GRADUATE STUDENT HANDBOOK

Spring 2006 Edition

Contents

For further information about the graduate program in Computer Science visit our Web site at `http://www.cs.sunysb.edu/` or write to `graduate@cs.sunysb.edu`.
Our postal address is:

Graduate Program Director
Department of Computer Science
Stony Brook University
Stony Brook, New York 11794-4400
Tel: (631) 632-8462, 8470

Stony Brook University is an affirmative action/equal opportunity educator and employer.

1 Introduction

This handbook describes the requirements for admission to the graduate programs of the Department of Computer Science and the requirements to earn a graduate degree. The handbook also contains general information about the graduate programs. *Students in the programs are responsible for understanding the material in this handbook and the Academic Regulations, Procedures and Degree Requirements in the Graduate Bulletin.* Exceptions to the requirements to accommodate special circumstances must be approved by the student's advisor and the graduate program director. Exceptions must be documented and included in the student's academic file.

2 Goals of the Programs

The Department of Computer Science offers an M.S. and a Ph.D. in Computer Science. The M.S. program is designed primarily to train students with professional goals in business, industry, or government, requiring a detailed knowledge of computer science concepts and applications. The program concentrates primarily on applied computer science, emphasizing software development, programming, computer systems, and applications. Each student is given the experience of working on a large scale software or hardware development project involving analysis, design, evaluation, and implementation.

The Ph.D. program is for students interested in obtaining academic or research positions in colleges and universities or in government or commercial research laboratories. The program gives students a rigorous and thorough knowledge of a broad range of theoretical and practical research subject areas and develops the ability to recognize and pursue significant research in computer science. The first two years of graduate study are devoted to coursework. By the end of the second year the research phase of the student's graduate career should be underway, with participation in advanced study and preliminary research work. The final years of graduate study are devoted to dissertation research. Upon entrance to the program, each student is assigned an academic advisor. Each Ph.D. student should seek a faculty member to serve as a research advisor no later than the end of the first semester in the program. The choice may be changed. However, each change of advisor may delay a student's progress. A research advisor is invaluable when it comes to issues such as financial support and progress through various examinations. Many faculty members have research groups and meetings by which a new student can become acquainted with the research being conducted in the Department. Please refer to Section 4.3.1 for the specific rules on choosing or changing an advisor.

A student who is progressing satisfactorily toward the Ph.D. can earn an M.S. degree as well. A student enrolled in the M.S. program can apply for admission to the Ph.D. program as described in Section 4.2.7.

3 Requirements for Admission to Graduate Study

Admission to the M.S. and Ph.D. Programs are handled separately by the departmental Admissions Committee. The requirements for admission to graduate study in computer science include.

A - Bachelor Degree: A bachelor's degree, usually in a science or engineering discipline or in mathematics, is required. The transcript should show a grade average of at least B (i.e., 3.0/4.0) in
(i) all undergraduate course work, and
(ii) in the science, mathematics, and engineering courses.

B - Basic Mathematics: Two semesters of college level calculus, plus a course in linear algebra. Also desirable is a course in either probability theory or probability and statistics.

C - Minimal Background in Computer Science: As a measure of that background the student must satisfy 5 of the 7 proficiency requirements listed in Section 4.2.1.

D. Acceptance by the Computer Science Department and Graduate School.

E. All applicants to the MS or PhD program must submit Graduate Record Examination scores for the general aptitude tests. Applicants are encouraged to submit GRE test scores for the advanced examination in Computer Science as well.

Students of exceptional promise with non-standard background or or who lack certain requirements may be considered for admission to the program on a provisional basis. The student will be informed of the requirements that must be satisfied for the termination of the provisional status.

3.1 Transfers into the Graduate Program

We do not have a separate procedure for transferring into our graduate programs. An applicant must apply for admission into one of our programs as usual. However, if admitted, an applicant might be able to transfer graduate credits from another school subject to the following rules:

- Only credits for bona fide graduate courses can be transferred. Graduate courses co-scheduled with undergraduate courses are not accepted.

- *No more than 9 credits* of graduate courses can be transferred. Students who took approved courses at Stony Brook can transfer up to 12 credits.

- In order to be counted towards graduation, the credits must be evaluated by one of our faculty members. The evaluation must establish an equivalence between a course being transferred from another institution to a Computer Science course in Stony Brook, which is accepted as part of the graduation requirements. The faculty member must be one of those who is teaching the corresponding graduate course on a regular basis.

- Graduate courses that do not meet the previous requirement can be transferred without being counted towards graduation. This can sometimes be useful because students who have earned 24 graduate credits of any kind need to be registered for only 9 credits (instead of 12) in order to have full status. Therefore, gaining this status early might reduce tuition liability.

4 Graduation Requirements

4.1 General Requirements

The requirements for the completion of a graduate degree sometimes change. Students are bound by the requirements that were in effect at the time of their admission to the program. Alternatively, they can choose to satisfy any subsequent set of requirements approved by the Department. You should keep abreast of the latest version of the Graduate Student Handbook, which is published on the Departmental Web site.

Students are expected to complete their degree requirements as expeditiously as possible. In particular, they should only take courses relevant to their degree. Taking courses outside the major (except English proficiency courses) must be approved by the Graduate Director.

4.1.1 Registration and Status

The status of a graduate student is defined as G1, G2, G3, G4, or G5. The first two, G1 and G2, refer to M.S. students and G3 through G5 to Ph.D. students.

An M.S. student typically enters the graduate program with status G1 and a Ph.D. student enters with status G3. After completing 24 graduate credits the student receives the status G2 and G4, respectively. **Note**: credits for incomplete courses are not counted towards the 24 credits required for the G2 and G4 status. Ph.D. students who have been advanced to candidacy are designated as G5 (see Section 4.3.5).

Students who enter the graduate program after obtaining a graduate degree or having completed 24 graduate credits at Stony Brook or at another institution in any discipline (not necessarily Computer Science related) can request G2 or G4 designation (whichever applies) from the Graduate School.

Students in the G1 or G3 status must register for 12 credits in order to have full-time status. G2 and G4 students must register for 9 credits each semester. G5 students must register for 6 credits of which at least 3 must be for CSE 699, Dissertation Research. Foreign students must be enrolled full-time throughout their course of studies in order to maintain legal immigration status.[1] Domestic students are not required to maintain full-time status, but they must register for at least one credit each semester. However, only full-time students are eligible for any kind of financial assistance. Also, part time students cannot live or work on campus.

An M.S. student (but not a Ph.D. student) can be considered to have full-time status in the last semester of studies even if he or she is registered for less than 9 credits. To obtain full-time certification, the student must be registered for the amount of credits that is sufficient to satisfy the graduation requirements. The certification is not automatic — the student has to submit a petition for an "underload" to the Graduate Secretary. Note that such an underload has a pitfall, which can seriously affect foreign students: The student cannot stay in the university past the semester in which an underload is granted. Therefore, failing any of the courses taken in this situation will put the student out of legal immigration status.

In addition, the following rules are in effect regarding Summer registration:

New students: New students who were admitted for full-time studies must register full-time during their first semester on campus. This means that summer admits must register for at least 6 credits for the summer session to which they were admitted. They do not need to register for Summer Session II if they have registered for Summer Session I.

Continuing students: Continuing students who have a GA or RA during the Summer are strongly encouraged to register for the summer. If no appropriate courses are available, students may register for 0 credits of CSE 800. The Graduate School advises this for reasons related to tracking federal grants, tax issues, and Homeland Security.[2]

Graduating students: Students need to be registered for the semester they plan to graduate. The Graduate School permits Summer graduates to register for 0 credits. Graduates in other semesters must register for at least 1 credit. (International students must make sure that they are registered for enough credits to have full-time status.)

4.1.2 Taking Undergraduate Courses as CSE 587

Sometimes a student might need to take an undergraduate course to satisfy an M.S. proficiency requirement (see Section 4.2.1) or a prerequisite to a graduate course. However, a graduate student **cannot register for such a course using an undergraduate course number**. Registering for a course using an undergraduate number may put a student out of legal immigration status, which may have serious implications as far as immigration authorities are concerned and jeopardize financial aid. An undergraduate (300 level) course can, *with permission of the Graduate Academic Director and the instructor*, be taken using the course

[1] Consult International Programs at the Graduate School regarding immigration and related legal issues.
[2] CSE 800 does not count towards any degree.

designation CSE 587 (Proficiency Requirement in Computer Science). To do so, a student should register for the section of CSE 587 which corresponds to the professor who is teaching the undergraduate course in question.
Please note the following restrictions:

- CSE 587 can only be used to take an undergraduate course. The syllabus of the undergraduate course **must** specify *additional work* that graduate students must do in order to pass the course. Graduate students taking an undergraduate course under the CSE 587 number must be graded separately from the undergraduate students.

- A student may not use CSE 587 to take an undergraduate course when he or she has previously taken an equivalent undergraduate course (at Stony Brook or elsewhere).

- When CSE 587 is taken to satisfy a proficiency requirement (see Section 4.2.1), it can count towards the 31 credits needed for the M.S. degree. However, a *maximum of 4 credits* of CSE 587 can be used in this way and the student must get a grade of B or better in each course. (CSE 587 is not counted towards the Ph.D. course requirements.)

- If a student uses CSE 587 to take an undergraduate course to satisfy an M.S. proficiency requirement and receives a grade of C, C+, or B−, the course will still satisfy the proficiency requirement, but *will not count towards the required 31 credits*. The grade of C− is not sufficient for proficiency purposes.

- CSE 587 can be used to take an undergraduate course that is not a proficiency requirement. This is allowed to enable a student to expand his or her knowledge in a non-proficiency area. However, such a course *will not count* towards the credits required for graduation. (Credits for this course *are* counted, however, towards maintaining the full-time status.)

- A student should register for 2 credits of CSE 587 when taking a 3 or 4 credit undergraduate course. The 2 credits registered for CSE 587 are counted towards maintaining full-time status as normal graduate credits.

CSE 587 courses, which an M.S. student plans to take to satisfy proficiency requirements, must appear on the student's plan that was previously approved by the Graduate Academic Advisor – see Section 4.2.1.

When a CSE 587 is taken not as part of a proficiency requirement, the student must still seek permission of the Graduate Academic Advisor and of the instructor. To this end, an M.S. student must use the form `http://www.cs.sunysb.edu/files/graduate/permission-to-enroll587.pdf` to petition the Graduate Academic Advisor and explain the reasons for taking CSE 587. If approved by the advisor, the petition must then be approved by the by the instructor.

Some Ph.D. students might also be advised by the Graduate Program Director to take an undergraduate course under the CSE 587 designation as part of their preparation for qualifying examinations. In this case, the course must be part of an approved student's plan for taking these examinations (see Section 4.3.4). In case there is a need to take an undergraduate course for reasons other than the qualifying examinations, Ph.D. students should get an approval from the Graduate Program Director using the form `http://www.cs.sunysb.edu/files/graduate/permission-to-enroll587.pdf`.

4.1.3 Grade Requirement

To be certified for graduation, a cumulative grade point average of 3.0/4.0 or better over all graduate courses is required by the Graduate School. In addition, a cumulative GPA of 3.0/4.0 is required for the courses taken to satisfy the M.S. or Ph.D. degree in Computer Science. In case of a repeated course, the latest grade is counted.

4.1.4 Curricular Practical Training

Some of the course credits required for the M.S. and the Ph.D. degrees can be satisfied with industrial internship. Due to government regulations related to work permits, international students must do internships through Curricular Practical Training (CPT) or Optional Practical Training (Section 4.1.5).

CPT can be taken *only* in conjunction with a course, as specified below, and certain restrictions apply.

- *M.S. Program*: CPT can be taken in conjunction with CSE 596 (Internship in Research), CSE 523/524 (M.S. Project), or CSE 599 (M.S. Thesis Research).
  - CPT can be taken in conjunction with CSE 596 *at most once*. The student must provide a detailed description (up to 1/2 page) of the duties to be performed as part of the internship and emphasize the educational value of the employment.
  - If CPT is taken in conjunction with CSE 523/524 or CSE 599, the work to be performed as part of the training must be an integral part of the student's M.S. project or thesis, whichever applies. The student must submit a detailed description (1-2 pages) of the work to be performed during the training and explain how it is integral to the project or thesis.
- *Ph.D. program*: CPT can be taken in conjunction with CSE 696 (Ph.D. Internship in Research) or CSE 699 (Ph.D. Dissertation Research).
  - CPT can be taken in conjunction with CSE 696 *at most twice*. The student must provide a detailed description of the duties to be performed as part of the internship and emphasize the educational value of the employment.
  - If CPT is taken in conjunction with CSE 699, it must be an integral part of the student's Ph.D. thesis work. The student must submit a detailed description (1-2 pages) of the work to be performed as part of the training and explain how it is integral to the dissertation research.

Whether CPT is taken in conjunction with CSE 596/696 or CSE 523/524/599/699, the aforesaid description of the work must be filed with the International Students Office after being endorsed by the student's project or thesis advisor, the employer, and the Graduate Program Director.

On completion of CPT, the student should supply an evaluation letter from the supervisor at the place of internship, written on company stationery, which describes the work performed during the internship and evaluates the student's job performance. This letter will be kept in student's file.

Note: An M.S. student will be denied CPT if all the remaining courses that are necessary for completion of the degree have incomplete grades. One exception is when the CPT is directly part of the M.S. thesis or CSE 523/4 project.

4.1.5 Optional Practical Training

International students are typically granted certain period when they can work in the U.S. during and after completion of their degree. This opportunity is known as Optional Practical Training (OPT). OPT is *not* part of the Computer Science graduate program. However, an international student who needs to take CSE 596 (Internship) can do so in conjunction with OPT, if for some reason this internship cannot be done as part of CPT. Please consult the specialists of the International Programs Department at the Graduate School regarding the rules governing the OPT option.

4.1.6 Controlling Course Load

Graduate courses and projects tend to require a substantial amount of work, so students are advised to plan carefully. A graduate student with fewer than 24 graduate credits (a G1 student)[3] must register for 12 credits to maintain full time status. He or she can complete the required number of credits by taking CSE 523/524/599 (M.S. Project or Thesis – whichever applies), CSE 593 (Independent Study), or CSE 698 (Teaching Practicum). A program including more than four regular courses is not advised.

4.2 Requirements for the M.S. Degree

Students in the M.S. degree program choose between two options, the M.S. with a thesis and the M.S. with a project. The course requirements depend on the option chosen.

Students are required to complete 31 graduate credits in the Computer Science Department. A list of graduate courses is provided in the course compendium at the end of this document. The required 31 credits should be obtained by satisfying the following key requirements of the M.S. program:

1. *Proficiency requirements (up to 4 credits)*. Students can take up to 4 credits of CSE 587 (at most two courses) to fill in *missing* proficiency requirements. All seven proficiency requirements must be satisfied by the time of M.S. certification.

2. *M.S. thesis or project (9 or 6 credits, respectively)*. This requirement can be satisfied by taking 6 to 9 credits of CSE 599 in case of the thesis option, and 6 credits of CSE 523/524 using the project option.

3. *Research, teaching, or industrial experience requirement (1 or 2 credits)*. This requirement can be satisfied using CSE 593 (Independent Study), CSE 698 (Practicum in Teaching), CSE 596 (Internship in Research), Computer Science seminars (excluding CSE 600), Special Topics courses, or regular 3 credit technical Computer Science graduate courses.

4. *Technical graduate courses*. The *remaining* credits can be satisfied by taking any graduate Computer Science course, **except** Seminars, Special Topics, CSE 523/4, CSE 587, CSE 593, CSE 596, CSE 599, and CSE 698. Relevant graduate courses in other departments can be used towards the 31 credits, if approved by the Graduate Program Director. To this end, the student must argue the case for taking a particular course by submitting a petition, which must also be endorsed by the student's project or thesis advisor.

Some of the above requirements are further clarified below.

4.2.1 Proficiency Requirements

These represent fundamental knowledge expected of any CS graduate by the time of M.S. certification. Each requirement may be satisfied at Stony Brook by completing an appropriate undergraduate course, taken as CSE 587, or a graduate course. Each may also be met by showing evidence of a similar course taken elsewhere if approved by an instructor of the course here. The proficiency requirements, together with acceptable Stony Brook courses that would demonstrate that the requirement is met, are enumerated below:

1. Theory of Computation: CSE 303 or CSE 540.
2. Algorithms: CSE 373 or CSE 548

[3] Graduate courses (even in a different discipline) completed at other universities may count, if appropriate documentation is submitted to the Graduate School.

3. Language/Compilers: CSE 304, CSE 307, CSE 504, or CSE 526
4. Architecture: CSE 320 or CSE 502
5. Databases: CSE 305 or CSE 532
6. Operating Systems: CSE 306 or CSE 506
7. Networks or Graphics: CSE 310, CSE 346, CSE 533, CSE 534, CSE 328, or CSE 528

Paperwork and Deadlines for Proficiency Requirements

The "Welcome Package" that every M.S. student gets when entering the M.S. Computer Science program includes a booklet, titled "Proficiency Requirements," which contains a page for each proficiency requirement. This form is also available on the Web at

`http://www.cs.sunysb.edu/files/graduate/proficiency-requirements.pdf`

Students who have taken one of the above proficiency courses (graduate or undergraduate) at Stony Brook should just mark the top portion of the corresponding proficiency page, since the requisite information will be available with the student's Stony Brook transcript.

Students who took an equivalent proficiency course at another institution must arrange an interview with a professor who teaches the same course in the Computer Science Department. The student **must** bring the transcript (from that other institution) together with the course syllabus and any other available supporting material. The professor would typically ask questions about the course material, and may not certify proficiency if he or she is not satisfied with student's current level of mastery of the subject.

If a student took one of the above undergraduate proficiency courses under the CSE 587 course designation, then he or she has to fill out the middle portion of the proficiency form and have it signed by the professor who taught the aforesaid course right after completion of the course. Note that CSE 587 can be taken to satisfy a proficiency requirement *only if* a similar undergraduate course was not taken by the student at another institution. Exceptions to this rule must be approved by the Graduate Academic Advisor. Please check Section 4.1.2 for the restrictions on taking CSE 587.

Two semesters rule: All proficiency requirements must be completed by the end of the *second semester* (excluding summers) in the M.S. program. Students who do not complete the proficiency requirements within this time frame will be blocked from registering for courses. Extensions can be granted only by the Graduate Academic Advisor.

First semester rule: *At the beginning of the first semester in the M.S. program*, before the add/drop deadline for courses, students must submit for approval a plan for completion of the proficiency requirements to the Graduate Academic Adviser. This plan will be kept in student's file and will be required for graduation clearance towards the end of the program. The forms for the plan are part of the "Proficiency Requirements" booklet.

One should keep in mind that undergraduate courses taken at Stony Brook or another institution *cannot* be taken again at Stony Brook as part of the proficiency requirements. (One can still take them *without* getting a graduation credit as CSE 587 with permission of the course instructor.) To help the Graduate Academic Adviser evaluate student's plan for completing the proficiency requirements, the student must bring a copy of the undergraduate transcript.

On completion of the last proficiency requirement (but no later than the end of the second semester in the M.S. program) the student must submit the remaining filled out pages from the "Proficiency Requirements" booklet to the Graduate Secretary who will add them to the student's file. The complete booklet has to be in the student's file before the student can be cleared for graduation.

4.2.2 No-Thesis Option

Students choosing the *no thesis* option are required to take the courses Laboratory in Computer Science CSE 523/524. These courses provide students with the experience of dealing with large-scale computer-oriented problems such as those encountered in commercial, industrial, or research environments. If a student has had such experience within the four years preceding entry into the program and is able to submit material (e.g., technical reports, publications, patents, etc.) describing the work, the CSE 523/524 requirement may be waived after review of the submitted material by the faculty member who is responsible for coordinating the Laboratory in Computer Science. Note, however, that no course credit will be given for this previous experience; the waived laboratory credits must be replaced by approved graduate electives in Computer Science.

Special restrictions on CSE 523/524:

1. Students taking CSE 523/524 may not use any CSE 599 (Thesis Research) credits toward their M.S. degree, except in case of switching between the options as described below.
2. CSE 523/524 may not be taken in the same semester.
3. CSE 523 and CSE 524 must involve a single substantial two-semester project, not two smaller projects with different advisors. Thus, switching project advisors implies that CSE 523 must be started anew.

4.2.3 Thesis Option

The thesis must be approved by a departmental faculty committee of no less than three members appointed by the Graduate Program Director. At the discretion of the committee, the student may be required to present a seminar on the topic of his or her thesis. A student registers for CSE 599 when writing a thesis. *At least 6, but no more than 9 credits* of this course can be applied towards the 31 credits required for the M.S. degree.

4.2.4 Project or Thesis Advisor

A student in the M.S. program must select a project (or thesis) advisor by the end of the *second* semester in the program. The role of the advisor is to guide the student through the M.S. studies, formulate a project or a thesis topic, and supervise the student towards the completion of the assigned task.

On selection of an advisor, a form must be filled out by both the advisor and the student, and submitted to the Graduate Director. Changing an advisor requires a new form. At the end of each semester (including summers, if the student is expected to work during summer semesters) the student is evaluated by the advisor. Two unsatisfactory evaluations in a row or three unsatisfactory evaluations in total will result in dismissal from the program.

4.2.5 Electing an Appropriate M.S. Option

A student is considered to have elected the Thesis Option by registering for CSE 599 and the No-thesis Option by registering for CSE 523. Switching between the options is permitted, but in general the credits accumulated under CSE 523/524 cannot be applied towards the Thesis option and the credits accumulated under CSE 599 cannot be applied towards the No-thesis option.

However, when all of the following conditions are met, 3 credits accumulated under CSE 599 can be used in lieu of CSE 523 and vice versa, provided that:

- The change of the option occurs without changing an advisor or topic.
- The change is approved by the thesis or project advisor.
- The advisor provides a statement, to be included in the student's file, certifying that the work performed so far under the old option is acceptable under the new option.

In exceptional circumstances (such as an unforeseen change of scope of the project), other swaps of credits between CSE 599 and CSE 523/524 might be approved by the Graduate Program Director on a recommendation of the student's faculty advisor.

4.2.6 Optional Research or Project Work in Computer Science

Students who wish to conduct research or participate in a project in addition to and separately from the options described in Sections 4.2.2 and 4.2.3 should register for CSE 593: Independent Study. This work must be done under the supervision of a faculty member in the Computer Science Department, and the student must register for the section of CSE 593 that corresponds to that faculty.[4] The student must prepare a description of the project or the course to be taken and submit it before the add/drop deadline to the project sponsor. The description will reside in the student's file. Note that although at most 2 credits of CSE 593 can be used towards the 31 credits required for the M.S. degree, students may register for additional credits of CSE 593 as appropriate.

M.S. students can register for up to 3 credits of CSE 593 in any semester.

4.2.7 Switching from the M.S. to the Ph.D. Program

An M.S. student who wishes to advance to the Ph.D. program must take the Qualifying Examination (see Section 4.3.4). *Regular applications to the Ph.D. program will not be considered from current M.S. students.* A student who passes the exam can *apply* for entry to the program with a letter or an email to the Graduate Program Director. Admission will generally be granted to those who pass the examination.

Since most M.S. students will have spent at least a year in the graduate program at Stony Brook before they apply for the Ph.D. program, it is expected that they already have identified an advisor for their Ph.D. work, and that they can secure RA support from this advisor. For this reason, departmental support in the form of TA or GA is not normally offered to M.S. students admitted to the Ph.D. program.

An M.S. student can attempt any part of the qualifying examination at any time during the M.S. program but must pass all three parts by the end of the second year (see Section 4.3 for a description of the qualifying examination). As in the case of Ph.D. students, an M.S. student can sit for at most 5 examinations and must pass all three parts.

An M.S. student who is admitted into the Ph.D. program must take the Research Proficiency Examination (RPE) within two semesters after entering the Ph.D. program.

4.3 Requirements for the Ph.D. Degree

4.3.1 Dissertation Advisor

A student in the Ph.D. program must select a dissertation advisor by the end of the *first* semester in the program. The role of the *dissertation advisor* is to guide the student through the Ph.D. studies, help with selection of a research topic, and teach the art of doing independent and significant research. Students are encouraged to contact individual faculty members to discuss their research interests.

[4] If the primary sponsor of the work is from a different department, the student must select a co-advisor from the Computer Science Department.

On selection of an advisor, a form must be filled out by both the advisor and the student, and submitted to the Graduate Director. Changing an advisor requires a new form. The student is expected to participate in research activities of the advisor's group and at the end of each semester (including summers, if the student is expected to work during summer semesters) the student is evaluated by the advisor. Two unsatisfactory evaluations in a row or three unsatisfactory evaluations in total will result in the dismissal from the program.

4.3.2 Research Proficiency, Thesis, and Dissertation Examination Committees

The purpose of the *Research Proficiency Examination (RPE) Committee* is to ascertain the student's preparation to undertake significant and original research investigation through the mechanism of the Research Proficiency Examination. The purpose of the *Thesis Committee* is to evaluate the student's Thesis Proposal and ascertain the progress towards the research objectives (see Thesis Proposal Requirement). The purpose of the *Dissertation Examination Committee* is to evaluate whether the student's dissertation meets the standards of the Ph.D. degree (see Section 4.3.8).

RPE Committee. The RPE committee must be formed by the end of the third semester in the Ph.D. program. It should include the research advisor(s) and at least two other faculty members from the Department. The research advisor(s) cannot chair the committee. The proposed RPE committee should be submitted for approval by the Graduate Program Director.

Thesis Committee. The Thesis Committee should include a least three members from the Computer Science Department: The thesis advisor(s), a committee chairperson (who cannot be an advisor), and another member (who is not an advisor). It may optionally include one or more members from outside of the Department or University. Typically, members of the RPE committee proceed to serve on the Thesis Committee.

The committee is appointed by the Dean of the Graduate School on the recommendation of the Graduate Program Director. The committee appointment form is submitted by the Graduate Program Director using information supplied by the student and advisor. It must be filed at least two weeks prior to the date of the examination.

Dissertation Examination Committee. The composition of this committee is the same as that of the Thesis Committee, except that the participation of an outside member is mandatory. The procedure for appointing the Dissertation Committee is the same as in the case of the Thesis Committee. Typically members of the Thesis Committee proceed to serve on the Dissertation Examination Committee.

4.3.3 Course Requirements

In the first year, a student seeking the Ph.D. degree will normally register for a full time load of courses selected in conjunction with an advisor in order to prepare for the Qualifying Examination. By the time of graduation, each student is required to accumulate at least 20 credits of full (regular lecture) courses, internship, special topics courses or seminars. At most 5 credits of seminars and internship can be included in the 20 credits required for graduation; generic courses, such as CSE 593, CSE 587, CSE 600, CSE 698, and CSE 699, **cannot** be included. In addition, the following requirements should be noted:

- *M.S.-specific courses*. Students in the Ph.D. program may *not* enroll in CSE 523/524 or CSE 599. These courses are specific to the M.S. program.

- *On-going research seminar*. The student must register and complete two semesters of CSE 600. Credits earned in this course cannot be used towards the 20 credits required for the Ph.D. program.
- *Internship, CSE 696*. At most two credits of Internship in Research can be counted towards the 20 credits required for the Ph.D. program.
- *Dissertation Research, CSE 699*. The Dissertation Research course can be taken *only* by Ph.D. students who have been advanced to candidacy (have G5 status). Prior to the advancement, students conduct research and participate in projects by taking CSE 593: Independent study. G4 students can register for up to 9 credits of CSE 593 in any semester. G3 students can register for only up to 3 credits of CSE 593.
- *Teaching requirement*. University policy requires that all doctoral students participate in an appropriately structured teaching practicum. This can be CSE 698 in conjunction with a TA in the first year.

4.3.4 Ph.D. Qualifying Examination

Students must pass the written qualifying examination to demonstrate their ability to undertake the course of study leading to the Ph.D. degree. Qualifying examinations are given twice a year: in May (usually the week after the finals period) and in early January. The student who plans to take any part of the Qualifying Examination must register with the Computer Science Graduate Secretary at least one week in advance and indicate which parts of the examination the student will attempt.

The examination consists of three parts, 4 hours each, based on undergraduate material as described below. Undergraduate Stony Brook courses covering that material are listed in parentheses. The Qualifying Examination Committee publishes more detailed information about the core undergraduate areas from which the Qualifying Examination is constructed (outline, chapters from textbooks). An appropriate way to prepare for the examination for students who have already taken an undergraduate course in a particular area is to take a graduate course in that area. Questions test not just routine knowledge but also the student's ability to use that material in a creative way.

Theory and Mathematics:
: Theory of Computation: Languages and Automata, Analysis of Algorithms, Discrete Mathematics, and Logic. The examination is based on the following courses: CSE 303, CSE 373, CSE 213, and CSE 371.

Software:
: Programming Languages, Compilers, Databases, and Graphics. The examination is based on CSE 307, CSE 304, CSE 305, and CSE 328.

Systems:
: Networks and Communication, Operating Systems, Computer Architecture, and Computer Organization. The examination is based on CSE 310, CSE 306, CSE 320, and CSE 220.

Parts can be taken individually at different times. Students will be graded on each part separately and the faculty as a whole will partition the grades into three ranges: *pass*, *retake*, or *fail*. Students who receive a grade of *fail* in any part are dismissed from the Ph.D. program. Students who receive a grade of *retake* for a part can retake that part. Students who receive a passing grade in all three parts are considered to have passed the entire examination. The following schedule must be satisfied in order to stay in the Ph.D. program:[5]

[5] The schedule does not count summer semesters

1. One part must be *attempted* at the end of the first semester
2. All three parts must be *attempted* by the end of the second semester
3. Two parts must be *passed* by the end of the second semester
4. All three parts must be *passed* by the end of the third semester

A student will be allowed *at most* two retakes. Hence, a student can sit for at most five examinations. Examinations are checked without revealing the identity of the students to the graders. In order to get credit for any particular problem in the examination, a student must not only provide a correct answer, but also explain the answer clearly. Hence the student is responsible if a correct answer is misunderstood by a grader.

The results of the written examination will be communicated to each student individually following a meeting of the faculty which evaluates the results of the examination along with the student's ability to do research and likelihood of completing the program. Students are not entitled to see their graded examination papers.

Planning for Qualifying Examinations

Prior to the add/drop deadline of the first semester in the program, each new Ph.D. student must prepare a *plan for taking and passing the qualifying examinations* using a form, which can be downloaded from `http://www.cs.sunysb.edu/files/graduate/plan-for-quals.pdf`. This plan must be submitted to and be approved by the Graduate Program Director. The plan must indicate the dates (month/year) in which each qualifying will be taken and the courses that the student is planning to take to help prepare for the exams. Usually students *audit* undergraduate courses to help with such preparation. Other times, however, Ph.D. students may feel that they must *take* undergraduate courses for credit. In this case, this must be done using the CSE 587 designation (see Section 4.1.2) and the course must appear as part of an <u>approved</u> plan for passing the qualifying examinations.

Ph.D. Qualifiers Handbook

More details on the qualifying examination can be found in the *Ph.D. Qualifiers Handbook for Students*, which can be downloaded from the Departmental Web site. Incoming Ph.D. students are bound by the rules of the Ph.D. Qualifiers handbook that were in effect at the time of admission. If subsequent changes to the examination occur, every effort will be made to provide smooth transition. M.S. students who aspire to enter the Ph.D. program by taking the qualifier examination are bound by the rules that are in effect when they take the examination.

4.3.5 Research Proficiency Examination (RPE)

The purpose of the Research Proficiency Examination is to ascertain the breadth and depth of the student's preparation to undertake a significant original research investigation.

By the end of the third semester since admission into the Ph.D. program[6] an RPE Committee will have been formed for each student and an agreement reached on a research project. (M.S. students who were admitted to the Ph.D. program after passing the qualifying examination must form the RPE committee by the end of their first semester in the Ph.D. program.) The project should be described by a one-page abstract which is signed by the student and the Committee's members and submitted to the Graduate Program Director. The abstract should describe a research area and, as narrowly as possible, a problem

[6] Excluding summers.

in that area. A list of relevant publications should be attached to the abstract. With the approval of the Committee a student may change the project description, but a change does not imply any deadline extension for taking the RPE.

By the end of the fourth semester (at the latest) the student will take the RPE.[7] (M.S. students who switched to Ph.D. must take the RPE by the end of their second semester in the Ph.D. program.) The student must submit a report, written in the form of a conference paper, which critically evaluates and integrates the current state of research relevant to the problem described in the abstract and presents the student's progress in solving the problem. Reports based on previously published or submitted papers, or on papers in progress, are acceptable provided that they satisfy the aforesaid requirements.

The student will give an oral presentation to the Committee, describing the work, which will be followed by a session where the committee will ask questions. The oral presentation should be about 1 hour long. The report should be made available to the committee at least one week before the presentation is given.

Each aspect of the RPE (written report, oral presentation, responses to questions) will be separately graded by each member of the Committee using special forms provided for this purpose (see `http://www.cs.sunysb.e`
. The Committee as a whole can decide three outcomes: *pass, retake, fail*. A student who receives a grade of *fail* is dismissed from the Ph.D. program. A student who receives a grade of *retake* must retake the examination within 30 days. If, on retaking the examination, the student does not pass, the student is dismissed from the Ph.D. program. A student who receives a grade of *pass* has fulfilled the research proficiency requirement.

Having passed both the qualifying examination and the RPE the student is advanced to candidacy. This status, called *G5*, is conferred by the Dean of the Graduate School upon recommendation of the Department. Note that unlike the change from G3 to G4, the change from G4 to G5 is not automatic — the student must request to be advanced to candidacy by notifying the Computer Science Graduate Secretary.

Students must advance to candidacy at least one year before defending their dissertations. In exceptional circumstances the Graduate Program Director may submit a written petition for a waiver of this requirement to the Dean of the Graduate School. The graduate school requires G5 students to register for 6 credits of which at least 3 must be for CSE 699, Dissertation Research. The remaining credits can come from either CSE 699 or other graduate Computer Science courses and seminars. Courses outside of the major require the approval of the dissertation advisor and Graduate Director.

Failure to complete the research proficiency examination within the specified time frame and obtain the G5 status is considered evidence of unsatisfactory progress. In particular, *students whose status remains at the G4 level beyond 4 semesters since being admitted into the Ph.D. program will loose the tuition waiver and may be dismissed from the Ph.D. program*.

4.3.6 Thesis Proposal Requirement

After the student has completed the requirements in Sections 4.3.3 and 4.3.5, and with the approval of the student's research advisor, the student will present a thesis proposal. The purpose of the thesis proposal is to assess student's progress towards the Ph.D. thesis. The proposal must be submitted to the student's Thesis Committee within 18 months of the time that the student had passed the research proficiency examination. Failure to fulfill this requirement by that time without a formal extension may be considered evidence of unsatisfactory progress towards the Ph.D. degree.

The major requirements of the thesis proposal are as follows:

1. The student must be thoroughly familiar with the background and current status of the intended research area.

[7] It is understood that the actual date of the examination might be scheduled 2-3 weeks past the end of the semester in order to accommodate individual schedules.

2. The student must have clear and well-defined plans for pursuing the research objectives.

3. The student must offer evidence of progress in achieving these objectives.

The student must be prepared to justify the effort to be expended in the research in terms of the value of the results expected, and to justify the extent and challenge of that research as evidence of research competence at the Ph.D. level.

The student will present the thesis proposal to the Thesis Committee in a seminar presentation. The presentation is not open to the general university community. It is limited to members of the committee, invited computer science faculty, and invited graduate students. Faculty members are free to question the student on any topics that they feel are in any way relevant to the student's objectives and career preparation.

Most questions, however will be directed towards verifying the student's grasp of the intended specialty in depth. The student will be expected to show complete familiarity with the current and past literature of this area.

The findings of the committee will be communicated to the student as soon as possible, and to the Graduate School within one week of the presentation of the proposal. If the committee finds the thesis proposal unsatisfactory, the student will submit an improved proposal, if such re-submission is approved by the Dean of the Graduate School.

4.3.7 Dissertation

An important requirement of the Ph.D. program is the completion of a dissertation which must be an original scholarly investigation. The dissertation shall represent a significant contribution to the scientific literature, and its quality shall be compatible with the publication standards of appropriate reputable scholarly journals.

4.3.8 Approval and Defense of Dissertation

The dissertation must be orally defended before a Dissertation Examination Committee, and the candidate must obtain approval of the dissertation from this committee. The oral defense of the dissertation is open to all interested faculty members and graduate students. The final draft of the dissertation must be submitted to the committee no later than three weeks prior to the date of the defense.

4.3.9 Satisfactory Progress and Time Limit

A student who does not meet the target dates for the Qualifying Examination, the Research Proficiency Examination, and the Thesis Proposal, or who does not make satisfactory progress towards completing thesis research may lose financial support. The candidate must satisfy all requirements for the Ph.D. degree within seven years after completing 24 credit hours of graduate courses in the Department of Computer Science at Stony Brook. In rare instances, the Dean of the Graduate School will entertain a petition to extend this time limit, provided it bears the endorsement of the Department's Graduate Program Director. A petition for extension must be submitted *before* the time limit has been exceeded. The Dean or the Department may require evidence that the student is still properly prepared for the completion of work.

4.3.10 Part Time Students

Students admitted into the Ph.D. program for part time study are bound by all the rules set out henceforth. In particular, part time students should adhere to the schedule for the Qualifying Examination, Research Proficiency Examination, and Thesis Proposal, as explained in Sections 4.3.4, 4.3.5, and 4.3.6, unless a different schedule has been approved in writing by the Graduate Director.

4.3.11 Obtaining an M.S. Degree on the Way to Ph.D.

A Ph.D. student who has passed the Research Proficiency Examination can complete the requirements for an M.S. degree by satisfying the proficiency requirements and completing 31 credits of course work.

Passing the qualifying examination is considered to have satisfied the proficiency requirements. (Another way to satisfy these requirements is, of course, to take the required courses.)

At most 9 credits of seminars (excluding CSE 600), special topics courses, or CSE 593 (Independent study) can be included in the required 31 credits. A student who has switched from the M.S. program to the Ph.D. program can in addition use the previously earned credits of CSE 523/524 towards the aforesaid 9 credits. These 9 credits together with the RPE are considered to be equivalent to the Thesis Option in the M.S. program.

The remaining 22 credits required for the M.S. degree must be satisfied by taking technical graduate courses in computer science (*i.e.*, excluding courses such as CSE 523/524, CSE 587, CSE 593, CSE 596, CSE 599, CSE 696, CSE 698, CSE 699, seminars, and special topics).

5 Financial Support

First year Ph.D. students are generally supported on teaching assistantships; research assistantships are used to support continuing Ph.D. students. Some M.S. students are also supported in this way and a number of support opportunities in other university academic and administrative departments generally become available to Computer Science students each year. Assistants are assigned part-time duties in the undergraduate instructional program or in faculty-supervised research projects, but are still able to carry a full academic program. For a teaching assistant working 20 hours per week for the ten month academic year, the 2005/6 state rate for stipends will be $11,947 plus tuition remission and health benefits. In addition, the Department often adds to the state rate for incoming students in the form of a supplement for the first summer. Thereafter, Ph.D. students are eligible for additional financial support through their research advisor.

Summer stipends are funded almost entirely by research grants, and the availability of such support and the amount of the stipend may vary from year to year and depend on the student's research area and advisor. Most of Summer stipends are funded at about $7,000.

Assistants and fellows also receive tuition awards. Currently academic year tuition is $6,900 for the 12 credits for full-time graduate students who are New York State residents or $10,920 for out-of-state residents. Out-of-state residents who are able to do so (U.S. citizens and permanent residents) **must** become New York residents during the first semester of their graduate studies in order to remain eligible for tuition awards. A full tuition award covers 12 credits per semester until advanced status (G4) is assigned after completion of 24 graduate credits.[8] Thereafter the tuition award covers 9 credits per semester. Ph.D. students who reached the G5 status, must register for 6 credits to be considered full-time, and their tuition award is, accordingly, for 6 credits. A student **must** be registered full time (i.e., for 12, 9, or 6 credits, as appropriate) in order to receive tuition scholarship. Registering but not attempting a course (receiving the NR grade) is treated the same way as if the course was never registered for.

In the health insurance scheme, N.Y. State pays 90% of the premium for single employees or 75% of family coverage. The state estimates this benefit to be about 12% of salary for single employees.

Renewal of financial assistance each academic year depends upon the student making satisfactory progress towards the degree, and satisfactory fulfillment of the duties and responsibilities of any assistantship. The University limits renewals of annual teaching assistantships to three after the first year, for a total of four years. Beyond the fourth year, support is dependent on financial aid other than university

[8] Graduate courses (even in a different discipline) completed at other universities may count, if appropriate documentation is submitted to the Graduate School.

assistantships, such as research grants or fellowships. All offers and renewals of financial assistance are subject to Graduate School approval and the availability of funds.

All assistants who receive a stipend perform assigned duties in accordance with the following formulae: A student on a full assistantship devotes no more than 20 hours/week to his/her assigned duties during the academic year and 40 hours/week during the summer; A student on a fractional assistantship must give the corresponding fraction of full service each week.

A graduate student who is assigned to teaching duties (teaching assistant) is responsible to the faculty member in charge of the course to which he or she has been assigned. Duties will be specified by that faculty member and will usually include some or all of the following: lecturing to students on any subject pertinent to the course that will amplify the faculty member's lectures; answering student's questions concerning the course work; proctoring examinations; preparing solutions; grading of examinations; correction of homework assignments; supervision of laboratory sections; holding regular office hours. A document that describes the responsibilities of a teaching assistant appears on the Departmental Web site, `http://www.cs.sunysb.edu/`.

Students with teaching duties may register for CSE 698 (Practicum in Teaching) for up to 3 credits. The credits from CSE 698 do not fulfill department elective requirements, but do satisfy credit requirements for full time enrollment.

6 English Proficiency Requirements for Foreign Students

All students who are foreign nationals or have taken their higher education in a non-English speaking country must demonstrate proficiency in English. Admission to the Graduate School is contingent upon satisfactory fulfillment of this requirement. International students should request that the TOEFL or TSE scores be sent to the Foreign Student Office. All students must have a TOEFL score of 550 for admission, and all teaching or graduate assistants must have a TSE score of 240 **or** TOEFL score of 600. The award of a Teaching Assistantship will be contingent on the candidate's ability to speak English proficiently. All non-native English speakers will be required to pass a test of spoken English upon arrival at Stony Brook (the SPEAK test) before being assigned to classroom or other teaching duties. To be eligible for a teaching assistantship, a student must secure the score of 55 or higher on the speak test. Students who fail to achieve this score must take remedial courses ESL 591, ESL 596, or ESL 598 depending on their SPEAK test score. First year foreign students are advised to take full advantage of every opportunity to improve their fluency in English through frequent conversation with their American counterparts, and by enrolling in appropriate English language courses at Stony Brook.

Students on assistantship who cannot fulfill their obligations will fail to have their assistantships renewed; students who entered without support or with partial support will not be considered for full support the second year if they cannot assume the obligations of a teaching assistant.

7 Academic Standing and Probation

Students who do not maintain a cumulative graduate grade point average of 3.0/4.0 or better may either be *dismissed* or *placed on graduate probation* by the Department of Computer Science. *Students on probation may not be eligible for Assistantships or research support.* A student on probation must bring his or her grade point average up to 3.0/4.0 within one semester (or 12 credits) after being placed on probation; otherwise, (s)he will be subject to dismissal by the Department of Computer Science. Satisfactory progress by a student will be determined by the requirements written in the Graduate Bulletin and this handbook in effect at the time the student entered graduate studies in the Department, or at the discretion of the student, from current editions of the Bulletin and Handbook.

8 Ethics

Students are expected to maintain the highest standards of ethical behavior. A computer science professional is in a position to develop products upon which the health, wealth, and well being of the entire society rests. Unethical behavior cannot be tolerated in both the workplace and in graduate school. The Department will severely punish a student who it determines has cheated on an examination, turned in project material or a homework that is essentially the work of another student, or invaded in any way another student's files without permission. Punishments include failure in a course, referral of the case to the Dean's Office or expulsion from the Department.

9 Incompletes and Petitions for Late Withdrawal from a Course

The Graduate School enforces a demanding and well-defined policy with respect to petitions for late withdrawal from a course.

After the 15th day of classes, no course may be added or dropped. Should it become impossible for a student to complete a course for a reason such as illness or accident, he or she may petition the Dean of the college for a waiver of the deadline. Such a petition must be approved by both the Chairman and the Graduate Program Director of the Department.

A petition for a waiver of the deadline can be approved *only* if one of the following conditions is met:

- Employment requires that a student be elsewhere at the same time that the class meets. This must be documented by the employer.
- Illness or injury prevents the student's attendance at the class. Since illness generally isn't selective and normally incapacitates a person equally for all courses, it would be expected that the student would withdraw for *all* academic work, unless special circumstances can be demonstrated.

The student must have a statement from the instructor affirming that he is in good standing at the time the petition is presented, and that he has been in regular attendance at classes and is up to date in all assignments.

The Graduate Program Director is bound by pledge not to send up a petition that does not satisfy the spirit of these requirements. Note that the Dean reserves the right to review and reject a weak petition.

It is the responsibility of the student to remain alert to the approach of the add-drop deadline if in doubt about his or her ability to complete the requirements for a course. The student should arrange a meeting with the instructor to determine whether requirement (3) above will be satisfied in case of a late withdrawal. Such a meeting would be indispensable for those classes where no exams and few homework assignments will have been given before the add/drop deadline date. Please consult the Graduate School Bulletin, `http://www.grad.sunysb.edu/Bulletin/Regulations.pdf`, for additional regulations.

Students who receive an "I" grade for a course must complete that course's requirements before the middle of the following semester or petition for an extension until the end of that semester. After that time, the "I" will become and "I/F" and completion of the course is no longer possible.

10 Research Facilities

The Computer Science laboratories and department offices are headquartered in the Computer Science Building, with over 20,000 square feet of laboratory space. The Department's research facilities are extensive. All computer systems communicate through ethernets connected by a main 2.8GB/sec switched network. The public facilities include the systems in student offices and public laboratories. Student offices contain Sun Sparc systems running Solaris and Pentium PC's running Microsoft Windows. For maximum connectivity the PC's also have X-Window capability besides a rich mix of MS Windows software. The public laboratories contains Sun workstations running Solaris and Pentium PC's running Windows. In addition to these public systems the Department supports a web server for publishing research results and activities and various servers to support memory/compute intensive activities, remote access, and dialup access. In addition, there are several well-equipped research laboratories, each offering an array of specialized computer equipment.

The *Applied Algorithms Laboratory* applies advanced algorithmic techniques to applications areas such as biology, graphics, robotics, and financial engineering. We also use experimentation to advance our theoretical understanding of algorithms and discrete mathematics. Recent projects include Discropt – a time-sensitive environment for combinatorial optimization; Combinatorica – a library of over 450 Mathematica functions for combinatorics and graph theory, and the development of a new low-cost design methodology for fabricating DNA microarrays. The Laboratory includes a number of workstations for computation and software development.

The *Applied Logic Laboratory* concentrates on the areas of concurrent systems verification, databases, Web information systems, and logic programming. The projects in the lab include building advanced deductive engines for intelligent information systems; software environments for specifying and verifying the correctness of complex concurrent systems; and Web agents for data extraction and semantic information processing on the Web. The lab is large and comfortable, and the students have access to as much compute power as they need for their research.

The *Concurrency and Verification Laboratory* is a modern facility housing 15 high-tech personal work cubicles, each equipped with personal lighting, a desk, bookshelf, storage space, and a high-end workstation. The research conducted in the Lab is concerned with the specification, verification, implementation, and testing of concurrent, multithreaded systems, such as air-traffic control systems, factory-control systems, and the embedded software found in devices as diverse as cardiac pacemakers, automobiles, and consumer electronics. The approach advocated by the Lab for developing concurrent systems calls for the extensive use of formal, mathematically well founded techniques (such as operational semantics, process algebra, model checking, and abstract interpretation) and extensive software tool-development efforts in support of these techniques.

The *Design and Analysis Research Laboratory* concentrates on the development of methods and tools for constructing reliable and efficient computer systems. The laboratory has projects in modeling and specification, analysis and verification, design and optimization, code generation, and testing. These projects are for reactive systems, embedded systems, database applications, and Web information retrieval. The laboratory is spacious, comfortable, and generously equipped.

The *Experimental Computer Systems Laboratory* concentrates on the development of high performance networking, multimedia systems, distributed systems, and computer architecture. The laboratory has projects in systems for the real-time guarantee of bandwidth in switched and nonswitched networks, real-time searchable video server systems, router performance enhancements, Internet telephone service, wireless computing, distributed computing, computer clustering and robotics. The Experimental Computer Systems Laboratory contains over fifty Pentium PC systems running FreeBSD and Linux. The systems vary from Intel Pentium PC systems down to hand-held devices and small robots with eight-bit microprocessors. There are more than enough machines to guarantee a machine to every graduate student researcher who is interested in joining this laboratory.

The *File systems and Storage Laboratory* (FSL) concentrates on research in operating systems, with a focus on file systems, storage, security, and networking. Active projects include a plethora of stackable file systems: strong encryption; transparent compression; elastic quotas; tracing, logging, and snooping for IDSs; virus-detection; and load-balancing, replication, failover, and unification file systems. OS performance projects include a compiler and in-kernel run-time environment to compile and run unmodified Unix/C programs in the kernel. Scalability projects use clever algorithms in critical kernel sections, to allow firewalls and routers to handle millions of active connections on commodity OSs. FSL includes over 2TB of storage, a fast switched network, an 802.11 network, and lots of software for kernel development. FSL systems run many variants of Linux, *BSD, Solaris, AIX, HP-UX, and Windows, on a variety of architectures.

The *Information Systems Laboratory* concentrates on the development of advanced information systems. The laboratory has projects relating to systems for marine sciences and health care. These systems typically involve real-time sensor data processing and sensor fusion, with a goal of improved recognition and detection of conditions of interest to the subject domain (e.g., medical errors). Projects utilize techniques in advanced image processing, software engineering, user-interface design, and wireless communications.

The *Natural Language Processing Laboratory* concentrates on research on natural language processing and spoken dialog systems. Active projects include: the Stony Brook Dialog System project, reference resolution projects, information extraction from captioned photographs, and machine translation. The laboratory includes a number of PCs and portable devices for lab-related research.

The *Security Systems Laboratory* concentrates on network security issues. In medical, finance, E-commerce, E-banking applications, security and reliability of data and transactions is paramount. Without security, privacy cannot be assured during and after network transactions. Among the current projects in the Laboratory are efforts at intruder detection, mechanisms to prevent unauthorized access via service providing software systems, and intelligent compilers that detect and correct errors in code. To support this research, the Laboratory has a number of Pentium based Linux systems on a private network with compute and file servers.

The combined *Visualization and Virtual Reality Laboratories* concentrate on the development of software and hardware volume rendering techniques for use in scientific, biomedical visualization and virtual reality applications. These laboratories have projects in architectures for volume rendering, parallel methods for volume rendering, development of tools for data and information visualization, flow visualization, and the use of various input/feedback devices in virtual reality. The visualization labatory is a spacious, professional and comfortable environment for student researchers to expand the envelope in visualization and virtual reality.

The *Wireless Networking and Simulation (WINGS) Laboratory* is involved in research in wireless networking and mobile computing areas along with developing technologies for efficient large-scale simulation software for such systems. Much of the activity in the laboratory is related to developing protocols and applications for various multihop wireless networks such as mobile ad hoc networks or sensor networks. The laboratory has a large cluster of powerful workstations on a high-speed network to run simulations, a testbed of wireless handheld devices and a sensor network testbed.

11 Library

The Computer Science Library, located within the Computer Science Building, provides a pleasant environment for serious study and houses a collection of over 15,000 books, conference proceedings, technical reports, and 350 journal titles. In addition, the Library has an impressive digital collection of electronic books, journals, and databases that is accessible on-site as well as remotely. A full range of library services is available, including research assistance, interlibrary loan, and STARS, the Stony Brook Automated Retrieval System, which provides online access to the catalogs of all campus libraries from any workstation as well as access to other electronic collections. The Library is staffed by a professional librarian, a full-time clerk and several student assistants. Students are encouraged to make free use of these services and to seek the assistance of the librarian in the course of their research.

12 Faculty

The following is a brief listing of the faculty in the Computer Science Department. Details of the research projects supervised by each faculty member are available on their individual Web sites, which are reachable from the general Departmental Web site at `http://www.cs.sunysb.edu/`.

Arie Kaufman Ph.D., 1977, Ben-Gurion University, Israel; *Department Chairman*
Computer graphics, Visualization, Interactive systems, Computer architecture, Virtual reality, Multimedia.

Michael Ashikmin Ph.D., 2001, University of Utah.
Computer graphics, animation, visualization.

Leo Bachmair Ph.D., 1987, University of Illinois at Urbana-Champaign; *Undergraduate Director*
Computational logic, Automated deduction, Symbolic computation.

Hussein G. Badr Ph.D., 1981, Pennsylvania State University
Computer communication networks and protocols, Stochastic processes and queuing theory, Simulation, Performance evaluation, Modeling and analysis.

Michael Bender Ph.D., 1998, Harvard University
Algorithms; scheduling; data structures; randomization; asynchronous parallel computing.

Arthur J. Bernstein Ph.D., 1962, Columbia University; *Graduate Advisor*
Transaction processing and Workflow management systems, Correctness of concurrent systems.

Tzi-cker Chiueh Ph.D., 1992, University of California at Berkeley
Experimental computer systems, Computer architecture, Database systems, VLSI hardware design/CAD

W. Rance Cleaveland II Ph.D., 1987, Cornell University
Specification and verification formalisms, automated verification algorithms and tools, models of concurrent computation.

Samir Das Ph.D., 1994, Georgia Tech
Wireless networking, network routing.

Jie Gao Ph.D., Stanford, 2004.
Sensor networks, ad hoc wireless networks, computational biology, temporal databases, algorithms.

Radu Grosu Ph.D., 1994, Technical University of Muenchen
Reactive systems, model checking, design automation for embedded systems, software engineering.

Xianfeng David Gu Harvard, 2003
Computer Graphics, computational conformal geometry, harmonic shape analysis, geometry image.

Himanshu Gupta Ph.D., 1999, Stanford University
Databases, data mining, data warehousing.

Michael Kifer Ph.D., 1985, Hebrew University of Jerusalem, Israel; *Graduate Director*
Database systems; semantic Web; knowledge representation; logic programming.

Ker-I Ko Ph.D., 1979, Ohio State University
Computational complexity, Theory of computation, Computational learning theory.

Philip M. Lewis Ph.D., 1956, Massachusetts Institute of Technology
Transaction processing, Workflow management, Concurrent systems.

Y. Annie Liu Ph.D., 1996, Cornell University
Programming languages and compilers, Program optimization and analysis, Programming Environments, Reactive systems, Algorithm design.

Alexander Mohr Ph.D., 2002, U of Washington
Multimedia communication, error correction coding.

Klaus Mueller Ph.D., 1998, Ohio State University
Computer graphics, volume rendering, visualization, image-based rendering, virtual environments, medical imaging.

Manuel Oliveira Ph.D., 2000, University of North Carolina at Chapel Hill
Computer graphics, Image-based modeling and rendering, Virtual reality, Visualization, Image processing, Computer vision.

Hong Qin Ph.D., 1995, University of Toronto, Canada
Computer graphics, geometric modelling, CAD, virtual reality, animation.

C. R. Ramakrishnan Ph.D., 1995, SUNY, Stony Brook
Analysis & verification of software, Programming language implementation, Logic programming, Deductive databases.

I. V. Ramakrishnan Ph.D., 1983, University of Texas at Austin
Declarative programming, verification of software.

Dimitris Samaras Ph.D., 2001, University of Pennsylvania
Computer vision, computer graphics, medical imaging, animation and simulation, image-based rendering, physics-based modeling.

Radu Sion Ph.D., Purdue University, 2004
Information security, watermarking, distributed systems, wireless networks, grid computing, relational databases.

R.C. Sekar Ph.D., 1991, SUNY at Stony Brook
Computer security, distributed systems, programming languages/software engineering.

Steven Skiena Ph.D., 1988, University of Illinois at Urbana-Champaign
Computational biology, Algorithms, Discrete mathematics.

Scott A. Smolka Ph.D., 1984, Brown University
Computer-aided verification of safety-critical systems, Model checking, Process algebra, Visual design languages.

Eugene Stark Ph.D., 1984, Massachusetts Institute of Technology
Programming language semantics, Theory of concurrency, Formal methods, Operating systems.

Amanda Stent Ph.D., 2001, University of Rochester
Natural language processing.

Scott Stoller Ph.D., 1997, Cornell University
Distributed systems, Software testing and verification, Program analysis and optimization.

David S. Warren Ph.D., 1979, University of Michigan
Logic programming, Database systems, Interactive systems, Artificial intelligence, Natural language and logic.

Anita Wasilewska Ph.D., 1975, Warsaw University, Poland
Logic, Knowledge representation, Artificial intelligence.

Larry D. Wittie Ph.D., 1973, University of Wisconsin
Distributed shared memory architectures, Distributed operating systems, Massively parallel scientific algorithms, Computer networks and interconnection topologies, Computer architecture, Neural networks.

Erez Zadok Ph.D., 2000, Columbia University
Operating systems, file systems, storage, networking, software engineering, and security.

Affiliated Faculty

Esther Arkin Applied Mathematics and Statistics, Ph.D., 1986, Stanford University
Combinatorial optimization, network flows, computational geometry.

Susan Brennan Department of Psychology, Ph.D., Stanford University
Cognitive psychology, linguistics, human-computer interaction.

David Ferguson Technology and Society, Ph.D., Berkeley, 1980
Quantitative reasoning, problem solving, educational technologies, and decision making.

Jerome Liang Department of Radiology, Ph.D., The City University of New York
Medical imaging, image processing.

Joseph Mitchell Applied Mathematics and Statistics, Ph.D., 1986, Stanford University
Operations research, computational geometry, combinatorial optimization.

Yuanyuan Yang Department of Electrical and Computer Engineering, Ph.D., Johns Hopkins University
Parallel and distributed computing systems, High speed networks, Multicast communication, Optical networks, High performance computer architecture, Computer algorithms, Fault tolerant computing.

Emeritus Faculty

Herbert Gelernter Ph.D., University of Rochester
Artificial intelligence; knowledge-based, heuristic problem-solving systems; scientific applications.

Jack Heller Ph.D., Polytechnic Institute of Brooklyn
Database systems; office automation; visualization.

Theo Pavlidis Ph.D., 1964, University of California at Berkeley
Image analysis, Document processing including OCR, Computer graphics.

David R. Smith Ph.D., 1961, University of Wisconsin
Hardware description languages and synthesis, VLSI design tools, Experimental chip architectures.

Research Faculty

George W. Hart Ph.D., 1987, MIT
Geometric sculpture, Computational geometry

Rong Zhao Ph.D., 2001, Wayne State University; *Coordinator for CSE 523/4, SPIR program coordinator*
Content-based and semantic-based multimedia information retrieval; Web retrieval and mining; digital library; computer vision; medical imaging.

Lecturers

Thomas J. Cortina M.S., 1989, Brooklyn Polytechnic University
Programming methodology, Computer science education, Computer music. Database Systems, Modeling and Simulation.

Ahmad Esmaili M.S., 1981, Stony Brook University
Algorithms, Information systems.

Robert Kelly Ph.D., 1991, NYU
Information systems, Software engineering, Electronic commerce, Parallel programming.

Richard McKenna M.S., 2002, Stony Brook University
User interfaces, Visual languages.

Mark Tarver Ph.D., 1985, University of Warwick
Automated reasoning.

Shaunak Pawagi Ph.D., 1986, University of Maryland at College Park
Analysis of algorithms, Parallel computing.

Tony Scarlatos M.A., 1995, Adelphi University
Multimedia.

Michael S. Tashbook M.C.S., University of Virginia, 2003
Databases, bioinformatics, computer science education, formal methods, security, computational linguistics, theory of computation.

13 Computer Science Graduate Courses

13.1 Required Courses for the M.S. Non-Thesis Option

CSE 523 Intro. to Software Engineering & Project Plan (3 repetitive credits; Fall, Spring, and Summer)

A project in programming or digital system design that will extend over two consecutive semesters. The student starts the project in one semester by registering for CSE 523 and completes the project in a following semester by registering for CSE 524. Before the deadline date designated by the course instructor, the student will prepare a one to two page description of the work that is expected to be completed during the two-semester sequence. This description, reviewed and approved by the student's advisor, will reside in the student's file. Performance in completing the course requirements will be evaluated with reference to the implied promise contained. Amendments to the project description must be approved by the advisor. This course is graded separately from CSE 524.

CSE 524 Project Completion (3 credits; Fall, Spring, and Summer)

Implementation and completion of the project undertaken in CSE 523. Results are to reflect all aspects of large-scale problem-solving, including cost analysis, design, testing, and documentation. A final report documenting requirements, design, implementation and testing is required. When appropriate, a user's manual may be written. Prerequisite: CSE 523.

13.2 Regular Graduate Courses

CSE 502/ESE 545 Computer Architecture (3 credits; Fall)

Instruction pipelines and memory caches to improve computer performance. Instruction-level parallelism. Machines: superscalar versus VLIW. Cache and main memory hierarchy design tradeoffs. Compiler optimizations to speed pipelines. Low-power computer system design: Processor, OS, and compiler support. Graphics, DSP and media processor design. Disk I/O system design. Interconnections and networking. Introduction to parallel architecture. Advanced topics: Asynchronous microprocessors, FPGA-based reconfigurable computing, system on a chip, embedded processors, intelligent RAM and super-conducting computers. Prerequisite: CSE 320

ISE 503 Data Management (3 credits)

This course provides an understanding of the issues in managing database systems as an essential organizational resource. Students learn the enterprise data architecture components, data storage configurations, and information retrieval methods. It expands from the relational model to the multidimensional model, object-relational techniques, and web accessed data. The course includes concepts, principles, issues, and techniques for managing corporate data resources. Techniques for managing the design and development of large database systems including logical data models, concurrent processing, data distribution, data administration, data cleansing, and data mining. Students will use current methods and tools for database design and development.

CSE 504 Compiler Design (3 credits; Spring)

Advanced topics in compilation, including memory management, dataflow analysis, code optimization, just-in-time compilation, and selected topics from compilation of object-oriented and declarative languages. Prerequisites: CSE 304 and CSE 307.

CSE 505 Computing with Logic (3 credits)

The course explores logic-based computing and logic programming. It includes an introduction to programming in logic, covering basic techniques for solving problems in a logic programming system. Particular attention will be paid to user interface issues and how a logic system can provide a useful computing environment. The course covers implementation issues, emphasizing how a logic programming system generalizes both traditional programming language systems and traditional database systems. Prerequisites: CSE 214.

CSE 506 Operating Systems (3 credits; Spring)
This course is an in-depth study of important concepts and techniques found in modern computer operating systems. An undergraduate course in operating systems is a prerequisite. The course focuses on in-depth study of such important issues as virtual memory, filesystems, networking, and multiprocessor support, with an eye to recent directions in these areas. Textbook readings are supplemented, where appropriate, by papers from the research literature. An important part of the course is the case study of an actual operating system. Students study the source code for this operating system, and do programming exercises and projects that involve modifying the operating system and measuring its performance. Prerequisites: CSE 306

ISE 506 Quantitative Computer Architecture (3 credits)
Explores the physical structure of a computer; machine representation of information; architecture and organization of various mainframe, mini-, and microcomputers; primary and secondary storage; and input and output communication. Architectural choices are compared and used to determine resulting function and performance. Architectural trade-offs are also identified.

CSE 507 Introduction to Computational Linguistics (3 credits)
Overview of computational approaches to language use. Core topics include mathematical and logical foundations, syntax, semantics, and pragmatics. Special topics may include speech processing, dialog system, machine translation, information extraction, and information retrieval. Statistical and traditional approaches are included. Students will develop familiarity with the literature and tools of the field. Prerequisites: CSE 537; CSE 541 recommended

CSE 508 Network Security (3 credits)
Principles and practice of Computer Network Security. Cryptography, authentication protocols, public key infrastructures, IP/www/E-commerce security, firewalls, VPN, and intrusion detection. Prerequisites: CSE 310 or CSE 346

CSE 509 System Security (3 credits)
Principles and practice of building and administering secure systems. Authentication and access control. Operating system security. Program security. Key management. Information flow. Assurance. Vulnerability analysis and instrusion detection. Prerequisites: CSE 306 or CSE 376

CSE 510 Hybrid Systems (3 credits; Spring)
Hybrid Systems combine discrete state-machines and continuous differential equations and have been used as models of a large number of applications in areas such as real-time software, embedded systems, robotics, mechatronics, aeronautics, process control and biological systems. The course will cover the state-of-the-art of modeling, design and analysis of hybrid systems. Prerequisite: limited to CSE graduate students; others, permission of instructor

CSE 515 Introduction to Transaction Processing Systems (3 credits; Fall)
Discusses transaction processing systems. Topics covered include: models of transactions, including nested transactions and workflows; architectures of transaction processing systems, including client-server, two-tiered and three-tiered architectures; concurrency controls for conventional and relational databases including two-phase locking and the SQL isolation levels; logging and recovery; distributed transactions including the two-phase commit protocol; replication; Internet commerce,

including encryption, the SSL and SET protocols, goods atomicity and electronic cash. Prerequisites: CSE 305

ISE 516 Systems Engineering Principles (3 credits)
An introduction to the full range of system engineering concepts, tools, and techniques. These elements are applied to both large-and small-scale projects. The course provides a review of the stages of an integrated, top-down, life-cycle approach to design engineering - from analysis of customer requirements to maintenance and support, from definition of system operational concepts through material disposal and recycling. It also includes a review of various disciplines applicable to the deisgn process, including reliability and maintainability engineering, human factors, safety, logictics engineering, quality engineering, and value/cost engineering. The course also includes a treatment of curcial management issues, such as the planning and development of System Engineering Management Plans (SEMPs), work breakdown structures (WBSs), cost projections, and supplier selection and management.

This course cannot be used to satisfy the M.S. or Ph.D. requirements in Computer Science.

ISE 517 Human Factors in System Engineering (3 credits) The course focuses on techniques to integrate human factors into the design of systems so that the systems match human abilities and limitations. It addresses techniques to translate system requirements into project specific design requirements. The course also describes the effect of human factors on each stage of project development.

This course cannot be used to satisfy the M.S. or Ph.D. requirements in Computer Science.

CSE 526 Principles of Programming Languages (3 credits; Spring)
Programming language concepts and design, with emphasis on abstraction mechanisms. Topics include: language paradigms (procedural, object-oriented, functional and logic), language concepts (values, bindings, types, modules), and foundations (lambda calculus, denotational semantics). Examples will be drawn from several representative languages, such as C, Java, Standard ML, and Prolog. Prerequisite: CSE 307.

CSE 527 Introduction to Computer Vision (3 credits)
Introduction to basic concepts in computer vision. Low level image analysis, image formation, edge detection, segmentation. Image transformations for image synthesis methods for 3D scene reconstruction, motion analysis, object recognition. Prerequisites: CSE 214, linear algebra, calculus, C/C++ proficiencies

CSE 528 Computer Graphics (3 credits; Fall)
This course emphasizes a hands-on approach to the use of computer graphics. The topics covered include: Models, picture description, and interaction; windowing, clipping, panning and zooming; geometrical transformations in 2D and 3D; algorithms for raster displays (scan-line conversion, polygon fill, polygon clipping, etc.); hidden line and hidden surface removal, shading models; user interaction. The students will implement a substantial graphics application program. Prerequisite: CSE 328.

CSE 529/AMS 553 Simulation and Modeling (3 credits; Fall)
A comprehensive course in formulation, implementation, and application of simulation models. Topics include data structures, simulation languages, statistical analysis, pseudo-random number generation and design of simulation experiments. Students will apply simulation modeling methods to problems of their own design. Prerequisites: CSE 214 & (AMS 310 or AMS 507).

CSE 530 Geometric Foundations for Graphics & Visualization (3 credits; Spring)

This course will focus on mathematical tools, geometric modeling techniques, and fundamental algorithms that are relevant to graphics, visualization, and other visual computing areas. The goal is to provide graduate students a comprehensive knowledge on geometric concepts and demonstrate the significance of these mathematical tools and geometric algorithms in graphics and relevant areas. Course topics include geometric algorithms for both polygonal and curved objects, theory of parametric and implicit representations, modeling methods of curves, surfaces, and solids; in-depth spline theory, rudiments of wavelet theory and multi-resolution shape representations, differential geometry fundamentals, and other sophisticated topics and latest advances in the field. Prerequisites: CSE 328 and CSE 332.

CSE 532 Theory of Database Systems (3 credits; Fall and Spring)

The course will cover advanced topics in modern database systems, including object-oriented databases, rule-based databases, temporal and active databases, parallel and distributed databases, distributed object model, data mining, on-line analytical processing, data warehousing, multimedia databases. Prerequisite: CSE 305

CSE 533 Network Programming (3 credits; Fall and Spring)

Socket and client-server programming, remote procedure calls, data compression standards and techniques, real-time protocols (e.g, audio chat, etc.) security and cryptography (specifically, application layer security issues such as authentication), web-related programming (CGI, Java/Java Script, HTTP, etc.,) network management (SNMP, dynamic, CORBA-based management). Prerequisite: CSE 306 and CSE 310

CSE 534 Fundamentals of Computer Networks (3 credits)

Data Transmission: Introduction to Fourier analysis; Data coding & signals, noise, Nyquist's Theorem, Shannon's Theorem, bandwidth/baud rate/bit rate; Data multiplexing techniques (ASK, FSK, PSK); Modems, and modem standards & techniques (e.g: Trellis coding, etc), Data Link Layer: Protocols; Error detection & Correction; Flow control; etc, Network Layer: Protocols; Routing algorithms; Flow & congestion control; Error detection & correction; etc, Transport Layer: Protocols; Error detection & correction; Congestion control; etc., Quality-of-Service issues at the Network & Transport layer, Local Area Networks (including MAC; High-Speed LANs; Wireless LANs; Bridges; etc), High-Speed Networks (BISDN; ATM standard; etc)

CSE 535 Asynchronous Systems (3 credits)

Discusses asynchronous systems, their description using concurrent and distributed programming languages and their verification. Topics include concurrent programming using shared memory and message passing, formal semantics of communication, reliability and concurrent algorithms. Prerequisite: CSE 306.

CSE 536 Introduction to User Interface Development (3 credits)

Survey of user-interface systems: includes command language, windowing, multiple input/output devices, architecture of user interface management systems, toolkits for designing user-interface, human factors, standards, visual languages. The course also includes discussion of emerging technologies, such as systems for cooperative work, physically distributed user-interfaces, parallelism and user-interfaces, and virtual reality. A substantial project requiring the design, implementation, and evaluation of a user-interface will be required.

CSE 537 Artificial Intelligence (3 credits; Fall)

A comprehensive introduction to the problems of artificial intelligence and techniques for attacking them. Topics include: problem representation, problem-solving methods, search, pattern recognition, natural language processing, learning, expert systems, AI programming languages and techniques. Covers both theoretical methods and practical implementations. Prerequisites: MAT 371 or CSE 541

CSE 540 Theory of Computation (3 credits; Fall)
Models of computation: finite-state machines, stack machines, Turing machines, Church's thesis; Computability theory: halting problem and unsolvability, introductory recursion theory; Complexity theory: complexity measures, time and space hierarchy, NP-complete problems. Prerequisites: CSE 303.

CSE 541 Logic in Computer Science (3 credits; Spring)
A survey of the logical foundations of mathematics and the relationships to computer science. Development of propositional calculus and quantification theory; the notions of a proof and of a model; the completeness theorem. Prerequisite: MAT 313 and CSE 213

CSE 542 Speech Processing (3 credits; Spring)
Introduction to the collection and anaylsis of speech data for speech processing Includes a brief introduction to corpus linguistics. Students will learn the range and types of spoken language collections, and will learn how to analyze speech data using the Praat tool. Introduction to speech recognition. Students will learn basic technologies for speech recognition, using the Hidden Markov Model Toolkit (HTK). Introduction to concatenative text-to-speech synthesis Students will learn the basics of text-to-speech synthesis (TTS), as well as current technologies for concatenative TTS. The TTS system Festival (or its Java version, FreeTTS) will be used. Integration of speech recognition and TTS into other technologies (by means of, e.g., VoiceXML and/or the speech SDKs under development by Microsoft, Sun (Java), and IBM) will also be discussed.
Prerequisites: Knowledge of C/C++, Java, and/or Perl preferred. Knowledge of phonetics a plus.

CSE/AMS 547 Discrete Mathematics (3 credits; Spring)
This course introduces such mathematical tools as summations, number theory, binomial coefficients, generating functions, recurrence relations, discrete probability, asymptotics, combinatorics and graph theory. Alternative: AMS 506. Prerequisite: AMS 301

CSE 548/AMS 542 Analysis of Algorithms (3 credits; Fall)
Techniques for designing efficient algorithms, including choice of data structures, recursion, branch and bound, divide and conquer, and dynamic programming. Complexity analysis of searching, sorting, matrix multiplication, and graph algorithms. Standard NP-complete problems and polynomial transformation techniques. Prerequisite: CSE 373

CSE 549 Computational Biology (3 credits; Fall)
This course focuses on current problems in computational biology and bioinformatics. Our emphasis will be algorithmic, on discovering appropriate combinatorial algorithm problems and the techniques to solve them. Primary topics will include DNA sequence assembly, DNA/protein sequence assembly, DNA/protein sequence comparison, hybridization array analysis, RNA and protein folding, and phylogenic trees.

CSE 555/AMS 545 Computational Geometry (3 credits; Fall)
We study the fundamental algorithmic problems associated with geometric computations, including convex hulls, Voronoi diagrams, triangulation, intersection, range queries, visibility, arrangements, and motion planning for robotics. Algorithmic methods include plane sweep, incremental insertion, randomization, divide-and-conquer, etc. Prerequisite: CSE 373 or CSE 548.

CSE 564 Visualization (3 credits; Spring)
The course emphasizes a hands-on approach to scientific visualization. Topics include: traditional visualization, the visualization process, visual perception, basic graphics and imaging concepts, volume and surface visualization, volume graphics, visualization of sampled and computed data, case studies, and visualization systems. This course presents introductory as well as more advanced topics on visualization, and students will have the opportunity to further explore a topic of their choice by ways of a final programming project.

CSE 581 Advanced Topics in Computer Science (3 repeatable credits)
Same as CSE 590

CSE 582 Advanced Topics in Computer Science (3 repeatable credits)
Same as CSE 590

CSE 587 Proficiency Requirement in Computer Science (2 credits; Spring and Fall)
Students can get credit for an undergraduate course by registering for CSE 587. The syllabus of the undergraduate course must specify additional work that graduate students must do in order to pass the course. Graduate students taking an undergraduate course under the CSE 587 number must be graded separately from the undergraduate students. Consult Section 4.2 for additional restrictions regarding the use of this option.

CSE 590 Advanced Topics in Computer Science (3 repeatable credits)
An advanced lecture course on a new topic in computer science. This course is primarily designed for M.S. students, but can be taken by Ph.D. students as well. Semester supplements to this Bulletin contain specific description when course is offered. May be repeated for credit as the topic changes, but cannot be used more than twice to satisfy the CSE major requirements for the M.S.

CSE 591 Advanced Topics in Computer Science (3 repeatable credits)
Same as CSE 590

CSE 592 Advanced Topics in Computer Science (3 repeatable credits)
Same as CSE 590

CSE 593 Independent Study in Computer Science (1–9 repetitive credits – G4 students, 1-3 credits all others; Fall, Spring, and Summer)
Students can register for this course in order to conduct research or participate in a project under the supervision of a Computer Science faculty member. The student must prepare a description of the project to be taken and submit it before the add/drop deadline to the project sponsor. The description will reside in the student's file. Both M.S. and Ph.D. students can take this course. This course cannot be taken as part of M.S. Thesis research — use CSE 599 in that case. Ph.D. students may take CSE 593 for any kind of research or project work prior to advancement to candidacy (G5 status). After the advancement, CSE 699 should be used to conduct Dissertation Research. G4 students can register for up to 9 credits of CSE 593 in any semester. G1–G3 students can register for up to 3 credits of CSE 593 in any semester.

CSE 596 M.S. Internship in Research (1–3 credits; Fall, Spring and Summer)
Participation in private corporations, public agencies, or non-profit institutions. Students will be required to have a faculty coordinator as well as a contact in the outside organization, to participate with them in regular consultations on the project, and to submit a final report to both. On completion of internship, the outside contact should provide the faculty coordinator with a letter evaluating student's performance during the internship period. At most 2 credits can be accepted towards the

M.S. degree. Note that Curricular Practical Training can be taken in conjunction with CSE 596 at most once.

CSE 599 M.S. Thesis Research (1–3 credits; Variable and repetitive credit)
This course can be used *only* for M.S. Thesis research; non-thesis research should be done under the designation of CSE 593: Independent Study. M.S. students who wish to enroll in CSE 599 for any number of credits must prepare a 1–2 page description of the work to be completed. The description must be approved by the research advisor, signed by both student and advisor, and reside in the student's file. Amendments to the proposal must be approved by the advisor. Up to 9 credits of CSE 599 can be counted towards the 31 credits that are required for graduation.

CSE 600 Topics in Modern Computer Science (1 credit, Fall and Spring – credit cannot be used towards graduation requirements)
A survey of current computer science research areas and issues. This seminar is comprised of lectures by faculty members and visitors, selected readings and introductory research problems. Possible topics include algorithms, systems, robotics, artificial intelligence, databases, graphics, networks, and more.
Cannot be taken to satisfy M.S. graduation requirements.

13.3 Advanced Courses

The following are courses normally considered appropriate for the Ph.D. program although they can be elected by M.S. students with permission of the advisor. Many are offered only every second year.

CSE 601 Advanced Image Processing (3 credits)
Modern approaches to image processing, statistical image formation and image models, image restoration, reconstruction and segmentation, applications to medical imaging.

CSE 602 Advanced Computer Architecture (3 credits)
The focus will be on the architectural rather than micro-architectural issues, and a systems approach to computer architecture taking into account the interaction between the architecture and the compiler, operating system, database and networking. The course will start with superscalar/VLIW processor architecture, and proceed to memory hierarchy, storage systems, network hardware, graphics processor, and database machines. The emphasis will be hands-on evaluation of architectural ideas, the exploration of software/hardware design tradeoffs, and the articulation of experimental procedures and performance analysis. A publication-quality class project will be required. Prerequisite: CSE 502.

CSE 605 Performance Evaluation of Computer Systems (3 credits)
The purpose of this course is to provide background and training in understanding and evaluating performance of computer systems, including centralized, distributed, parallel, client/server based systems, and computer communication networks. The goal is to develop a perspective on how the performance of computer systems or networks should be evaluated in order to decide on various design alternatives. The course will include various analytical techniques, mainly based on Markov models and queueing theory, and simulation modeling.

CSE 608 Advanced Computer Security (3 credits)
Advanced course on principles and practice of engineering secure information systems. Topics covered include threats and vulnerabilities, counter measures, legal and policy issues, risk management and assurance. In-depth coverage of various research problems, which vary from one offering of the course to another.

CSE 610 Parallel Computer Architectures (3 credits)
Parallel computer systems; important parallel applications; parallel computation models; interconnection networks; SIMD and MIMD architectures; hybrid architectures; memory management; cache coherence; distributed shared memory; synchronization methods; operating systems; compilers; programming tools. Prerequisite: CSE 502.

CSE 611 Transactions Processing (3 credits)
An advanced course in transaction processing systems covering the latest developments in the area. Topics include stable storage, distributed database systems, commitment protocols, failures, replication, and advanced models of transactions. Prerequisite: CSE 515.

CSE 612 Advanced Visualization (3 credits)
Discusses advanced concepts in the area of volumetric data modeling and visualization. Topics included are: Visual exploration of multi-variate and multi-dimensional datasets on regular and irregular grids, modeling of natural phenomena and simulation of realistic illumination, volumes as magic clay for sculpting and deformation effects, non-photorealistic rendering for illustration and artistic works, information-centric exploration of large datasets, and exploitation of hardware for acceleration. The course strives to provide a snapshot on the current state of the art and will be supported mostly by recent research papers. Students will expand on a topic of their choice by completing an individual project. Prerequisite: CSE 564.

CSE 613 Parallel Programming (3 credits)
Algorithms and techniques for programming highly parallel computers. Trends in parallel and distributed computing; shared address space and message passing architectures; design issues for parallel algorithms; converting sequential algorithms into equivalent parallel algorithms; synchronization and data sharing; improving performance of parallel algorithms; interconnection network topologies, routing, and flow control; latency limits on speedup of algorithms by parallel implementations. Prerequisite: CSE 502 and (consent of instructor or CSE 610).

CSE 614 Advanced Programming Languages (3 credits)
Selected topics on advanced programming languages technology. Program analysis and transformation, program optimization, and program manipulation systems. Very high-level and declarative languages such as sets and relations based languages and deductive and object-oriented languages. Prerequisite: CSE 526.

CSE 615 Advanced Computer Vision (3 credits)
Survey of methods used for the analysis of images by computer, including computer vision and pattern recognition. Topics to be covered are image formation, image segmentation and edge detection, binary images and shape analysis, shape from shading, motion field and optical flow, surface inference, classification techniques. Prerequisite: B.S. degree in Computer Science, Engineering, Mathematical or Physical Sciences.

CSE 616 Digital Multimedia Systems (3 credits)
In-depth survey of multimedia computing, including media conversion, data compression, multimedia data representation and modeling, authoring techniques, audio and video editing, 2D and 3D animation, media synchronization, distributed multimedia, and advanced application development. Prerequisite: CSE 333 or CSE 536

CSE 618 Advanced Computer Graphics (3 credits)
Advanced topics in rendering and modeling realistic 3D imagery, including texture mapping and

synthesis, radiosity, amorphous phenomena, artificial life, and animation. Further contents include introductions to free-form curves and surfaces, volume rendering, and image-based rendering.

CSE 620 Virtual Reality (3 credits)
Practical issues in the design and implementation of virtual environments. Topics include: system requirements, transformations, user-interaction models, human vision models, input/output devices and techniques, tracking systems, augmented reality, and virtual-reality applications. The course will involve a substantial programming project to implement an immersive virtual reality system. Prerequisite: CSE 328 or 528 or 332 or 564

CSE 621 Physics-Based Modeling for Visual Computing (3 credits)
A unified approach to various fields such as graphics, visualization, computer-aided geometric design, biomedical imaging, vision, and virtual environment. The course will explore select research topics centered on physics-based modeling methodology and associated computational methods for theoretical and practical problems in widespread areas of visual computing. The emphasis will be on: geometric and solid modeling, geometric design techniques, wavelets and multi-resolution analysis, deformable models based on mathematical physics, variational analysis, optimization methods, numerical simulation with finite-difference and finite-element algorithms, differential equations for initial-value and boundary-value problems, force-driven interaction with constraints, dynamic sculpting system, and a large variety of applications for visual computing. Prerequisites: CSE 528.

CSE 622 Advanced Database Systems (3 credits)
The course will cover selected topics on the cutting edge of database technology, such as deductive database query languages and systems, object-oriented data models, persistent programming languages, heterogeneous databases, and advanced transaction models.
Prerequisite: CSE 532.

CSE 624 Advanced Operating Systems (3 credits)
This is a survey of modern operating system techniques, especially those needed for distributed operating systems. Topics include: network topologies, interprocess communication, failure detection and system recovery, local kernel functions, global network services, location transparency, large network constraints, distributed control algorithms (synchronization, configuration, deadlock detection, and searches), and existing distributed operating systems.
Prerequisite: CSE 506.

CSE 625 Advanced Asynchronous Systems (3 credits)
Formal specification and verification of asynchronous systems. Topics include concurrent programming, process algebras, logics for describing the properties of concurrent systems, and formal semantics of communication. Prerequisite: CSE 535 or permission of instructor.

CSE 626 Switching and Routing in Parallel and Distributed Systems (3 credits)
This course covers various switching and routing issues in parallel and distributed systems. Topics include message switching techniques, design of interconnection networks, permutation, multicast and all-to-all routing in various networks, nonblocking and rearrangeable capability analysis, and performance modeling. Prerequisite: ESE 503 and 545 or CSE 502 and 547, or permission of instructor.

CSE 628 Natural Language Processing (3 credits)
A survey of computational approaches to natural language processing issues in phonology, morphology, syntax, semantics and pragmatics. Topics to be discussed include natural language parsing algorithms, generation algorithms, and knowledge representations. Models for speech recognition

systems, story understanding systems, and natural language front-ends to databases and other application programs will be investigated. Prerequisite: CSE 537.

CSE 630 Theory of Computational Complexity (3 credits)
Machine-based polynomial-time complexity theory, including nondeterministic computation, probabilistic computation, time and space tradeoff, and complexity hierarchy; applications to related areas such as combinatorial algorithms and cryptography. Prerequisite: CSE 540 or CSE 548.

CSE 631 Advanced Logic in Computer Science (3 credits)
The course may include the following: deductive theorem proving (resolution, sequent-style calculi, natural deduction), inductive theorem proving, equational reasoning (rewrite systems), non-classical logics (modal logics, intuitionistic logic). Prerequisite: CSE 541.

CSE 633 Computability and Undecidability (3 credits)
Computability theory based on Turing machines and recursive functions; proof by diagonalization and reducibility; unsolvable problems in set, group, number, and language theory; reducibility orderings and degrees of unsolvability; priority methods and Post's problem. Prerequisite: CSE 540.

CSE 634 Data Mining Concepts and Techniques (3 credits)
Data Mining is a new, promising, and flourishing interdisciplinary field drawing work from areas including database technology, artificial intelligence, machine learning, pattern recognition, high-performance computing, and data visualization. It focuses on issues relating to the feasibility, usefulness, efficiency, and scalability of techniques for automated extraction of patterns representing knowledge implicitly stored in large databases, warehouses, and other massive information repositories. The course gives a broad, yet in-depth overview of the field of data mining and presents one or two techniques in detail. Prerequisite: CSE 305 or CSE 532

CSE 636 Analysis and Synthesis of Computer Communication Networks (3 credits)
Analysis of message queueing and buffering in computer networks. Survey of OSI layered architecture. Network topology. Local, metropolitan, and wide area networks. Circuit and packet switching techniques. High Speed and Lightwave Network Concepts: Synchronous Optical Network (SONET), Fiber Distributed Data Interface (FDDI), Distributed Queue Dual Bus (DQDB-QPSX), Integrated Services Digital Networks (ISDN), Broadband-ISDN, Asynchronous Transfer Mode (ATM). Prerequisite: CSE 533

CSE 637 Program Semantics and Verification (3 credits)
Formal approaches to defining semantics of programming languages: denotational, operational, axiomatic, and transformational semantics. Formal systems for program verification. Logics of program, type theory, lambda calculus. Further topics selected from term rewriting approach to proving properties of data types, and semantics and verification of languages with concurrent and parallel constructs. Prerequisite: CSE 541

CSE 638 Advanced algorithms (3 credits)
This is an advanced course in the design and analysis of combinatorial algorithms, focusing on recent material and special topics, including: randomized algorithms, approximation algorithms for NP-complete problems, string algorithms, amortized analysis of data structures, and heuristic methods such as simulated annealing. Material will be selected to have little or no overlap with traditional introductory algorithms courses. Prerequisite: CSE 548.

CSE 690 Advanced Topics in Computer Science (3 repeatable credits)
An advanced lecture course on a new topic in computer science. This course is primarily designed

for Ph.D. students, but can be taken by M.S. students as well. Semester supplements to this Bulletin contain specific description when course is offered. May be repeated for credit as the topic changes, but cannot be used more than twice to satisfy the CSE major requirements for the M.S.

CSE 691 Advanced Topics in Computer Science (3 repeatable credits)
Same as CSE 690

CSE 692 Advanced Topics in Computer Science (3 repeatable credits)
Same as CSE 690

13.4 Ph.D. Teaching and Research Experience

CSE 696 Ph.D. Internship in Research (1–3 credits; Variable and repetitive credit)
See CSE 596 for similar description.

CSE 698 Practicum in Teaching (1–3 credits; Spring and Fall, Variable and repetitive credit)
Normally taken by Ph.D. students in their first year in conjunction with a TA assignment. However, an M.S. student who is appointed as a TA can take this course as well.

CSE 699 Ph.D. Dissertation Research (1–9 credits; Variable and repetitive credit)
This course is taken by advanced Ph.D. students when they conduct research towards their thesis. Only Ph.D. students who have been advanced to candidacy (G5 status) can take this course. Students who have the G3 or G4 status and participate in a research project with their advisor can register for CSE 593: Independent Study.

13.5 Seminars and Special Topics Courses

These are offered irregularly in the Spring or Fall and normally cater to the needs of Ph.D. and research students. The seminars can be taken for 1 credit and the special topics courses for 2 credits. The list of all seminars and special topics courses appears in Table 1.

13.6 Summer Research

CSE 800 Summer Research (0–9 credits; Variable and repetitive credit)
Continuing students who have a TA, GA, or RA appointment during the Summer are strongly encouraged to register for 0 credits of CSE 800 during the summer. The Graduate School advises this for reasons related to tracking federal grants, tax issues, and Homeland Security.

Table 1: Seminars and Special Topics

Seminar	**Special topics**	**Area**
CSE 640	CSE 665	Theory of Computing
CSE 641	CSE 666	Logic in Computer Science
CSE 642	CSE 667	Algorithms
CSE 643	CSE 668	Concurrency
CSE 644	CSE 669	Database
CSE 645	CSE 670	Languages
CSE 646	CSE 671	Artificial intelligence
CSE 647	CSE 672	Image processing
CSE 648	CSE 673	Graphics
CSE 649	CSE 674	Operating Systems
CSE 650	CSE 675	Architecture
CSE 651	CSE 676	Applications
CSE 652	CSE 677	User Interfaces
CSE 653	CSE 678	Virtual Reality
CSE 654	CSE 679	Visualization
CSE 655	CSE 680	Modeling and Simulation
CSE 656	CSE 681	Computer Vision
CSE 657	CSE 682	Design and Analysis
CSE 658	CSE 683	Mobile and Wireless Networking
CSE 659	CSE 684	Computer Security
CSE 660	CSE 685	Media Networks
CSE 661	CSE 686	Data Privacy
CSE 662		Seminar in Computer Science I
CSE 663		Seminar in Computer Science II
CSE 664		Seminar in Computer Science III