act as

official Advisor. Such a student will have two years to finish his or her Dissertation before the Department will no longer be bound to accept it. The Thesis Defense must still be held at Yale, according to the usual rules.

Students can expect to have office space in Arthur K.Watson Hall, subject to availability, for their first six years.

3.1.5. The Fast Track

Fast-track status enables students whose Computer Science education is already well under way when they enter the Ph.D. program (e.g., after receiving a Master's Degree in Computer Science from another institution) to take fewer courses and to get started sooner on research.

A student who wants to get onto the fast track must discuss the issue with the DGS upon admission to the program. The status becomes official if, by the end of the first year of study, the student has taken CS690 (i.e., begun research), passed an Area Exam, and passed six courses with grades of HIGH PASS or HONORS. The DGS will examine the student's academic history at that point to decide how many more courses, if any, must be taken in order to satisfy the distribution requirements.

Students who expect to qualify as "fast track" may, with permission of the Director of Graduate Studies, begin the 690 project in the first or second term of study. It may also be granted in those cases where the intended Area Examination covers work done for the 690 Project (which is now the case in Programming Languages and Systems). Such an early start on research will not affect the eventual attainment of fast-track status nor the number of courses that will be waived, both of which are determined as described above.

3.1.6. Master's Degrees en Route to the Ph.D.

A student in the Doctoral Program can earn a Master of Science (M.S.) degree and/or a Master of Philosophy (M.Phil.) degree en route to the Ph.D. The requirements for the M.S. degree are described in 3.2 below. The requirements for the Master of Philosophy (M.Phil.) degree are the same as for the Ph.D. except for requirements having to do with the Dissertation.

3.2 The Master's Program

The terminal Master's Program of graduate study is normally completed in one year, but a part-time program may be spread over as long as four years. To qualify for the Master of Science degree, the student must pass eight courses at the 500-level or above from an approved list. An average grade of at least HIGH PASS is required, with at least one grade of HONORS.

A one-term Independent Project course (CS-692) may be applied towards the Master's degree with prior permission of the Director of Graduate Studies, provided that a faculty member is willing to supervise the project. Faculty are under no obligation to supervise independent projects for Master's students, and the expectation should be that a Master's student will not do an Independent Project.

Advanced graduate courses in other departments that involve concepts from computer science and are particularly relevant to an individual program may, with permission of the Director of Graduate Studies, be counted towards the degree. Generally at most two such courses may be used to satisfy the requirements of the Master's Program. Here an advanced course is generally one with at least one intermediate course as a prerequisite and an intermediate course is generally one with at least one (introductory) course as a prerequisite. But five courses must be in Computer Science.

4. Departmental Computing Facilities

The faculty, researchers, and students in the Department of Computer Science have access to a wide variety of ever-changing state-of-the-art computing resources, ranging from laptops, conventional PC's and scientific workstations to high-powered compute-servers and workstation clusters used as parallel computers. All of the computer systems are interconnected by a switched Ethernet local network, which is connected to the Internet via fiber optic technology to the campus backbone. Also recently installed is a wireless network permitting instant laptop access to the Internet anywhere in the building and many places on campus.

Most faculty, Ph.D. students, and researchers are equipped with a personal workstation or advanced-technology PC, all running some variant of the Linux or Windows operating systems. The computing needs of undergraduate and master's students are met through the "Zoo", an educational laboratory with approximately 40 dual-processor Linux workstations which allow for remote as well as on-site access.

Individual research groups have additional specialized equipment for robotics, computer vision, computer music, networking, and other research efforts. There are a number of parallel and distributed computing testbeds, including a Silicon 1Graphics Origin 2000, a DEC Alphaserver 4100, and an IBM Beowulf Netfinity 1350 cluster consisting of 14 dual processors and a half terabyte data repository. Students in computer science, both graduate and undergraduate, have liberal access to all of these facilities. In this way students play a vital role in contributing to our understanding of theoretical and experimental issues in computer science. The Department's computing resources are professionally managed by Workstation Support Services (WSS), a unit of the Yale office of Information and Technology Services (ITS). WSS staff follow policy set by a faculty oversight committee in providing first-class responsive service to all departmental users.

5. Graduate Student Life

5.1 The McDougal Graduate Student Center

Much of the graduate student life is based in the various departments and in dormitories or apartment complexes. The new McDougal Center is a place where graduate students from across the campus regularly meet and share interests. It is located in the Hall of Graduate Studies (HGS), 320 York Street (432-2583), mcdougal.center@yale.edu, http://www.yale.edu/mcdougal.

5.1.1 Mission

A generous gift from Mr. Alfred McDougal, a Yale alumnus, and his wife, Ms. Nancy Lauter, enabled Yale to create the McDougal Graduate Student Center in 1997. The McDougal Center provides space and program funding for building intellectual, cultural, and social life, and for facilitating professional development activities across the departments of the Graduate School of Arts and Sciences. The McDougal Center warmly welcomes the participation of students from other Yale Graduate and Professional Schools, postdoctoral fellows, faculty, staff, alumni/ of the Graduate School, and members of the larger Yale community. Its website (http://www.yale.edu/mcdougal) provides all kinds of information relating to graduate student life. The Center provides members of the graduate student community with a place of their own on campus.

5.1.2 Facilities

The facilities of the McDougal Center enhance student life in many ways. The magnificently restored Common Room has been transformed into a lounge with comfortable furnishings, internet ports, newspapers and magazines, and a student- run café serving coffee and light food throughout the day. In an adjacent wing on the first floor of HGS the Center has a large multi-purpose Program Room (HGS 119) with a portable stage, seating for up to 100, and advanced video and sound projection equipment. The Program Room provides space for lectures, conferences, performances, film series, workshops and other events by and for students. The Center also has smaller conference and meeting rooms. Graduate student groups and departments may request to reserve space by contacting the center office at 432-8273, stopping by HGS 123, or filling out a request on line at www.yale.edu/mcdougal/rooms.htm. There is a public computer cluster supported by Academic Computing Services, a public copy machine, a public phone, bulletin boards and information kiosks as well. The lower floor also offers offices for the Assembly of Graduate Students, graduate student organizations, rooms for Teaching Fellows to meet with students, lockers for graduate student use and vending machines. The McDougal Center is open days, evenings, and weekends.

5.1.3 Student Life Programs

The Center offers a variety of activities open to the Graduate and Professional community. These include weekly movies on the Really Big Screen, coffeehouse musical evenings, happy hours, poetry readings, students' research presentations, health and wellness workshops, teas with campus and community figures, and service opportunities such as blood drives. It hosts activities organized by student groups and departments, including cultural festivals, movies, lectures, receptions, and conferences. Activities are publicized in campus publications, in McDougal Notes calendar, on the website, and via email lists. Find out what’s going on at your Center today! Lisa Brandes, Director, Graduate Student Life, 123 HGS, 432-2583.

5.1.4 Graduate Career Services (GCS)

Graduate Career Services was established to guide and educate graduate students about academic and non-academic career opportunities and job search strategies. The office offers programs such as professional career development workshops, seminars, resume/CV reviews, individual counseling, on-campus interviews, dossier service and current job listings. Mary Johnson, 123 HGS, 432-2583 .

5.1.5 The Office of Teacher Preparation

The Office of Teaching Fellow Preparation and Development provides a wide variety of services and resources for graduate student teachers at Yale. In addition to coordinating teacher training and teaching resources in the graduate school, the office serves to facilitate departmental and faculty involvement in the development of Yale's Teaching Fellows. Contact William Rando, Director, 203.432.2583, 123 HGS.

5.2 Life in the Department

The Department of Computer Science at Yale is a stimulating environment in which new ideas, experimental designs, and concrete artifacts are plentiful. In trying to shape the very nature of computer science, it is not enough to ask why things are, nor to ask how things will be — but rather, to ask how things should be now and in the future. How should computers be used in our society, and why? How should we design software, algorithms, new theories of computation? How should computer science be taught? What should the legacy of our efforts be? Department Colloquium Series (in which distinguished researchers from other universities are invited on a monthly basis to speak to a general CS audience) Theory Seminar (hosting talks on all aspects of theoretical computer science) SPAM (the "Systems Personal Activity Meeting", hosting talks on programming languages and systems, Vision Lunch (hosting talks on topics related to computer vision), CVC Round Table (open to any CS, EE, or Biomedical Engineering students, and focusing on computer vision, A.I., control theory, and biomedical engineering).

5.3 Life About Town

Yale is the focal point for much of the intellectual and cultural life of New Haven. Yale offers two symphony orchestras, a symphonic wind ensemble, a jazz ensemble, the Yale Repertory Theater, the Yale Art Gallery, the British Art Center, and more than a thousand informal concerts, recitals, and theatrical productions each year. Many of these events are presented by undergraduate members of Yale College; others are presented by the Schools of Fine Art, Drama, and Music.

Beyond the campus is a small Yankee town of 136,000. Birthplace of the cotton gin, the modern telephone exchange, and pizza, New Haven dates back to 1638. In the midst of a busy urban center, several areas of the city still retain the atmosphere of earlier days. Several clubs in the area feature jazz and rock bands. Late-night coffee houses near campus allow you to sit for hours over a cup of the best espresso south of Boston. Nearby is a 24-hour bookstore, a haven for fantasy and alternative literature enthusiasts. There are many movie theaters in the area, several featuring art films and retrospective shows.

Indeed, New Haven has a rich cultural life independent of that provided by the University. There is an excellent resident theater company, the Long Wharf Theatre, which produces plays from the standard repertoire and one or two new works each season. The historic Shubert Theater and the Palace present a wide selection of musical theater and drama. New Haven also has its own professional symphony orchestra, chamber ensembles, and a small ballet company. The town is also host of the widely acclaimed International Festival of Arts and Ideas. Every June, world-class theater, film, and dance productions, art and photography exhibits, panel discussions and poetry readings, and many musical events turn the city into a cultural and intellectual Mecca.

New Haven boasts a wide variety of culinary establishments, from the mundane to the exotic. Available at just about any hour is the “sub” sandwich and pizza, but a variety of other fare is also available at restaurants within walking distance of the central campus: Italian, Chinese, Mexican, Japanese, Thai, Indian, Cuban, and “natural".

For outdoor and sports enthusiasts, New Haven boasts professional hockey and baseball teams, the Connecticut Tennis Center (host of the annual Pilot Pen Tennis Tournament), and over 800 acres of beautiful trails and fields at nearby East and West Rock Parks for jogging and biking enjoyment. Yale’s famous Payne Whitney Gymnasium is open to students at no charge during the academic year and for a nominal fee in the summer. Students also have the opportunity to participate in numerous intramural sports activities during the year as well as individual sports activities such as golf, tennis, and figure skating. Sailing, rowing, and canoeing are also available at Yale facilities in nearby towns.

And of course New Haven, Connecticut is part of New England, and is thus in proximity to all of New England’s great resources; from its quaint towns, beautiful beaches and seaports to its mountain peaks and lakes for hiking, swimming, skiing, and mountaineering. It is also easy to reach the big city experience: New Haven is only 75 miles from New York and 100 miles from Boston, and both are connected by frequent and convenient train service.

6. Graduate Admissions and Financial Aid

6.1 Admissions Procedures

Students are admitted for entrance in the Fall term only. An applicant should have strong preparation in mathematics, engineering, or science. He or she should be competent in programming but needs no computer science beyond the basic level. The Graduate Record Examination Aptitude Test and some pertinent Advanced Achievement Tests are required (GRE General and Subject).

Application for admission in the Fall of 2007 should begin in the Fall of 2006. Forms may be obtained from:

http://www.yale.edu/graduateschool/admissions

Contact Information for the Graduate School is:
Graduate School Admissions
Yale University
105 Hall of Graduate Studies
320 York Street
New Haven, Connecticut 06511

Telephone: 203.432.2771

Applications to the Graduate School is an online process only.

Note that the Graduate School does not accept faxed copies of letters of recommendations, transcripts or other supplemental material.

Prospective students can obtain further Information by sending email to **graduate.admissions@yale.edu**
Also, the Computer Science website has very useful information for prospective students; **http://www.cs.yale.edu/education/faq.html**

The deadline for completed applications, including all letters of recommendation and test scores, is **December 15, 2006.**

Applicants will be notified of action concerning admission as soon as the decision has been made, generally between March 15 and April 1. Those who are undergraduates at the time of admission must present evidence of having satisfactorily completed the Bachelor's degree or its equivalent in order to register. Those who are in graduate school must present transcripts giving evidence of satisfactory completion of the current year's work prior to registration.

There is a non-refundable application fee of $85. Applicants from countries under exchange restrictions should seek the help of their state banks or of friends already in the United States for payment of this fee.

Applicants should arrange to take the GRE's no later than October testing.The results of later testing are usually not available before admissions decisions are made. Remember that ETS will report scores only by mail and only at the written request of the student. Address inquiries to:

GRE-ETS
P.O. Box 6000
Princeton, NJ 08541-7670
609.771.7650
www.ets.org

Except by prearrangement with the Dean's office, foreign applicants whose native language is not English must present evidence of proficiency in English by satisfactorily completing the Test of English as a Foreign Language, administered in foreign countries by the Educational Testing Service. TOEFL scores must be received by December 15th. Address inquiries to:

TOEFL/TSE Services – ETS GRE-ETS
P.O. Box 6000
Princeton, NJ 08541-7670
609.771.7670
www.toefl.org

6.2 Financial Aid Policy

The Department tries to find financial support for every student who needs it, for at least the first four years of study. During the first year, support is usually provided by a University Fellowship.

After the first year, students often receive research assistantships in their field of specialization or other forms of support. These may also be supplemented by teaching fellowships. The standard teaching fellowship in this department is at the level of a Teaching Fellow 2 and requires approximately 10 hours of work per week. Advanced students are sometimes supported through a combination of a University Fellowship and a teaching fellowship.

Only in exceptional cases is financial aid available after the fourth year. However, the lack of support is mitigated by the fact that a nominal "Continuous Registration Fee" (currently $256.00 per term) replaces the tuition requirement after the fourth year. (See the Graduate School "Programs and Policies" book for more information on the tuition policy.)

Applicants seeking financial aid must complete the forms included in the application packet sent out by the Graduate School.

7. Graduate Courses

CPSC 521a, *Compilers and Interpreters*. Zhong Shao
MWF 1:00-2:15
Compiler organization and implementation: lexical analysis, formal syntax specification, parsing techniques, execution environment, storage management, code generation and optimization, procedure linkage, and address binding. The effect of language-design decision on compiler construction.

CPSC522b, *Operating Systems*. James Aspnes
MW 1:00-2:15
The design and implementation of operating systems. Topics include synchronization, deadlock, process management, storage management, file systems, security, protection, and networking.

CPSC 524b, *Parallel Programming Technique.* David Gelernter
TTh 1:00-2:15
Software structures, architectures, and algorithms for parallel and distributed applications, focusing on coordination frameworks for asynchronous concurrency (on the code that creates and manages multiple processes and performs the interprocess communication necessary to create integrated ensembles). Coordination languages and program-development environments. The fast-changing WAN-software picture. Parallel and distributed programming exercises on LANs. (Taught in alternate years.)

CPSC 525b, *Theory of Distributed Systems.*
(not taught in 2006-2007)
Models of asynchronous distributed computing systems. Fundamental concepts of concurrency and synchronization, communication, reliability, topological and geometric constraints, time and space complexity, and distributed algorithms. (Taught in alternate years.)

CPSC 528b, *Language-Based Security.*
(not taught in 2006-2007)
Basic design and implementation of language-based approaches for increasing the security and reliability of systems software. Topics include proof-carrying code, certifying compilation, typed assembly languages, runtime checking and monitoring, high-confidence embedded systems and drivers, and language support for verification of safety and liveness properties.

CPSC 529a, *Functional Programming.*
(not taught in 2006-2007)
Methods for synthesizing functional programs from formal specifications and verifying correctness properties of programs. Topics include higher order functions, pattern-matching, abstract algebraic datatypes, polymorphic types, advanced typing issues such as type classes and higher-order modules, lazy/eager evaluation, equational reasoning, and realization of effects via continuation and monads. The functional languages Haskell and/or ML are used in the course. (Not taught every year.)

CPSC 530a, *Formal Semantics.*
(Not taught in 2006-2007)
Introduction to formal approaches to programming language design and implementation. Topics include the lambda-calculus, type theory, denotational semantics, type-directed compilation, higher-order modules, and application of formal methods to systems software and Internet programming. (Taught in alternate years).

CPSC 531a, *Fundamentals of Computer Music*. Paul Hudak
TTh 1:00-2:15
Introduction to formal approaches to programming language design and implementation. Topics include the lambda-calculus, type theory, denotational semantics, type-directed

compilation, higher-order modules, and application of formal methods to systems software and Internet programming. (Taught in alternate years.)

CPSC 532a, *Graphs and Networks*. Daniel Spielman
TTH 2:30-3:45
A Mathematical examination of graphs and their applications in the sciences. Families of graphs to be studied include social networks, small-world graphs, Internet graphs, planar graphs, well shaped meshes, power-law graphs and classic random graphs. Phenomena examined include connectivity, clustering, communication, ranking and iterative process. (Not taught every year.)

CPSC533b, *Computer Networks.*
(not taught in 2006-2007)
An introduction to computer networks with emphasis on the Internet. Topics include network and protocol architectures; communication and switching techniques; link layer and local area networks; performance analysis; network layer and routing; multimedia and Integrated services; flow and congestion control; and network security. (Not taught every year.)

CPSC 534a, *Mobile Computing & Wireless Networking*. Richard Yang
MW 2:30-3:45
An introduction to the principles of Mobile Computing and its enabling technologies. Topics include: principles of mobile computing, wireless systems, information management, location-Independent and location-dependent computing models, disconnected and weakly-connected operation models, human-computer interactions, mobile applications and services, security, power management, and sensor networks. (Not taught every year.)

CPSC 536a, *Networked Embedded Systems and Sensor Networks*. Andreas Savvides
TTH 2:30-3:45
Introduction to the fundamental concepts of networked embedded systems and wireless sensor networks, presenting a cross-disciplinary approach to the design and implementation of smart wireless embedded systems. Topics include embedded systems programming concepts, low-power and power-aware design, radio technologies, communication protocols for ubiquitous computing systems, and mathematical foundations of sensor behavior. Laboratory work includes programming assignments on low-power wireless devices.

CPSC 537a, *Introduction to Databases*. Avi Silberschatz
TTH 2:30-3:45
An introduction to database systems. Data modeling. The relational model and the SQL query language. Relational database design, integrity constraints, functional dependencies, and normal forms. Object-oriented databases. Implementation of databases: file structures, indexing, query processing, transactions, concurrency control, recovery systems, and security. (not taught every year.)

CPSC 539b, / E&AS907b, *Computer Systems*. Andreas Savvides
TTH 2:30-3:45, lab HTBA
The organization of computer systems as hardware and software systems. Instruction-set architecture, assembly programming, computer arithmetic, datapath architecture and control, pipelining, memory hierarchy. Concepts illustrated by exploration of an instructional RISC microprocessor.

CPSC 540b, *Numerical Computation I.* Vladimir Rokhlin
MW 2:30-3:45
Algorithms for numerical problems in the physical, biological, and social sciences: solution of linear and nonlinear systems of equations, interpolation and approximation of functions, numerical differentiation and integration, optimization.

CPSC545b, *Introduction to Data Mining*. Martin Schultz and Mark Gerstein
MW 2:30-3:45
This course studies algorithms and systems that allow computers to find patterns and regularities In databases, perform prediction and forecasting, and generally improve their performance through interaction with data.

CPSC 552a, *Genomics and Bioinformatics.* Dieter Soll, Mark Gerstein and Michael Snyder
MW 1:00-2:15
Genomics describes the determination of the nucleotide sequence as well as many further analyses used to discover functional and structural gene information about all the genes of an organism. Topics include the methods and results of analysis on a genome-wide scale as well as a discussion of the implications of this research. Bioinformatics describes the computational analysis of gene sequences and protein structures on a large scale. Topics include sequence alignment, biological database design, geometric analysis of protein structure, and macromolecular simulation.

CPSC555b, *Economics and Computation.*
(Not taught in 2006-2007)
A mathematically rigorous Investigation of the interplay of economic theory and computer science with an emphasis on the relationship of incentive-compatibility and algorithmic efficiency. Particular attention will be paid to the formulation and solution of mechanism-design problems that are relevant to data networking and Internet-based commerce. Suitable for advanced undergraduates and beginning graduate students. Familiarity with basic microeconomics and game theory is desirable but not required. (Not taught every year.)

CPSC 557b, *Sensitive Info in a Wired World.*
(not taught in 2006-2007)
Research-oriented course that addresses issues of ownership, control, privacy, and accuracy of the huge amount of sensitive information about people and organizations

that is collected, stored, and used by today's ubiquitous information systems. Students will read research papers that explore both the power and the limitations of existing privacy-enhancing technologies such as encryption and "trusted platforms." Suitable for advanced undergraduates and first-and second-year graduate students in Computer Science and computationally sophisticated students in related fields. (Not taught every year.)

CPSC 560b, *Theoretical Methods in Computer Science*. Joan Feigenbaum
(not taught in 2006-2007)
An introduction to the main areas of theoretical computer with the goal of providing the theoretical background for doing research in computer science. Topics to be covered are:
• Complexity theory: review of machine models (Turning and RAM machines), basic complexity classes (polynomiality, nondeterminism, randomization, parallel models), measures of complexity (computational, communicational, informational).
• Algorithms and their analysis (fundamental algorithms in graph theory, number theory, sorting, and searching).
• Data structures and their role in the efficient implementation of algorithms.

CPSC 561b, *Foundations of Cryptography.*
(Not Taught in 2006-2007)
Foundations of modern cryptography and their application to computer and network security. Topics include randomized models of computation, indistinguishability, computationally hard problems, one-way and trapdoor functions, pseudorandom generators, zero-knowledge, secure computation, and probabilistic proofs. (Not taught every year.)

CPSC563a, *Introduction to Machine Learning.*
(not taught in 2006-2007)
Paradigms and algorithms for learning classification rules and more complex behaviors from examples and other kinds of data. Topics may include version spaces, decision trees, artificial neural networks, Bayesian networks, instance based learning, genetic algorithms, reinforcement learning, inductive logic programming, the MDL principle, the PAC model, VC dimension, sample bounds, boosting, support vector machines, queries, grammatical inference, and transductive and inductive inference.

CPSC 564b, *Quantum Computing.*
(not Taught in 2006-2007)
A tutorial introduction to quantum mechanics and computer science will be given in the context of quantum computation. Hardware (quantum gates and data representation) and algorithms (the quantum Fourier Transform, the Shor factorization algorithm and the Grover search algorithm) will be described. Topics for research will be discussed.

CPSC 565a, / AMTH511a, *Topics In Algorithms.*
(not taught in 2006-2007)
Introduction to the fundamental tools used in approximation algorithms: linear, convex and semi-definite programming; dynamic programming; and geometric tools. Recent progress in the design of approximation algorithms for graph problems, combinatorial problems, and other NP-hard optimization problems. Results on the hardness of approximation based on probabilistically checkable proofs. (Taught in alternate years.)

CPSC 566b, *Combinatorics.*
(Not Taught in 2006-2007)
This course will present basic results and methods of combinatorics. Among the topics are: extremal combinatorics, enumerative combinatorics and generating functions; basic objects: graphs, partially ordered sets, hypergraphs, partitions, tableaux, basic methods, linear-algebraic and spectral, probabilistics and relations with Computer Science.

CPSC 567a, *Cryptography and Computer Security*. Michael Fischer
TTh 1:00-2:15
A survey of such private and public key cryptographic techniques as DES, RSA, and zero-knowledge proofs, and their application to problems of maintaining privacy and security in computer networks. The main focus is on technology, but the course will also consider such societal issues as balancing individual privacy concerns against the needs of law enforcement, vulnerability of societal institutions to electronic attack, export regulations and international competitiveness, and development of secure information systems.

CPSC 569b, *Randomized Algorithms.* Ravindran Kannan
TTh 2:30-3:45
Beginning with an introduction to tools from probability theory including some inequalities like Chernoff bounds, the course covers randomized algorithms from several areas; graph algorithms, algorithms in algebra, approximate counting, probabilistically checkable proofs, and matrix algorithms. (Taught in alternate years.)

CPSC 570a, *Artificial Intelligence.* Drew McDermott
MWF 2:30-3:20
An introduction to artificial intelligence research, focusing on reasoning and perception. Topics include knowledge representation, predicate calculus, temporal reasoning, vision, robotics, planning, and learning.

CPSC 572a, *Techniques for Symbolic Programming.*
(Not taught in 2006-2007)
The LISP programming language: review of basics; packages; macros. Object-oriented techniques; use of procedures as data; backtracking and similar control structures; knowledge representation. Predicate calculus: unification; backward chaining; logic programming; indexing techniques; numerical algorithms; planning algorithms; GUIs. Interfacing to other programming languages.

CPSC 573b, *Intelligent Robotics*. Brian Scassellati
MWF 10:30-11:20
An introduction to the basic principles of building a purposeful autonomous robotics team with an emphasis on human-machine interaction and cognitive modeling. (Not taught every year)

CPSC 574b, *Autonomous Systems.*
(Not Taught in 2006-2007)
The goal of this course is to acquaint students with the basic principles of building a purposeful autonomous robotic system. Lectures will cover the theory and practice of control systems, sensors, representation of the environment, and planning. Students will construct a simulated autonomous system, and also be given the opportunity to work with a real mobile robot. (Taught in alternate years.)

CPSC 575b, *Computational Vision and Biological Perception.* Steven Zucker
MW 1:00-2:15
We provide an overview of computational vision with a biological emphasis. The course is suitable as an introduction to biological perception for computer science and engineering students, as well as an introduction to computational vision for mathematics, psychology, and physiology students.

CPSC 576b, / E&AS 914b, *Computer Vision.*
(Not taught in 2006-2007)
Computational accounts of visual perception: image formation, image transformations, line and curve extraction, segmentation, shape, stereo, motion, texture, and model-based object recognition. A review of relevant mathematical tools, algorithms, and results from studies of human vision.

CPSC 577a, *Neural Networks for Computing.* Willard Miranker
TTh 11:30-12:45
Artificial neural networks as a computational paradigm studied with application to problems in associative memory, learning, pattern recognition, perception, robotics, and other areas. Models for the dynamics of neurons and methods such as learning for designing neural networks will be developed. Concepts, designs, and methods compared and tested in software simulation. Brain and consciousness studies are optional topics. Programming required (Not taught every year).

CPSC 578b, *Computer Graphics*. Julie Dorsey
TTh 4:00-5:15
An introduction to the basic concepts of two and three dimensional computer graphics. Topics include affine and projective transformations, clippings and windowing, visual perception, scene modeling and animation, algorithms for visible surface determination, reflection models, illumination algorithms, and color theory. Assumes solid C or C++ programming skills and a basic knowledge of calculus and linear algebra.

CPSC 579b, *Advanced Topics in Computer Graphics.*
(Not taught in 2006-2007)
An in-depth study of an active research area in computer graphics. Topic changes with each offering. Recent topics include rendering techniques, appearance modeling, and architectural modeling. Readings and a project. May be repeated for credit.

CPSC 586b, *Computational Complexity.* Joan Feigenbaum
TTH 1:00-2:15
Introduction to the theory of computational complexity. Basic complexity classes, including Polynomial Time, Nondeterministic Polynomial Time, Probabilistic Polynomial Time, Polynomial Space, Logarithmic Space, and Nondetermininistic Logarithmic Space. The roles of reductions, completeness, randomness, and Interaction. In the formal study of computation. (Not taught every year.)

CPSC 690a or b, *Independent Project I.* By arrangement with faculty.
Individual research for students in the Ph.D. Program. Requires a faculty supervisor and the permission of the Director of Graduate Studies.

CPSC 691a or b, Independent Project II. By arrangement with faculty.
Continuation of CPSC 690a or b.

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

CPSC 820a or b, *Directed Readings in Programming Languages and Systems.*
By arrangement with faculty.

CPSC840a or b, *Directed Readings in Numerical Analysis.*
By arrangement with faculty.

CPSC 860 a or b, *Directed Readings in Theory.*
By arrangement with faculty.

CPSC 870 a or b, *Directed Readings in Artificial Intelligence.*
By arrangement with faculty.

Related Courses in Other Departments

E&AS 875a, *Introduction to VLSI System Design.*

E&AS 912a, *Digital Image Processing and Analysis.*

ENAS 507b, *Digital Systems Testing and Design for Testability.*

LING 541b, *Language and Computation.*

E&AS 600a, *Computer-Aided Engineering.*

STAT 541a, *Probability Theory.*

STAT 542b, *Theory of Statistics.*

STAT 551b, *Stochastic Processes.*

STAT 664b, *Information Theory.*

8. Personnel

8.1 Advanced Students

Hai Fang
RESEARCH AREA: Design & Implementation of Scalable Certifying Compiler Infrastructure

David Goldenberg
RESEARCH AREA: Localization in Sensor Networks

Liwen Huang
RESEARCH AREA: Robot Programming & Functional Reactive Programming

Gang Li
RESEARCH AREA: Toward a Differential Geometrical Model for Binocular Stereo Vision

Zheng Ma
RESEARCH AREA: Resource Management in Overlay Networks

Zhaozhong Ni
RESEARCH AREA: Modular Machine Code Verification

Adam Poswolsky
RESEARCH AREA: Programming with Dependently Typed Logical Frameworks

Peishen Qi
RESEARCH AREA: Semantic TupleSpace Coordination Model for Web Services

Raphael Ryger
RESEARCH AREA: Practical Privacy in Network Computations

Jatin Shah
RESEARCH AREA: Formal Framework for Representing NP Complete Problems

Andrew Smith
RESEARCH AREA: Toward a General Framework, Integration of Databases, and Tools for Genomics Domain Specific Datamining

8.2 Recent Graduates
Jiang Chen, Associate Research Scientist, Columbia University
Thesis: *Learning via Queries*, 2006

Aleksandr Yampolskiy, Sr. Technical Staff, Oracle Corporation
Thesis: *Efficient Cryptographic Tools for Secure Distributed Computing*, 2006

Patrick Huggins
Thesis: *Sparse Coding via Geometry*, 2005

Vijay Ramachandran, Post-Doc, Stevens Institute of Technology
Thesis: *Foundations of Inter-Domain Routing*, 2005

Jian Zhang, Research Scientist, SRI International
Thesis: *Massive Data Streams in Graph Theory and Computational Geometry*, 2005

Antony Courtney, Gabis Connections, Inc.
Thesis: *Modeling User Interfaces in a Functional Language*, 2004

Ohad Ben-Shahar, Assistant Professor, Ben-Gurion University
Thesis: *The Perceptual Organization of Visual Flows*, 2003

Stefan Monnier, Assistant Professor, University of Montreal
Thesis: *Principled Compilation and Scavenging*, 2003

Rahul Sami, Post-Doc, MIT
Thesis: *Distributed Algorithmic Mechanism Design*, 2003

Gauri Shah, IBM, Almaden
Thesis: *Distributed Data Structures for Peer-to-Peer Systems*, 2003

Petros Drineas, Assistant Professor, Rensselaer Polytechnic Institute
Thesis: *Randomized Algorithms for Matrix Operations*, 2003

Christopher League, Assistant Professor, Long Island University
Thesis: *A Type-Preserving Compiler Infrastructure*, 2002

Gabriel Loh, Assistant Professor, Georgia Institute of Technology
Thesis: *Microarchitecture for Billion-Transistor VLSI Superscalar Processors*, 2002

Bratin Saha, Researcher, *Microprocessor Research Lab*, Intel
Thesis: *A Type System for Certified Runtime Type Analysis*, 2002

Zhanyong Wan, Software Design Engineer, Microsoft Corporation
Thesis: *Functional Reactive Programming for Real-Time Reactive Systems*, 2002

Karhan Eren Akcoglu, Quantitative Strategist, Goldman Sachs
Thesis: *Provably Good Computational Approximations to Some Difficult Problems in Finance and Economics*, 2002

Mark Anders Tullsen, Senior Research Associate, Oregon Graduate Institute
Thesis: *PATH, A Program Transformation System for Haskell*, 2001

Miklos Csuros, Assistant Professor, University of Montreal
Thesis: *Reconstructing Phylogenies in Markov Models of Sequence Evolution*, 2000

Christopher Eric Rasmussen, National Institute of Standards & Technology
Thesis: *Integrating Multiple Visual Cues for Robust Tracking*, 2000

Hong Xiao, Assistant Professor, UCLA at Davis
Thesis: *Prolate Spheroidal Wave Functions, Quadrature Interpolation, and Asympototic Formulae*, 2001

Zachary Bechman Dodds, Assistant Professor, Harvey Mudd College
Thesis: *Task Specification Languages for Uncalibrated Visual Servoing*, 2000

Chih-Ping Chen, Compaq Computer Systems
Thesis: *Mutable Abstract Datatypes--A Connection Between Linear Types and State Monads*, 2000

Martin Sulzmann, Lecturer, University of Melbourne
Thesis: *A General Framework for Hindley/Milner Type Systems with Constraints*, 1999

Sheng Liang, Staff Engineer, JavaSoft Sun Microsystems, Inc.
Thesis: *Modular Monadic Semantics and Compilation*, 1998

9. Graduate School Calendar

Schedule of Academic Dates and Deadlines 2006–2007
Fall Term, 2006

Friday, August 25 Welcome Center: New Student Orientation Check-in begins.

Monday, August 28 New student orientation week begins.

Wednesday, August 30 SPEAK Test for new international students in Ph.D. programs.

Thursday, August 31 Matriculation ceremony.

Friday, September 1 Fall-term On Line Course Selection (OCS) begins. Orientation in departments for all new students begins. Departmental New Student Orientations Begin

Monday, September 4 Labor Day. Administrative offices closed.

Tuesday, September 5 Orientation for all new Teaching Fellows; Registration for returning students begins.

Wednesday, September 6 Fall-term classes begin, 8.30 a.m.

Friday, September 8 Final day to pick up registration materials from academic departments.
SPEAK Test for international students in Ph.D. programs (alternative Date).

Friday, September 15 Final day to apply for a fall-term personal leave of absence. The entire fall-term tuition charge or continuous registration fee (CRF) will be canceled for students who withdraw from the Graduate School on or before this date or who are granted a leave of absence effective on or before this date.

Wednesday, September 20 Fall-term On Line Course Selection (OCS) ends. Final day for registration. A fee of $25 is assessed for course schedules accepted after this date.

Friday, September 29 One-half of the fall-term full-tuition charge will be canceled for students who withdraw from the Graduate School on or before this date or who are granted a medical leave of absence effective on or before this date (The CRF is not prorated.)

Monday, October 2 Final date for the faculty to submit grades to replace Temporary Incompletes (TI's) awarded during the 2005-2006 academic year.
Due date for dissertations to be considered by the Degree Committees for award of the Ph.D. in December.
Final day to file petitions for degrees to be awarded in December.

Friday, October 27 Midterm.
Final day to add a fall-term course.
One-quarter of the fall-term full-tuition charge will be canceled for students who withdraw from the Graduate School on or before this date or who are granted a medical leave of absence effective on or before this date. The CRF is not prorated.

Friday, November 3 Readers' Reports are due for dissertations to be considered by the Degree Committees for award of the Ph.D. in December.
Final day to change enrollment in a fall-term course from Credit to Audit or from Audit to Credit.

Final day to withdraw from a fall-term course.

Friday, November 10 Departmental recommendations are due for candidates for December degrees.
Final day to withdraw a degree petition for degrees to be awarded in December.

Thursday, November 16 SPEAK Test for international students in Ph.D. programs.

Friday, November 17 Fall recess begins, 5:20 p.m.

Monday, November 27 Classes resume, 8:30 a.m.

Friday, December 8 Classes end, 5:20 p.m.
YALE CORPORATION MEETING IN DECEMBER (December 8 & December 9).
Final grades for fall-term courses are due for candidates for terminal M.A. and M.S. degrees to be awarded in December.

Friday, December 22 Fall term ends; winter recess begins.

Spring Term, 2007
Wednesday, January 10 Final grades for fall-term courses due.

Thursday, January 11 SPEAK Alternative Test for new international students in Ph.D. programs.

Monday, January 15 Martin Luther King Jr. Day. Administrative offices closed. Classes do not meet.

Tuesday, January 16 Registration and spring ID validation begins.
Spring-term classes begin, 8:30 a.m.

Thursday, January 25 Final day to apply for a spring-term personal leave of absence. The entire spring-term tuition charge or CRF will be canceled for students who withdraw from the Graduate School on or before this date or who are granted a leave of absence effective on or before this date

Friday, January 26 Registration and spring ID validation end. Spring-term On Line Course Selection (OCS) ends. Final day for registration. A fee of $25 is assessed for forms accepted after this date.
.
Friday, February 9 One-half of the spring-term full-tuition charges will be canceled for students who withdraw from the Graduate School on or before this date or who are granted a medical leave of absence effective on or before this date. The CRF is not prorated.

Friday, March 9 Midterm.

Spring recess begins, 5:20 p.m.
Final day to add a spring-term course.
One-quarter of the spring-term full-tuition charge will be canceled for students who withdraw from the Graduate School on or before this date or who are granted a medical leave of absence effective on or before this date. The CRF is not prorated.

Friday, March 16 Due date for dissertations to be considered by the Degree Committees for award of the Ph.D. in May.
Final day to file petitions for degrees to be awarded in May.

Monday, March 26 Classes resume, 8:30 a.m.

Monday, April 2 Final day to change enrollment in a spring-term course from Credit to Audit or from Audit to Credit.
Final day to withdraw from a spring-term course.

Friday, April 6 Good Friday; classes meet. Administrative offices closed.

Monday, April 16 Readers' Reports are due for dissertations to be considered by the Degree Committees for award of the Ph.D. in May.

Friday, April 20 SPEAK Test for international students in Ph.D. programs.

Wednesday, April 25 Departmental recommendations are due for candidates for May degrees.

Friday, April 27 Final day to withdraw a degree petition for degrees to be awarded in May.

Monday, April 30 Classes end, 5:20 p.m.
Final day to submit Dissertation Progress Reports and petitions for extended registration.

Tuesday, May 15 Spring term ends.

Friday, May 18 Final grades for spring-term courses are due for candidates for terminal M.A. and M.S. degrees to be awarded at Commencement.

Sunday, May 27 Graduate School Convocation.

Monday, May 28 University Commencement.

Monday, June 4 Final grades for spring-term courses and full-year courses are due.

Friday, June 8 SPEAK Alternative Test for new international students in Ph.D. programs.

Inquiries concerning the contents of this handbook may be referred to:

Director of Graduate Studies
Department of Computer Science
Yale University
P.O. Box 208285
New Haven, CT 06520-8285

email: gradinfo@cs.yale.edu
phone: 203.432.1283

Last Revised: August 2006