GRADUATE HANDBOOK

2020–2021 Edition

Department of
Computer Science
Yale University

DEPARTMENT CHAIR

Zhong Shao

DIRECTOR OF GRADUATE STUDIES

Vladimir Rokhlin†

DIRECTOR OF UNDERGRADUATE STUDIES

James Aspnes

PROFESSORS

Dana Angluin, James Aspnes, Dirk Bergemann*, Ronald Coifman*, Julie Dorsey, Stanley Eisenstat, Joan Feigenbaum, Michael Fischer, David Gelernter, Mark Gerstein*, John Lafferty*, Rajit Manohar*, Drew McDermott (*Emeritus*), Dragomir Radev, Vladimir Rokhlin†, Holly Rushmeier, Brian Scassellati, Martin Schultz (*Emeritus*), Zhong Shao (*Chair)*, Avi Silberschatz, Daniel Spielman, Leandros Tassiulas*, Nisheeth Vishnoi, Y. Richard Yang, Lin Zhong, Steven Zucker†

ASSOCIATE PROFESSORS

Abhishek Bhattacharjee, Theodore Kim, Ruzica Piskac, Jakub Szefer*

ASSISTANT PROFESSORS

Yang Cai‡, Wenjun Hu*, Julian Jara-Ettinger*, Anurag Khandelwal [Spring], Smita Krishnaswamy*, Sahand Negahban*, Robert Soulé, Marynel Vázquez

SENIOR RESEARCH SCIENTISTS

Robert Bjornson, Andrew Sherman

SENIOR LECTURER

James Glenn, Stephen Slade

LECTURERS

Timothy Barron, Benedict Brown, Rob Brunstad, Kyle Jensen*, Cody Murphey, Scott Petersen, Brad Rosen

* A secondary appointment with primary affiliation in another department or school.
† A joint appointment with another department.
‡ A primary appointment with secondary affiliation in another department or school.

For a complete list of CS Department personnel, please visit our website at http://cpsc.yale.edu/.

CONTENTS

1. INTRODUCTION

The Yale Department of Computer Science was founded in 1969 as a small Ph.D. program. It now has over 70 graduate students, including both Ph.D. and master's degree students.

The Ph.D. program in Computer Science stresses unique and innovative research. Toward that end, the course requirements are modest, 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 Department of Computer Science 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 the Yale community is increasingly dependent on and interested in 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, and in recent years it has developed collaborations with other disciplines important to Yale, including art, architecture, biology, psychology, economics, business, statistics, linguistics, music, medicine, and physics.

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 leads to relatively small changes in the character and perimeter of the field. Computer science is different because it's younger. When a few researchers parachuted into this field in the 1940's, they had no territory to extend. They invented computer science from scratch. They faced basic questions in science, algorithms, design, and philosophy; by answering them, they established the intellectual basis of the field. What does "computable" mean? How do we classify computations, distinguishing practical from impractical from impossible? How programming languages should 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 Department of Computer Science was founded in 1969, the fledgling field was viewed as comprising 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 contains many sub-areas. The original four areas are still represented in the Department, and they have been joined by other areas, including graphics, robotics, programming languages, and the cross-cutting area of security and privacy.

Computer Science has also become both a theoretical and an experimental discipline. 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 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 computer science faculty have engaged in collaborations with researchers in 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 YINS (Yale Institute for Network Science, members of which come from CS, EE, and Sociology), the Center for Scientific Computing (members of which come from CS and Engineering), and the C2 Initiative (Creative Consilience of Computing and the Arts). C2 provides students with opportunities to apply mathematical, computational, and technological tools to problems inspired by the arts (Art, Art History, Music, Theater Studies, and Architecture).

Of course, with such diversity and depth, it has become increasingly difficult for any Department of Computer Science to cover all areas, and no department really tries to do so. The many changes that are taking place in the field have so far led Yale’s CS Department to pursue research in the following areas: algorithms and complexity theory, databases, distributed computing, machine learning, programming languages and compilers, scientific computing and applied math, graphics, vision and robotics, security and cryptography, and Computing and the Arts. In March 2015, the Department joined Yale’s School of Engineering and Applied Science, and it is currently recruiting new faculty and broadening and deepening its efforts in both research and teaching.

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 who are preparing for research careers, in either academia or industry. The terminal master’s degree program is intended for students who want to pursue advanced study in computer science but do not necessarily intend to have research careers.

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 if he or she intends to obtain a master’s degree along the way. A student who has completed the master’s program and wishes to go on for a Ph.D. is not guaranteed admission to the doctoral program but is welcome to apply to it 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 5 or 6 years. The M.S. and the M.Phil. Degrees are granted to qualified students in the doctoral program who wish to obtain intermediate degrees on the way to the Ph.D. If you are admitted to candidacy, the university registrar’s office will auto-petition an M.Phil. on your behalf. (See 3.1.6.)

3.1.1. An Overview of the Doctoral Program

A typical Ph.D. student progresses through the program as follows.

First Year

A student is expected to take courses and familiarize himself or herself with the research activities underway in the Department. By the end of spring semester of the first year, the student must have teamed up with a research *advisor,* i.e., a faculty member who takes primary responsibility for the student's progress. Hence every student should begin discussion with a professor (or professors) whom he or she might want to work with early in his or her first year (or even before he or she arrives) in order to identify an advisor.

After the first year, the student comes under the direction of a *supervisory committee,* consisting of the advisor and two or three other faculty members who monitor and mentor the student. Usually the committee forms “automatically” and consists of faculty working in the same area as the advisor, but there are many exceptions, especially when the student's work crosses disciplinary or departmental boundaries. In particular, if the day-to-day advisor's primary appointment is not in the Department of Computer Science, the committee must contain an “advisor of record” whose primary appointment is in CS. Once a student has selected a supervisory committee, changes to the committee require consultation among the director of graduate studies (DGS), the old committee, and the proposed new one. In rare cases, such changes may require the approval of the entire faculty.

Second Year

A student continues taking courses, completing a total of ten courses by the end of the year. Two of these must be the CPSC 690 and CPSC 691 sequence, in which the student carries out a research project and writes a report on the results; these are known as the “690 project” and the “690 report.” Each student must serve as a teaching fellow (TF) for two terms; this is a key part of academic training. First-year students are not normally eligible to serve as TFs; thus, this requirement typically becomes relevant in the second year. However, it is not necessary for the student to satisfy the teaching fellow requirement entirely in his or her second year. The end of the second year is the culmination of two years of study and research. The student must pass an area exam, administered by the supervisory committee, in which he or she demonstrates breadth of knowledge in the research area of the 690 project. He must also finish the project itself and submit the report. A copy of the approved report must be delivered to the registrar.

Third Year

A student is expected to begin their dissertation research, usually under the direction of the same supervisory committee but committee membership can be changed at this point, in view of the direction that the student’s research has taken during his or her second year. If the student or the original supervisory committee believes that the research is not going well, this is a good time to find a new advisor, a new supervisory committee, and/or a new research topic. Students should notify the DGS of any difficulties they meet during their work with an advisor. The departmental registrar and the DGS must be notified in writing via email of the pending changes and any circumstances that led to the difficulties.

Completion and passing of a qualifying exam along with submittal of a dissertation thesis are both due by the conclusion of the spring term in which you complete your third year in the doctoral program.

Post Third Year

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. It is turned into the Graduate School, and the readers file reports approving it. At this point, the DGS will approve the degree, so far as the department is concerned. In the unlikely event that the DGS fails to approve the degree, the subsequent course of action will be decided by the Department Chair, on a case-by-case basis. The requirements for the Ph.D. have been met, and the degree is granted.

Admission to Candidacy

The thesis committee will vote to admit the student to candidacy and admission will be approved by the DGS when all of the requirements described above have been satisfied: course requirements, 690 project and report, area exam, two terms of serving as a teaching fellow, thesis advisor, and thesis prospectus. In the highly unlikely event that the Dissertation Committee and the DGS are not in agreement, the subsequent course of action will be decided by the Department Chair, on a case-by-case basis.

It is *required* that the student be admitted to candidacy by the end of his or her third year. The prospectus is usually the last piece to fall into place, and it *must* be done before the thesis committee votes on admission to candidacy. If the student has not been admitted to candidacy by the end of the third year, the Graduate School will not allow the student to register for classes at the beginning of his or her fourth year or to be paid as a research assistant or TF. Correcting this registration problem requires the Computer Science Department to negotiate with the Graduate School; negotiation usually results in the student's being placed on probation for a term with the expressed intention of all concerned that he or she will be admitted to candidacy before the end of that term. The possibility of probation for one term should *not* be interpreted to mean that the "real" deadline for completion of requirements is August 31 of the year in question. Rather, as explained above, *the spring term ends in May. The exact date on which the term ends in any particular year can be found in the Yale College calendar on the Yale website; it is this end-of-term date this is the deadline for admission to candidacy. A certification must be signed by the DGS and sent to the university registrar's office*

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 research 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 research when his or her committee is satisfied that the work is finished, and a complete draft of the dissertation has been written. At least one week before the defense, the dissertation draft must be placed on the web, and the faculty must be notified that it is available. If the student or the advisor does not want the draft to be publicly accessible, it can be placed on the Department's website in the "Internal CS Only" section (on the "Faculty Use Only" page.)

The defense consists of a one-hour presentation of the results followed by a 15-minute question and discussion period, both of which are open to the entire department and its guests. The committee of readers and any other faculty members who wish to participate then conduct an oral examination in closed session.

At least three readers must be present at the defense, but "presence" may be achieved technologically, e.g., by teleconference or Zoom. It is not necessary for the external reader to attend.

The defense must be scheduled by the student and advisor in consultation with the DGS and the business office, announced one month in advance, and re-announced one week in advance. The announcement must be sent by email to the entire department and must include the date, time, and location of the defense, the student's name, the title of the dissertation, a one-paragraph abstract, and the names of the student's advisor and committee members.

Dissertation Submission

The dissertation must be submitted to the Graduate School after the thesis defense has been passed and final corrections to the dissertation draft have been made. The Graduate School requires that the dissertation be submitted by **October 1** for a December degree and by **March 15** for a May degree, *and these deadlines are strictly enforced.* Thus the dissertation must be submitted *after* the student completes the defense and final corrections and *on or before* the Graduate School's deadline for the degree; exactly when in this interval a particular student should submit his or her dissertation to the Graduate School depends on which term is the last in which he or she will register as a PhD student. The submission date should be determined by the student and the advisor in consultation with the DGS at the time that the defense is scheduled. After the dissertation is submitted, copies are sent to the readers, each of whom reads the dissertation and completes a "reader's report" form. When all readers' reports are in, the DGS will approve the degree, so far as the department is concerned. In the unlikely event that the DGS fails to approve the degree, the subsequent course of action will be decided by the Department Chair, on a case-by-case basis.

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 (Yale University), and one must be external (another university). 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 Computer Science Department. In addition, at least two internal readers must be regular ladder faculty in the Yale Computer Science Department. Before submitting the final dissertation to the Graduate School, be sure your readers have been submitted online here. Your readers will have thirty days to read and submit a report upon receipt of your dissertation.

Exceptions to these rules require approval of the DGS. For the purposes of these rules, "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 Yale CS-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 Department of Computer Science. 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. Please notify your registrar of the name, address and phone number of your external reader, along with other readers. Enter the name of you readers to the Dissertation Reader Report system.

3.1.2. Requirements

We now describe in detail the milestones along the way to the Ph.D. The course requirements, examinations, and 690 project and report must be completed by the end of the second year. All requirements for admission to candidacy must be completed by May of the third year. The Graduate School will not allow a student to register for a fourth year of study until this step has been completed. The dissertation and defense must be completed no later than the end of the sixth year.

Milestone Requirements

690 Project

The student must submit a written report on his or her 690 project to his or her supervisory committee; the committee members grade the report, taking into account the quality of the work, the quality of the technical writing, and English proficiency. The grade and a one-page abstract must be transmitted to the DGS.

Note that the course grades for CPSC 690 and CPSC 691 are not the same as the grade on the 690 *report*. The advisor files a grade of "SAT" for CPSC 690 and a grade of H or HP for CPSC 691 if the student is making satisfactory progress toward completing the research and the report. If the student is not making satisfactory progress, for a grade of "UNSAT" for CPSC 690 or below HP for CPSC 691, the advisor must notify the supervisory committee and the DGS. The course grade is submitted at the end of the term, with other course grades. The grade on the 690 report is submitted after the supervisory committee has read it and agreed on a grade, which may occur before or after the course grade is submitted.

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 scholarly proficiency in a subject area that includes the topic of the student's 690 project but is broader. The exam is formulated and administered by the student's supervisory committee. The committee must decide whether to give the same exam simultaneously to a cohort of students in the same area or to give a customized exam to each student in the area.

Thesis[1] Advisor

A regular faculty member must agree to direct the student's dissertation, thereby certifying that the student is capable of doing original research. This requirement is not automatically satisfied by the student's receiving a grade of H or HP on his or her 690 report. The advisor may be a ladder faculty member from another Yale department if the student's supervisory committee and the DGS approve; if the advisor's primary appointment is not in CS, then another member of the supervisory committee whose primary appointment is in CS must serve as the student's "advisor of record." It is generally unproductive for a student to attempt a dissertation in an area not covered by the advisor's interests. The thesis advisor must be chosen by the beginning of the third year.

Thesis Prospectus

A thesis prospectus must be filed with the DGS and the Graduate School by the end of the spring term of the student's third year in the program. The prospectus is a written summary (usually about three or four pages long) of the nature and scope of the research, along with a tentative title of the dissertation. The prospectus must also

[1] Throughout this Handbook, the words "thesis" and "dissertation" are synonyms.

include a proposed committee of readers (as defined below) and be signed by the advisor. *Note that the spring term ends in May. The exact date on which the term ends in any particular year can be found in the Yale College calendar on the Yale website.*

By the end of the third year, the student should have satisfied all requirements for admission to candidacy, including writing a *dissertation prospectus* that describes the general area and direction of the dissertation research and proposes a committee of readers. Once all requirements have been satisfied, and the thesis committee has voted to admit, the admission will be approved by the DGS. In the highly unlikely event that the Dissertation Committee and the DGS are not in agreement, the subsequent course of action will be decided by the Department Chair, on a case-by-case basis.

Course Requirements

Students are required to pass ten courses that, together, satisfy the following constraints:

Two of the courses must be CPSC 690 and CPSC 691, which were mentioned in Section 3.1.1. CPSC 690 must be passed with a grade of SAT and CPSC 691 must be passed with a grade of HIGH PASS (HP) or HONORS (H). Two of the remaining courses must be passed with a grade of H; the rest must be passed with a grade of at least HP. In order to satisfy the program's depth requirement, each student must pass three advanced courses in a particular field by the end of his or her second year.

CPSC 991, "Ethical Conduct of Research," must be taken in the fall semester of the first year in the PhD program and passed with a grade of SAT.

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

General constraints:

- 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 10-course requirement and the depth requirement. For example, if a theory student takes a 500-level theory course, it simultaneously counts toward both the 10-course requirement and the depth requirement.
- The CPSC 690 and CPSC 691 course sequence counts only toward the 10-course and research requirements.
- A 600-, 700-, or 800-level course may count toward the 10-course and depth requirements if it involves regular meetings with a faculty member, requires written work that can be evaluated, and results in a regular grade of H or HP. A student should check with the DGS at the time he or she registers for a 600-, 700-, or 800-level course in order to be sure that the course will count toward the 10-course and/or depth requirement. Typically, most 600-level courses do count, but 700- and 800-level courses count only with explicit DGS approval at the time of registration.
- *Non-CS Department courses:* A student can count a graduate-level course outside the Department towards the 10-course and depth requirements if it is relevant to the student's program of study. The decision about whether a non-CS course satisfies a requirement is made by the DGS, possibly in consultation with the student's advisor.

Distribution constraints:

Two of the ten courses must be in "theoretical computer science," currently including 557, 561, 562, 563, 565, 567, 568, 569, 662, and 667. Two must be in "programming languages and systems," currently including 521, 522, 523, 524, 526, 527, 528, 530, 533, 534, 535, 536, 537, 538, 539, 554, 625, 638, and 639. Two must be in “applications,” currently including 510, 511, 512, 531, 532, 540, 545, 551, 558, 570, 571, 572, 573, 575, 576, 578, 579, 679, 745, and 752.

At least one course taken at Yale must be a Computer Science course designated as "programming-intensive," with a workload whose goal is to produce several thousand lines of code at a high level of proficiency. The programming-intensive courses currently are 521, 522, 524, 526, 527, 539, 573, 578, 579, 625, and 639.

A student should check with the DGS if he or she has questions about whether a Computer Science course not mentioned in the previous two paragraphs satisfies a particular requirement.

Exceptions to these requirements and constraints may be made by the DGS.

Other constraints:

All the courses should contain substantial material beyond what the student has learned before coming to Yale. 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, usually 700-level, presented on an ad-hoc basis that may consist mainly of paper presentation and discussion. 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.

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, courses must be completed with a grade of HP or better. When a student receives a grade of P in a course, he or she is expected to take another course (in order to have passed at least 10 courses with grades of H or HP by the end of the second year).

After the first year, progress is monitored by the student’s supervisory committee. The committee looks at grade records, exam results, the 690 report, and research progress. Students beyond the first year receive written annual evaluations of their progress, drafted by the supervisory committee. Copies of evaluations are 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.

Students Not in Good Standing (failing to achieve required milestones or by recommendation of his or her supervisory committee)

1. 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 and a course of action will be determined for the student. Possibilities at this stage can include continuation in the program with revised expectations, academic probation, or dismissal from the graduate program.

2. The DGS will inform the student in writing of the faculty’s determination and, if continuation in the program is permitted, conditions that must be met in order to return to good standing will be outlined.

3. If the reason for the student’s trouble is inability to do research under the supervision of his or her current committee, a new committee can be formed and an appropriate period of time is given to the student

(e.g., a term or a summer) to demonstrate ability to conduct a research project successfully. The new committee will report to the faculty at the end of this period so that a decision can be made about whether the student has returned to good standing.

4. If a supervisory committee determines that a student has not met one of his or her requirements, it should report that determination to the faculty and make a recommendation about how to proceed. A recommendation may range from termination, probation, a specific project assigned for completion under the direction of a specific faculty member, or change of advisor.

Subsequently, the committee should continue to monitor the student’s progress and to make recommendations to the faculty about how to proceed. A committee may recommend that the student change area of research and proceed under the direction of a new supervisory committee. This recommendation is not routine and should not be considered the normal consequence of failing an area exam.

If the problem is corrected in a timely fashion, the committee will eventually recommend the student (having met all pre-dissertation requirements) for admittance to candidacy.

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, so that a faculty meeting can be held before the end of May to act on any recommendations.

3.1.4. Teaching Assistance and other Requirements

In order to gain teaching experience, all graduate students are required to serve as TFs 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 for a degree or a financial-aid package may receive additional compensation – see Section 6.2.

The Graduate School requires that a Ph.D. student spend a minimum of three years in residence and that full tuition be paid for five 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.

Each student can expect to have office space in the same building as his or her advisor, subject to availability, for his or her 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 CS 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 CPSC690 (i.e., found an advisor and begun research) and passed six courses with grades of HP or H. The DGS will examine the student's academic history to decide which courses already taken satisfy which distribution requirements, and which requirements remain to be satisfied.

Students who expect to qualify as "fast track" may, with permission of the DGS, begin the 690 project in the first or second term of study. A student may also be granted permission to begin the 690 project in the first or second term if his or her 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 will it affect the number of courses that are eventually waived; both of those decisions are made by the DGS as described above.

Here in detail is the procedure that is now required to certify that a student is taking the fast track: A table must be prepared showing which Yale courses are obviated by which *graduate-numbered* courses at the institution where the student earned his or her M.S. All of these courses must have been taken during the student's post-graduate education, after the award of the bachelor's degree or equivalent. A theory course here must be paired with a theory course there, and so forth. However, the titles of the two courses do not have to be identical, or even similar, provided that, in the judgment of the DGS, the two courses occupy the same "ecological niche." That is, if Yale offered a course on that topic, it would be classified as a course that satisfied the same clause of the distribution requirement as the Yale course it is paired with. (Example: Suppose Yale offers a course on distributed sensor networks, and the M.S. institution offers a course on mobile computing. These can be paired, because, if Yale had a course on mobile computing, it [like the distributed sensor-networks course] would satisfy the systems requirement.)

The DGS prepares a certificate incorporating this table and sends it to the Dean of the Graduate School, who must approve waiving the courses proposed by the Department of Computer Science. No more than three courses may be waived.

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 it may be spread over a period as long as four years. To qualify for the Master of Science (M.S.) degree, the student must pass eight courses at the 500-level or above. These must satisfy the same "general constraints" as for Ph.D. students (see section 3.1.2). An average grade of at least HP is required, with at least one grade of H.

CPSC 990, "Ethical Conduct of Research," must be taken in the fall semester of the first year and passed with a grade of SAT.

A one-term independent project course (CPSC692) may be applied towards the M.S. with prior permission of the DGS, provided that a faculty member is willing to supervise the project, applying the same standards as for a Ph.D. student. The faculty is under no obligation to supervise independent projects for M.S. students.

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 DGS, be counted towards the M.S. degree. Generally, at most two such courses may be used to satisfy the requirements of the M.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. At least 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. Recently, the department has created the Yale CS Cloud (http://cloud.cs.yale.edu/), which allows research groups to conduct experiments without having to fund the full equipment costs on their own. All of the computer systems are interconnected by a switched Ethernet local network, which is connected via fiber optic technology to the campus backbone and the Internet. Also installed is a wireless network permitting instant laptop access to the Internet anywhere in the building and many places on campus.

Each faculty member or Ph.D. student is equipped with a PC or laptop, running some variant of the Linux, Windows, or OSX operating systems. The computing needs of undergraduate and M.S. students are met through the "Zoo," an educational laboratory with approximately 36 dual-processor Linux workstations that 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. 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 follows the 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 (203-432-2583), mcdougal.center@yale.edu, http://gsas.yale.edu/life-yale/mcdougal-graduate-student-center

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, and alumni of the Graduate School, and members of the larger Yale community. Its website (http://gsas.yale.edu/life-yale/mcdougal-graduate-student-center) provides all kinds of information relating to graduate student life. The McDougal 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 Common Room is the McDougal Center's main space for hanging out. Whether you come for relaxation or work, the Common Room – with its own Blue Dog Café, comfortable couches, many tables and chairs, wireless internet access, and beautiful painted ceiling – make it the perfect place to connect with Yale's graduate community. A variety of magazines, a women's book collection, board games, a baby grand piano, a microwave, and two email kiosks are also available.

The McDougal Center offers lots of resources for busy grad students. Stop by and see the student workers in HGS 124 for more information about these resources:

- Family Resource Room and Lactation Room, B45 HGS
- Music Practice Room, with upright piano - for occasional practice use by Graduate students
- ITS BluePrint Station on lower level HGS
- Lockers for student use - see room 124 HGS staff to request
- Vending Machines for snacks and soft drinks (when the Blue Dog Cafe is closed)
- Safer sex supplies from Student Wellness
- Magazines and Yale Daily News
- Brochure Racks with Yale and New Haven info
- CT Transit bus schedules
- High chair for children in Common Room, baby changing tables in restrooms nearby.

5.1.3 Student Life Programs

The McDougal 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. Contact Lisa Brandes, the Assistant Dean for Student Affairs and Director, Graduate Student Life, GSAS, by email, in person at HGS 126 or by phone at 203-432-2583 for answers, advice, and referrals, with confidentiality if requested. For a full listing of services visit the website.

5.1.4 Office of Career Strategy (OCS)

Yale Office of Career Strategy (OCS) offers career advising, alumni connections, interview preparation, employment opportunities, and extensive career development resources. OCS works with Yale students and alumni of Yale College and Yale Graduate School of Arts and Sciences to clarify career aspirations, identify employment and educational opportunities, and offer advising and support at every stage of career development. http://ocs.yale.edu/

5.1.5 The Teaching Fellow Program

The Teaching Fellow Program (TFP) provides the principal framework at Yale in which graduate students learn, under faculty guidance, to become effective teachers and to evaluate student work. Such learning is integral to the preparation of graduate students for professional lives of teaching and scholarship. In order to obtain the greatest benefit from this training, teaching fellows are urged to participate in the programs offered by the Center for Teaching and Learning (http://ctl.yale.edu/teaching/graduate-student-professional-student-and-postdoctoral-teaching-development) designed to prepare them for the variety and complexity of classroom environments that they will encounter. Howard El-Yasin, Assistant Director, 203-432-2757. Warner House, 108.

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, and new theories of computation? How should computer science be taught? What should the legacy of our efforts be?

The Department runs a Colloquium Series in which distinguished researchers from other universities are invited throughout the year to speak to a general computer science audience. Individual research groups also host talks for more focused audiences.

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 124,000. 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 sandwiches 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, vegetarian, and vegan.

For outdoor and sports enthusiasts, New Haven boasts 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 from Union Station.

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.

Application for admission in the fall of 2020 should begin in the fall of 2019. Information may be obtained from:
http://www.yale.edu/graduateschool/admissions

Contact Information for the Graduate School is:

Yale University Graduate School of Arts and Sciences
Office of Graduate Admissions
302 Warner House
1 Hillhouse Avenue
New Haven, Connecticut 06511
Telephone: (203) 432-2771

Application 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**

The deadline for completed applications, including all letters of recommendation and test scores, is **January 2, 2020**.

Applicants will be notified of action concerning admission as soon as the decision has been made, generally between March 15 and April 15. 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 $105. 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 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
609-771-7650
www.ets.org | www.toefl.org

6.2 Financial Aid Policy

The department will provide financial support for at least the first five years of study.

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.

Advisors will make every effort to provide financial support after the fifth year. In those cases in which support is not available, PhD. students are encouraged to explore paid teaching opportunities. Note that a nominal "Continuous Registration Fee" (currently $650.00 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.)

7. GRADUATE COURSES

Prerequisites are not listed. If you are not sure that your background is adequate, please speak to the instructor.

CPSC 510a *The Law and Technology of Cyber Conflict*
Not offered 2019-2020 Academic Year
A cross-disciplinary seminar that addresses both technical and legal aspects of cyber conflict. Recent events, including the hacks of Sony and the U.S. Office of Personnel Management, illustrate the need for new thinking about the particular issues raised when cyber-attacks originate from state or quasi-state actors. Professors from Yale Law School and the Computer Science Department will lead in-depth explorations of cyber conflict from both legal and technical points of view. Enrollment is limited to ten Yale Law School students and ten students from Yale College or GSAS. This is the first half of a yearlong course; the second half is CPSC 511b. Students are required to make a yearlong, two-term commitment.

CPSC 511b *The Law and Technology of Cyber Conflict: Practicum*
Not offered 2019-2020 Academic Year
A cross-disciplinary practicum that addresses both technical and legal aspects of cyber conflict. Recent events, including the hacks of Sony and the U.S. Office of Personnel Management, illustrate the need for new thinking about the particular issues raised when cyber-attacks originate from state or quasi–state actors. Professors from Yale Law School and the Computer Science Department will oversee intensive student projects on both legal and technical aspects of cyber conflict. Enrollment is limited to ten Yale Law School students and ten students from Yale College or GSAS. This is the second half of a yearlong course; the first half is CPSC 510a. Students are required to make a yearlong, two-term commitment.

CPSC 512a *Designing the Digital Economy*
Not offered 2019-2020 Academic Year
Digitization is transforming a variety of markets from personal transportation services to advertising. This course explores the economic tools (market design, price theory, causal inference, etc.) and technical tools from computer science (machine learning, the analysis of algorithms, user interface design, etc.) students need to contribute meaningfully to this transformation. Prerequisites: elementary training in both economics and computer science and some intermediate/advanced training in at least one relevant field.

CPSC 521 *Compilers and Interpreters*

MW 1:00-2:15 Robert Soule

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 decisions on compiler construction.

CPSC 522a *Design and Implementation of Operating Systems*

TTH 2:30-3:45 Zhong Shao

The design and implementation of operating systems. Topics include synchronization, deadlock, process management, storage management, file systems, security, protection, and networking.

CPSC 523b *Principles of Operating Systems*

TTH 9:00-10:15 Avi Silberschatz

A survey of the underlying principles of modern operating systems. Topics include process management, memory management, storage management, protection and security, distributed systems, and virtual machines. Emphasis on fundamental concepts rather than implementation.

CPSC 524a *Parallel Programming Techniques*

MW 9:00-10:15 Andrew Sherman

Practical introduction to parallel programming, emphasizing techniques and algorithms suitable for scientific and engineering computations. Aspects of processor and machine architecture. Techniques such as multithreading, message passing, and data parallel computing using graphics processing units. Performance measurement, tuning, and debugging of parallel programs. Parallel file systems and I/O.

CPSC 525 *Cloud Networking and Computing*

Not offered 2019-2020 Academic Year

Study of critical technology trends and new challenges in cloud and data center designs for different trade-offs of performance, scalability, manageability, and cost in the networking layers and big data analytical frameworks. Consideration of cloud infrastructure, including network topology, network traffic management, network management, transport protocols, programmable switches, network functions, virtualization, network reliability, and security.

CPSC 526a *Building Decentralized Systems*

Not offered 2019-2020 Academic Year

An exploration of the challenges and techniques for building decentralized computing systems, in which many networked computers need to cooperate reliably despite failures and without assuming centralized management. Topics include: decentralized storage systems, mobile and remote execution, hosting untrusted code, [byzantine] fault tolerance, naming, capabilities, information flow control, distributed shared memory, distributed hash tables, content distribution, and practical uses of cryptography.

CPSC 527a *Object-oriented Programming*

MW 4:00-5:15 Glenn James

Object-oriented programming as a means to efficient, reliable, modular, reusable code. Use of classes, derivation, templates, name-hiding, exceptions, polymorphic functions, and other features of C++.

CPSC 528 *Language-Based Security*
MW 2:30-3:45 Zhong Shao

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 530 *Formal Semantics*
Not offered 2017-2018 Academic Year

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.

CPSC 531a *Computer Music: Algorithmic and Heuristic Composition*
MW 11:35-12:50 Scott Petersen

Study of the theoretical and practical fundamentals of computer-generated music, with a focus on high-level representations of music, algorithmic and heuristic composition, and programming languages for computer music generation. Theoretical concepts are supplemented with pragmatic issues expressed in a high-level programming language.

CPSC 532b *Computer Music: Sound Representation and Synthesis*
WF 11:35-12:50 Scott Petersen

Study of the theoretical and practical fundamentals of computer-generated music, with a focus on low-level sound representation, acoustics and sound synthesis, scales and tuning systems, and programming languages for computer music generation. Theoretical concepts are supplemented with pragmatic issues expressed in a high-level programming language.

CPSC 533a *Computer Networks*
Not offered 2019-2020 Academic Year

An introduction to the design, implementation, analysis, and evaluation of computer networks and their protocols. Topics include layered network architectures, applications, transport, congestion, routing, data link protocols, local area networks, performance analysis, multimedia networking, network security, and network management. Emphasis on protocols used in the Internet.

CPSC 534 *Topics in Networked Systems*
MW 4:00-5:15 Richard Yang

Networked systems such as the Internet and mobile networks provide the major infrastructure components of our information-based society. This course covers the design principles, implementation, and practical evaluation of such systems in new settings, including cloud computing, software-defined networking, 5G, Internet of things, and vehicular networking. The final project involves the design, implementation, and evaluation of a new networked system.

CPSC 535 *Internet-scale Applications*
Not offered 2019-2020 Academic Year

An introduction to the design and implementation of Internet-scale applications and services. Topics include: service-oriented software design; cloud computing paradigms; infrastructure scalability and reliability; adaptive, open clients; protocol specification; performance modeling; debugging and diagnosis; and deployment and licensing.

CPSC 536 *The Hardware/Software Interface*

TTH 1:00-2:15 Abhishek Bhattacharjee

Interactions of computer architecture and systems software for high-performance systems. In particular, the focus is on advanced aspects of compilation, particularly program analysis, code generation, optimization (i.e., the back end of modern compilers). No prior exposure to compilers is assumed, but the front end (i.e., lexing, parsing, etc.) is covered only briefly, in as much as is needed for studying compilation back end. Hands-on exposure to programming projects using LLVM.

CPSC 537a *Introduction to Databases*

TTH 9:00-10:15 Abraham Silbershchatz

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. Database data structures: files, B-trees, hash indexes.

CPSC 538 *Database System Implementation and Architectures*

Not offered 2019-2020 Academic Year

A study of systems programming techniques, with a focus on database systems. Half the course is spent studying the design of a traditional DBMS, supplemented by a hands-on exercise where students build various components (e.g., a catalog-manager, a buffer-manager, and a query execution engine) of a DBMS prototype. The other half is spent on non-traditional architectures (parallel databases, data warehouses, stream databases, Web databases).

CPSC 539b *Software Engineering*

Not offered 2019-2020 Academic Year

Introduction to building a large software system in a team. Learning how to collect requirements and write a specification. Project planning and system design. Increasing software reliability: debugging, automatic test generation. Introduction to type systems, static analysis and model checking.

CPSC 540b *Numerical Computation*

Not offered 2019-2020 Academic Year

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.

CPSC 545a *Introduction to Data Mining*

Not offered 2019-2020 Academic Year

A study of algorithms and systems that allow computers to find patterns and regularities in databases, to perform prediction and forecasting, and to improve their performance generally through interaction with data.

CPSC 546b *Data and Information Visualization*

MW 9:00-10:15 Holly Rushmeier

Visualization is a powerful tool for understanding data and concepts. This course provides an introduction to the concepts needed to build new visualization systems, rather than to use existing visualization software. Major topics are abstracting visualization tasks, using visual channels, spatial arrangements of data, navigation in visualization systems, using multiple views, and filtering and aggregating data. Case studies to be considered include a wide range of visualization types and applications in humanities, engineering, science, and social science.

CPSC 549 *Computer Architecture and Artificial Intelligence*

TH 1:30-3:20 Ricard Lethin

Introduction to the development of computer architectures specialized for cognitive processing, both offline "thinking machines" as well as embedded devices. History of machines starting with early conceptions in defense systems to contemporary initiatives. Instruction sets, memory systems, parallel processing, analog architectures, probabilistic architectures, graph computing architectures, machine-learning architectures. Application and algorithm characteristics.

CPSC 551b *The User Interface*

TTH 9:00-10:15 David Gelernter

The user interface (UI) in the context of modern design, where tech has been a strong and consistent influence from the Bauhaus and U.S. industrial design of the 1920s and 1930s through the IBM-Eames design project of the 1950s to 1970s. The UI in the context of the windows-menus-mouse desktop, as developed by Alan Kay and Xerox in the 1970s and refined by Apple in the early 1980s. Students develop a detailed design and simple implementation for a UI.

CPSC 553a *Machine Learning with Biological Applications*

TTH 11:35-12:50 Smita Krishnaswamy

This course introduces biology as a systems and data science through open computational problems in biology, the types of high-throughput data that are being produced by modern biological technologies, and computational approaches that may be used to tackle such problems. We cover applications of machine-learning methods in the analysis of high-throughput biological data, especially focusing on genomic and proteomic data, including denoising data; nonlinear dimensionality reduction for visualization and progression analysis; unsupervised clustering; and information theoretic analysis of gene regulatory and signaling networks. Students' grades are based on programming assignments, a midterm, a paper presentation, and a final project.

CPSC 554a *Software Analysis and Verification*

MW 11:35-12:50 Ruzica Piskac

Introduction to concepts, tools, and techniques used in the formal verification of software. State-of-the art tools used for program verification; detailed insights into algorithms and paradigms on which those tools are based, including model checking, abstract interpretation, decision procedures, and SMT solvers.

CPSC 555 *Economics and Computation*

TTH 2:30-3:45 Yang Cai

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 to the formulation and solution of mechanism-design problems that are relevant to data networking and Internet-based commerce

CPSC 556 *Wireless Technologies*

MW 2:30-3:45 Wenjun Hu

Fundamental theory of wireless communications and its application explored against the backdrop of everyday wireless technologies such as WiFi and cellular networks. Channel fading, MIMO communication, space-time coding, opportunistic communication, OFDM and CDMA, and the evolution and improvement of technologies over time. Emphasis on the interplay between concepts and their implementation in real systems. The labs and homework assignments require Linux and MATLAB skills and simple statistical and matrix analysis (using built-in

MATLAB functions).

CPSC 557b *Sensitive Information in a Connected World*

Not offered 2019-2020 Academic Year

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. Readings consist of research papers that explore both the power and the limitations of existing privacy-enhancing technologies such as encryption and "trusted platforms."

CPSC 558 *Automated Decision Systems*

Not offered 2019-2020 Academic Year

People make dozens of decisions every day in their personal and professional lives. What would it mean for you to trust a computer to make those decisions for you? It is likely that many of those decisions are already informed, mediated, or even made by computer systems. Explicit examples include dating sites like match.com or recommendation systems such as amazon or Netflix. Most Internet ads on sites like Google or Facebook are run by real-time bidding (RTB) systems that conduct split second auctions in the hopes of getting your attention. Driverless cars offer the promise of safer highways. Corporations and other enterprises invest in decision support systems to improve the quality of their products and services. This course considers the spectrum of automated decision models and tools, examining their costs and effectiveness. Examples will come from a variety of fields including finance, risk management, credit-card fraud, robotics, medicine, and politics.

CPSC 559 *Building Interactive Machines*

MW 1:00-2:15 Marynel Vazquez

This advanced course brings together methods from machine learning, computer vision, robotics, and human-computer interaction to enable interactive machines to perceive and act in a variety of environments. Part of the course examines approaches for perception with different sensing devices and algorithms; the other part focuses on methods for decision-making and applied machine learning for control. The course is a combination of lectures, state-of-the-art reading, presentations and discussions, programming assignments, and a final team project.

CPSC 560 *Automata Theory and Formal Languages*

TTH 1:00-2:15 Andrew Bridy

Introduction to the theory of automata and formal languages, one of the building blocks of theoretical computer science. Major topics covered are finite automata, pushdown automata, and Turing machines, and their associated languages.

CPSC 561 *Foundations of Cryptography*

Not offered 2019-2020 Academic Year

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.

CPSC 562 *Spectral Graph Theory*

MW 2:30-3:45 Daniel Spielman

An introduction to spectral graph theory motivated by computer science, covering advanced topics in linear algebra and exploring the combinatorial meaning of the eigenvalues and eigenvectors of matrices associated with graphs. Applications to optimization, numerical linear algebra, error-correcting codes, pseudorandomness, and the discovery of graph structure.

CPSC 563 *Algorithm via Continuous Optimization*

NW 11:35-12:50 Nisheeth Vishnoi

Continuous optimization has played a major role in the recent development of fast algorithms for problems arising in areas such as theoretical computer science, discrete optimization, and machine learning. The approach is to first formulate the problem as a continuous optimization problem, even if the problem may be over a discrete domain, adapt or develop deterministic or randomized continuous-time dynamical systems to solve it, and then design algorithms for the problem via appropriate discretization. The goal of this course is to design state-of-the-art algorithms for various classical discrete problems through the use of continuous optimization/sampling. The algorithmic applications include shortest paths, bipartite matching, flows, linear programming, sampling, and counting. We present approaches including gradient descent, mirror descent, multiplicative weights update method, accelerated gradient descent, Riemannian descent, Newton's method, cutting-plane methods, Langevin dynamics, and Hamiltonian dynamics.

CPSC 565 *Theory of Distributed Systems*

Not offered 2019-2020 Academic Year

Models of asynchronous distributed computing systems. Fundamental concepts of concurrency and synchronization, reliability, topological and geometric constraints, time and space complexity, and distributed algorithms.

CPSC 567a *Cryptography and Computer Security*

Not offered 2019-2020 Academic Year

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. Focus on technology, with consideration of 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 568a *Computational Complexity*

TTH 2:30-3:45 James Aspnes

Introduction to the theory of computational complexity. Basic complexity classes, including Polynomial Time, Nondeterministic Polynomial Time, Probabilistic Polynomial Time, Polynomial Space, Logarithmic Space, and Nondeterministic Logarithmic Space. The roles of reductions, completeness, randomness, and interaction in the formal study of computation.

CPSC 569 *Randomized Algorithms*

MW 1:00-2:15 James Aspnes

Beginning with an introduction to tools from probability theory including some inequalities like Chernoff bounds, the course will cover randomized algorithms from several areas: graph algorithms, algorithms in algebra, approximate counting, probabilistically checkable proofs, and matrix algorithms.

CPSC 570b *Artificial Intelligence*

MWF 10:30-11:20 Stephen Slade

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 571 *Topics in Artificial Intelligence*

Not offered 2019-2020 Academic Year

An in-depth study of one area of artificial intelligence. Possible topics include automated planning, scheduling with explicitly represented objective functions, AI and philosophy of the mind, computational approaches to stereo vision, multiagent systems, and automated diagnosis using functional models. After CPSC 570a or with permission of instructor.

CPSC 572b *Intelligent Robotics*

MWF 10:30-11:25 Brian Scassellati

Introduction to the construction of intelligent, autonomous systems. Sensory-motor coordination and task-based perception. Implementation techniques for behavior selection and arbitration, including behavior-based design, evolutionary design, dynamical systems, and hybrid deliberative-reactive systems. Situated learning and adaptive behavior.

CPSC 573 *Intelligent Robotics Laboratory*

MWF 10:30-11:20 Brian Scassellati

Students work in small teams to construct novel research projects using one of a variety of robot architectures. Project topics may include human-robot interaction, adaptive intelligent behavior, active perception, humanoid robotics, and socially assistive robotics.

CPSC 574a *Computational Intelligence for Games*

TTH 11:35-12:50 James Glenn

Introduction to techniques used for creating computer players for games, particularly board games. Topics include combinatorial and classical game theory, stochastic search methods, applications of neural networks, and procedural content generation.

CPSC 575a *Computational Vision and Biological Perception*

MW 2:30-3:45 Steven Zucker

An overview of computational vision with a biological emphasis. 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 *Advanced Computational Vision*

Not offered 2019-2020 Academic Year

Advanced view of vision from a mathematical, computational, and neurophysiological perspective. Emphasis on differential geometry, machine learning, visual psychophysics, and advanced neurophysiology. Topics include perceptual organization, shading, color and texture analysis, and shape description and representation.

CPSC 577b *Natural Language Processing*

TTH 1:00-2:15 Dragomir Radev

Linguistic, mathematical, and computational fundamentals of natural language processing (NLP). Topics include part of speech tagging, Hidden Markov models, syntax and parsing, lexical semantics, compositional semantics, machine translation, text classification, discourse, and dialogue processing. Additional topics such as sentiment analysis, text generation, and deep learning for NLP.

CPSC 578a *Computer Graphics*

MW 11:35-12:50 Theodore Kim

An introduction to the basic concepts of two- and three-dimensional computer graphics. Topics include affine and projective transformations, clipping 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*
M 9:25-11:25 Julie Dorsey
An in-depth study of advanced algorithms and systems for rendering, modeling, and animation in computer graphics. Topics vary and may include reflectance modeling, global illumination, subdivision surfaces, NURBS, physically based fluids systems, and character animation

CPSC 610b *Computer Science and Law*
W 9:25-11:15 Joan Feigenbaum
This course focuses on socio-technical problems in computing, i.e., problems that cannot be solved through technological progress alone but rather require legal, political, or cultural progress as well. Examples include but are not limited to computer security, intellectual property protection, cyber-crime, cyber war, surveillance, and online privacy. The course is addressed to graduate students in Computer Science who are interested in socio-technical issues but whose undergraduate work may not have addressed them; it is designed to bring these students rapidly to the point at which they can do research on socio-technical problems. Students do term projects (either papers or software artifacts) and present them at the end of the term.

CPSC 625 *Advanced Cloud Computing Systems*
Not offered 2019-2020 Academic Year
This course will focus on the fundamental systems research that enabled and continues to power modern cloud computing. We will cover the production systems that run within data centers operated by large cloud companies such as Google, Microsoft and Amazon, as well as the ground-breaking academic research that paved the way for these systems. Technically, we'll focus on the abstractions and mechanisms required to build online services that are scalable, highly available, durable, and consistent. We'll cover the entire stack, ranging from single-machine systems to protocols for distributing and replicating data within and across data centers. By the end of the course, students should be able to critically read a systems paper; distill, apply and improve upon key ideas from it; and write a short paper describing their own systems project.

CPSC 634 *Building an Internet Router*
MW 2:30-3:45 Robert Soulé
This course combines seminar-style readings and discussions with practical, hands-on development of a term-long project. Students read a selection of papers to get both a historical perspective and exposure to current research in networking. Students write reviews of the papers to make sure everyone keeps up with the readings and to develop their (technical) communication skills. Throughout the term, students work in teams to develop a fully functional IP router. Students design the control plane in Python on a Linux host and design the data plane in the new P4 language. Teams must demonstrate that their routers can interoperate with those of the other teams by building a small topology utilizing everyone's router. At the end of the course, teams participate in an open-ended design challenge.

CPSC 638 *Database Architectures*
Not offered 2019-2020 Academic Year
This course focuses on modern database architectures. Although the traditional database architecture (centralized, relational, disk-resident, ACID) has been very successful and continues to be in widespread use, new applications and technologies have led to a wide variety of nontraditional architectures in commercial and open-source projects as well as academia. This course examines why these new architectures have developed and explores aspects of their novel features.

CPSC 639 *Cloud-Scale Software Engineering*

Not offered 2019-2020 Academic Year

This course teaches students engineering methodology, design, and implementation skills that are needed for developing software systems that may span a range of scales, from desktop/mobile applications to Cloud-level distributed systems. The course begins by covering software engineering foundations including the software lifecycle, software engineering models such as extreme programming and agile development, design patterns, modularity/reusability, version control, multi-threaded design, sockets, and file 1/0. The latter portion of the course extends these foundations by focusing on software scalability and reliability. To this end, we examine distributed Cloud platforms and cover Cloud-specific concepts such as MapReduce, key-value stores, and log-based platforms.

CPSC 640b *Topics in Numerical Computation*

MW 2:30-3:45 Vladimir Rokhlin

This course discusses several areas of numerical computing that often cause difficulties to non-numericists, from the ever-present issue of condition numbers and ill-posedness to the algorithms of numerical linear algebra to the reliability of numerical software. The course also provides a brief introduction to "fast" algorithms and their interactions with modern hardware environments. The course is addressed to Computer Science graduate students who do not necessarily specialize in numerical computation; it assumes the understanding of calculus and linear algebra and familiarity with (or willingness to learn) either C or FORTRAN. Its purpose is to prepare students for using elementary numerical techniques when and if the need arises.

CPSC 659a *Building Interactive Machines*

Not offered 2019-2020 Academic Year

This course brings together methods from machine learning, computer vision, robotics, and human-computer interaction to enable interactive machines to perceive and act in dynamic environments. Part of the course examines approaches for perception with a variety of devices and algorithms; the other part focuses on methods for decision-making. The course is a combination of lectures, reviews of state-of-the-art papers, discussions, coding homework, and a final team project.

CPSC 662a *Spectral Graph Theory*

Not offered 2019-2020 Academic Year

An applied approach to spectral graph theory. The combinatorial meaning of the eigenvalues and eigenvectors of matrices associated with graphs. Applications to optimization, numerical linear algebra, error-correcting codes, computational biology, and the discovery of graph structure.

CPSC 663b *Deep Learning Theory and Applications*

TTH 4:00-5:15 Smita Krishnaswamy

Deep neural networks have gained immense popularity in the past decade due to their outstanding success in many important machine-learning tasks such as image recognition, speech recognition, and natural language processing. This course provides a principled and hands-on approach to deep learning with neural networks. Students master the principles and practices underlying neural networks, including modern methods of deep learning, and apply deep learning methods to real-world problems including image recognition, natural language processing, and biomedical applications. Course work includes homework and a final project—either group or individual, depending on the total number enrolled—with both a written and oral (i.e., presentation) component.

CPSC 667 *Advanced Cryptography and Security*

Not offered 2019-2020 Academic Year

Recent developments in cryptography. Topics include secure multiparty computation, verifiable computation, cryptographic obfuscation, functional encryption, and more. We study the motivation for, applications of, and security requirements for each of these primitives. We then focus on a few different constructions that instantiate each primitive and the formal proofs of security for them. Another point of consideration is the efficiency properties for the constructions, both asymptotically and in concrete practical terms when implementations are available.

CPSC 671 *Advanced Artificial Intelligence*

Not offered 2019-2020 Academic Year

This terms' topic is Artificial General Intelligence, or AGI. After 50 years, AI has scored some impressive successes, but has not yet produced a satisfying "artificial person," that is, an entity that possesses a person's ability to cope with many different situations, including linguistic discourse. Some people think this is because the field has made wrong turns toward overly specialized research. They have created a series of conferences on AGI to address the need for research that keeps its eyes on the long-term prize-the artificial person. To investigate all this, the course will be oriented around research papers in both the AGI subfield and its completion, "narrow AI" work on robotics, question answering, and such. Students will have opportunities to present and of course, discuss these papers.

CPSC 677a *Advanced Natural Language Processing*

Not offered 2019-2020 Academic Year

Advanced topics in natural language processing (NLP), including related topics such as deep learning and information retrieval. Included are: (1) fundamental material not covered in the introductory NLP class such as text summarization, question answering, document indexing and retrieval, query expansion, graph-based techniques for NLP and IR, as well as (2) state-of-the-art material published in the past few years such as transfer learning, generative adversarial networks, reinforcement learning for NLP, sentence representations, capsule networks, multitask learning, and zero-shot learning.

CPSC 679a *Movie & Game Physics Simulation*

TTH 2:30-3:45 Theodore Kim

This course covers computational methods that commonly arise when simulating physics in movies, games, and computational fabrication. In particular, we learn state-of-the-art methods for simulating fluids (fire and water) and solids (muscles, clothing, and skin). The algorithms discussed span offline techniques suitable for movies and fabrication, as well as real-time techniques for games. We cover finite difference and finite element representations as well as solver practicalities such as conjugate gradients, preconditioning, and Newton iteration.

CPSC 690a or b *Independent Project:* By arrangement with Faculty

Individual research for students in the PhD program. Requires a faculty supervisor and the permission of the Director of Graduate Studies.

CPSC 691a or b *Independent Projec*t: By arrangement with Faculty

Continuation of CPSC 690b or a.

CPSC 692a or b *Independent Project:* By arrangement with Faculty

Individual research for students in the MS program. Requires a faculty supervisor and the permission of the Director of Graduate Studies.

CPSC 745 *Advanced Topics in Machine Learning and Data Mining*
Not offered 2019-2020 Academic Year

An overview of advances in the past decade in machine learning and automatic data-mining approaches for dealing with the broad scope of modern data-analysis challenges, including deep learning, kernel methods, dictionary learning, and bag of words/features. This year, the focus is on a broad scope of biomedical data-analysis tasks, such as single-cell RNA sequencing, single-cell signaling and proteomic analysis, health care assessment, and medical diagnosis and treatment recommendations. The seminar is based on student presentations and discussions of recent prominent publications from leading journals and conferences in the field.

CPSC 752 *Bioinformatics: Practical Application of Simulation & Data Mining*
MW 1:00-2:15 Mark Gerstein

Bioinformatics encompasses the analysis of gene sequences, macromolecular structures, and functional genomics data on a large scale. It represents a major practical application for modern techniques in data mining and simulation. Specific topics to be covered include sequence alignment, large-scale processing, next-generation sequencing data, comparative genomics, phylogenetics, biological database design, geometric analysis of protein structure, molecular-dynamics simulation, biological networks, normalization of microarray data, mining of functional genomics data sets, and machine learning approaches for data integration.

CPSC 800a or b *Directed Readings:* By arrangement with Faculty

CPSC 990a and 991a *Ethical Conduct of Research.*
F 1:30-3:20 Holly Rushmeier

Related Courses in Other Departments

ENAS 503	*Probabilistic Networks, Algorithms, and Applications*
ENAS 525	*Optimization I.*
ENAS 530	*Optimization Techniques*
ENAS 563	*Fault-Tolerant Computer Systems*
ENAS 875	*Introduction to VLSI Systems Design*
ENAS 907	*Computer Architectures for Cognitive Processing and Machine Learning*
ENAS 912	*Biomedical Image Processing and Analysis*
ENAS 944	*Digital Communication Systems*
ENAS 962	*Theoretical Challenges in Network Science*
PHIL 567	*Mathematical Logic I*
PHIL 568	*Mathematical Logic II*
STAT 541	*Probability Theory*
STAT 542	*Theory of Statistics*
STAT 551	*Stochastic Processes*
STAT 664/ENAS 954	*Information Theory*
STAT 665	*Data Mining and Machine Learning*

8. GRADUATE SCHOOL CALENDAR

Academic dates and deadlines are subject to change. Please refer to the online calendar for up to date information. Go to http://gsas.yale.edu, click on “Calendar” in the upper right, and then click on “View Academic Calendar” in the upper left.

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: sabrina.whiteman@yale.edu

Phone: 203-432-1283

Last Revised: September 20, 2019