2007 – 2008

Computer Sciences Graduate Guidebook

University of Wisconsin - Madison

Madison, Wisconsin

August 2007

This document is also available in the World-Wide Web (`http://www.cs.wisc.edu/~pubs/grad-guidebook`).

Contents

Introduction

The Department of Computer Sciences at the University of Wisconsin - Madison strives to maintain the highest standards in education and research. Through our educational programs and our research we have made significant contributions to the field of computer science. Our faculty and students have earned high regard nationally and internationally for their achievements. In both education and research, we stress both theoretical and experimental methods for solving fundamental as well as practical problems.

The Department offers graduate programs leading to the Master of Science degree and Doctor of Philosophy degree in Computer Sciences. This guidebook is intended for Computer Sciences graduate students who are developing programs leading to one of these degrees. The *Graduate School Catalog* and the *Graduate Student Handbook* provide additional, essential information regarding general University requirements. The information in this guidebook should also be supplemented by individual consultation with the Graduate Advising Committee so that both individual needs and interests, and all degree requirements, are met.

Additional information is available via the Department's World-Wide Web (WWW) page.[1] Students may also wish to consult the Graduate School's WWW page.[2] Foreign students may find useful the Graduate School document *Essential Information for Newly Admitted International Students.*

[1] http://www.cs.wisc.edu/
[2] http://www.wisc.edu/grad/

Admission

Applications for admission are accepted from students with undergraduate majors in many different fields, including computer science, mathematics, engineering, psychology, linguistics, economics, physics, philosophy, and business. Minimally, the student should have had some programming experience (including courses in data structures and machine organization) and should have had a year of college-level mathematics at the level of calculus or above. If the intention is to concentrate in numerical areas, the student should have substantially more background in mathematics. Good students can sometimes be admitted to graduate work in computer science without having had programming experience or mathematics courses as described, but they will have to make up the deficiencies.

Professional computer scientists who wish to enter the Department to pursue graduate work in their field will be given special consideration. In some cases work experience at a high professional level can be substituted for certain formal academic requirements.

Departmental requirements for admission are more stringent than those of the Graduate School. These additional requirements, to which there can be no exceptions, are listed below. Applications are normally accepted for *fall semester admission only.* Completed applications are due no later than *December 31.* Application for admission to the graduate program must be done online via the webpage at http://www.cs.wisc.edu/grad_admissions.html.

Letters of Recommendation

All applicants, whether requesting financial aid or not, must have forwarded to the Department three letters of recommendation.

GRE Scores

All applicants must submit Graduate Record Examination (GRE) scores for the general (verbal and quantitative examinations) and for the advanced examination in any field in which they have majored or are otherwise specially qualified. An applicant should sign up to take GRE's at the earliest possible date. To make arrangements to take a GRE, applicants should write to: Educational Testing Service, Box 955, Princeton, New Jersey 08540. Processing of scores takes approximately six weeks. The date when GRE's were taken or when they will be taken should be indicated on the top of the application for graduate study.

Official School Records

Official transcripts are required from each institution where the applicant did prior academic work. If an institution does not issue official transcripts, a letter from an administrator of the institution should be sent including: (a) year of admission, (b) number of years enrolled at the institution, (c) reference to quality of work (analysis of grading system), (d) evidence that examinations were passed, (e) diploma certifying degree, class, and year, and (f) General Certificate of Education or equivalent.

TOEFL and TSE Scores (Foreign Students only)

Foreign applicants whose native language is not English are required to take the Test of English as a Foreign Language (TOEFL) and the Test of Spoken English (TSE). Since the GRE, TOEFL and TSE are all offered by the Educational Testing Service, one letter to the Service should suffice to inquire about all required examinations.

Financial Aid

Financial aid is available to graduate students in the form of teaching assistantships, research assistantships, University fellowships, National Science Foundation fellowships, and research assistantships or fellowships sponsored by various companies. The Graduate School has placed on the World-Wide Web information about graduate-student financial support.[3] All applications, whether or not they involve requests for financial aid, can be for the fall semester only and must be completed by December 31.

Outstanding students are strongly encouraged to apply to the National Science Foundation for an NSF Graduate Fellowship[4] before the November deadline. For information, write National Science Foundation, Forms and Publication Unit, Room P15, 4201 Wilson Blvd., Arlington, VA 22230.

Most teaching assistantships are awarded at approximately the same time that a student is admitted into the Department. Occasionally, there are last minute temporary appointments also available. Students interested in such temporary appointments should contact the Assistant to the Chair.

To receive a new or renewal appointment as a research assistant (RA) or teaching assistant (TA), a student must be making satisfactory academic progress (SAP) in the Department. See the chapter below on SAP requirements for details.

It is often possible for a Computer Sciences graduate student to find, after arrival on campus, a part-time job which pays well enough to support the student while in graduate school. There are several thousand computers on the Madison campus, and many of them require programmers. These computers are owned and operated by a wide variety of departments and projects on the campus, and the kind of programmer needed varies greatly. There is no one way to find out about all of these jobs. Students should also consider contacting the Division of Information Technology (DoIT)[5] and individual departments on campus for information.

[3] http://www.wisc.edu/wiscinfo/academics/admissions
[4] http://www.nsf.gov/
[5] http://www.wisc.edu/doit

M.S. Degree Requirements

In addition to the requirements of the Graduate School[6] — see their publication *Expecting Your Master's Degree?* [7] — and the University in general, the Computer Sciences Department has the following requirements for the M.S. degree:

I. Courses

The student must receive at least 24 credits in courses numbered 400 or above, with an average grade of at least B in these courses (the grade average enforced for admissions starting after January 1, 1997). Fifteen of these credits must be received for *core* courses: CS courses numbered 700-889 (excluding 799). No courses – including CS 837, 838, and 880 – may be counted more than once, though up to six credits of CS 790 may be counted as core credit if a Master's thesis or project is filed with the Department (see next section). A qualifying core course for which a grade of S is received may be counted only for non-core credit. Also, all sections of CS 837, 838 and 880 will be considered non-core courses, unless designated as a core section by their instructor. Non-core courses will not be counted if a grade of S is received. At its discretion, the Graduate Advising Committee (GAC) may declare an individual section of CS 837, 838, or 880 exempt from the repetition restriction.

During registration week of the first semester of the student's full- or part-time graduate tenure in the Department, the student must, in consultation with GAC, file with the Department a formal proposal of a course plan leading to the Master's degree. This plan must be continually updated for accuracy, substitute courses being noted prior to the end of the semester or summer session during which they are taken.

A course taken outside the Department which is to be counted toward degree requirements must be formally approved in writing by GAC by the end of the semester or summer session during which the course is taken. Such courses can only be used for non-core credit. Students from other departments admitted as "Masters only" cannot use non-CS courses for their Computer Science Degree. Credits transferred from other educational institutions will not count toward the M.S. degree requirements.

II. Thesis or Project (Optional)

Students may choose to write a Master's thesis or project report. A maximum of six hours of CS 790 (Master's thesis) may be used to satisfy the M.S. requirements provided that the thesis is approved by two Computer Sciences faculty members. A maximum of three hours of CS 790 may be used to satisfy the M.S. requirements provided that a Master's project report is filed with the Department. Note that CS 790 can be counted as core credit provided that the advisor agrees to this via an email or letter to the department office. The responsibility for finding a thesis or project advisor is solely that of the student; the Department does not guarantee that an advisor will be provided.

Only a few Master's theses have been written in the Department, but this option is worth considering. There are no rules about the form of a Master's thesis, but it is expected to be a

[6] http://www.wisc.edu/grad/

[7] http://info.gradsch.wisc.edu/admin/academicservices/mdegree.html

substantial piece of work, for example, a comprehensive survey of a particular area. The difference between an M.S. thesis and a Ph.D. thesis is that the former need not contain original research work. An M.S. thesis might well serve as a basis and major first step toward subsequent Ph.D. research. A Master's project can be somewhat less formal than a Master's thesis, describing a project carried out under the supervision of a faculty advisor. Students who write an M.S. thesis should consult the Graduate School publication *A Guide to Preparing Your Master's Thesis*.[8]

[8] http://info.gradsch.wisc.edu/admin/academicservices/mguide.html

Ph.D. Degree Requirements

The requirements for a Ph.D. in Computer Sciences include satisfying the general University residency and minor field requirements, satisfying the Department's breadth requirement, passing the Department's qualifying, preliminary, and final oral examinations, and writing a dissertation. The student will be formally admitted to candidacy for the Ph.D. only when he or she has satisfied requirements I, II, and III below. A student who has satisfied requirements I, II, III, and IV below, and has satisfied the Graduate School's[9] residence requirement, is officially a dissertator in Computer Sciences. Details of the Graduate School's requirements can be found in the *Graduate School Catalog*.

Requirements I and II comprise the Department's qualifying process. The student must pass the qualifying process by the end of the sixth semester. Students who received 8 or more Master's credits for courses taken before being admitted as a Master's degree candidate are allowed only five semesters to complete the qualifying process, and only four semesters if 16 or more credits were received. Those entering with a Master's degree in Computer Science are allowed only four semesters. Students who believe their situation warrants additional time should consult with the GAC chair during the first semester after entry. After this time, the time limit can be changed only by a written appeal to GAC.

Students who do not register and are not on probation may resume their studies up to one year later without re-applying for admission. No additional time will be permitted for passing the qualifying process without written approval from the chair of GAC.

I. Breadth Requirement

To fulfill the Breadth Requirement for the Ph.D. degree, a student must meet the following: Take at least one course from each of the following bands 1, 2 and 3 outside of your depth area. All grades must be at the A/B level or above.

1. Computer Architecture and VLSI: 552, 752, 755, 757, 758
   Operating Systems: 537, 642, 736, 739
   Networking: 640, 707, 740
   Programming Languages and Compilers: 536, 538, 701, 703, 704, 706

2. AI and Vision: 540, 731, 760, 766, 769
   Bioinformatics: 576, 776
   Computer Graphics: 559, 679, 777, 779
   Database Systems: 564, 764, 784

3. Theory of Computing: 520, 577, 787, 810
   Modeling and Analysis of Computer Systems: 547, 737, 747
   Optimization: 525, 635, 719, 720, 726, 730
   Numerical Analysis: 513, 514, 515, 712, 713, 717

   Three of these courses must be at the 700 level, or two must be at the 700 level and two at the 500 level. If you take four courses, they must all be from different areas outside of your depth area.

[9] http://www.wisc.edu/grad/

Only courses taken after receiving a Bachelor's degree may be used to fulfill the breadth requirement. (Note that courses taken at UW-Madison as a Special Student do count, as long as the student has a Bachelor's degree from some college or university.)

II. Qualifying Depth Examination

Students must pass a demanding depth examination in one of the main research areas of the Department. Every semester, the Department offer a four-hour written depth exam in each of the following areas:

- Artificial Intelligence
- Computer Architecture and VLSI
- Computer Graphics
- Database Systems
- Optimization
- Modeling and Analysis of Computer Systems
- Numerical Analysis
- Operating Systems
- Programming Languages and Compilers
- Theory of Computing

A student is allowed at most two chances to pass any area's depth exam. Each exam is graded on a scale of P+ (high pass), P (pass), P- (near pass), or F (fail). A grade of P+ or P is required to pass the Depth Examination Requirement.

A syllabus is published in advance of the exams listing the topics to be covered in each exam. Depth exams are designed to test the preparation of students intending to do Ph.D. research in a given area. These exams cover topics included in courses, as well as additional papers and publications. Copies of previous exams are available on-line.[10]

Qualifying examinations are currently offered early in each regular semester. Students are required to register with the Graduate Coordinator in advance for these exams. Registration deadlines and qualifying exam dates are announced in advance on the Department bulletin board. Registration dates are strictly enforced.

III. Preliminary Examination

To pass this oral examination, the student is expected to display depth of knowledge in the area of specialization in which research for the dissertation will be conducted. Students should plan for the examination, and determine when they are ready to take it, in consultation with the major professor. Students requiring longer than one year beyond the deadline for completing the qualifying

[10] http://www.cs.wisc.edu/includes/archive.html

exam must have a major professor certify that progress is being made toward the exam. In any case, the preliminary exam must be taken within two years after the deadline for completing the qualifying exam.

The preliminary examination committee consists of three members. The composition of the committee will be suggested by the student's major professor in consultation with the student and must be approved by the Chair of the Department. The student should approach each proposed member of the committee, secure agreement to serve, and then discuss a program for preparing for the examination. It is advisable for the student to do this about a semester before the examination is to be scheduled.

Before the preliminary examination, students need to get a *warrant* from the Graduate School. The Department's Graduate Coordinator can provide assistance. Students should be sure to check the Graduate School's calendar to see the cutoff dates by which the exam needs to be completed in order to be eligible for dissertator status (becoming a dissertator has the added benefit of substantially lowering one's tuition).

IV. Minor Field

The Graduate School determines the requirements for fulfilling the minor field requirement for a Ph.D. There are two methods of fulfilling this requirement. The more common option (Option A) is to fulfill the minor field requirement as specified by a department other than Computer Sciences. Students should contact the individual department for details. Computer Sciences graduate students minoring in mathematics should note that special regulations apply to the use, for satisfying minor requirements, of courses cross-listed in both Computer Sciences and Mathematics.

A second option (Option B) is available under which the Computer Sciences Department can arrange special minors for Computer Sciences graduate students who require a program not covered by an orthodox choice of courses in some other single department. For Computer Sciences graduate students, the requirements for an interdepartmental minor are:

1. Approval by the Department of the use of the Option B minor and of the content of the minor course program. This approval must occur no later than half-way through the minor course sequence.

2. Completion of at least 12 graduate credits in two or more departments other than Computer Sciences, in related courses selected for their relevance to a particular area of concentration, with an average grade no lower than B, and with no individual grade lower than BC. One course cross-listed with Computer Sciences may be included in this program if it is staffed by another department and is not applicable to requirements of the student's major program.

Students selecting Option B to satisfy their minor requirement should:

- Consult with the major professor early in the degree program, and decide on a proposed Option B set of courses. Complete a form provided by the departmental office, and have it signed by the major professor and the Department Chair. This constitutes Departmental approval of the proposed courses for satisfying the minor requirement.
- File the approved Minor Agreement Form with the Graduate Coordinator before starting the third minor course.
- After completing the program, bring the preliminary examination warrant to the Graduate Coordinator. The warrant will then be signed by the Department Chair to certify completion of the minor program, and will be returned to the student.

Students who do not yet have a major professor and who want some preliminary advice on the kinds of programs likely to be approved may speak with the minor advisor or a member of GAC.

V. Dissertation

The student must conduct a substantial piece of original research in computer science and report it in a dissertation that meets the highest standards of scholarship.

The initiative in selection of a research supervisor (major professor) is entirely with the student. A professor should be approached for this purpose at as early a stage in the student's graduate work as possible, though usually not until after the student has taken some of the professor's courses or has worked for and demonstrated ability to the professor in some way.

The Graduate School publishes several relevant documents:

A Guide To Preparing Your Doctoral Dissertation[11]

The 3-D's: Deadlines, Defending, Depositing[12]

Doctoral Dissertation and Degree Completion Requirements

Everything You Wanted To Know About Dissertator Status But Were Afraid To Ask

VI. Final Oral Examination

In a final oral examination, the student must explain and defend the contents of the dissertation and exhibit detailed knowledge of the general area in which the reported research falls.

The final oral examination and defense of the dissertation will take place before a committee of five, three of whom will constitute the reading subcommittee. The composition of the committee will be suggested by the student's major professor and approved by the Chair of the Department. At least one of the five members must have a faculty appointment in some department other than Computer Sciences.

As with the preliminary exam, students need to obtain a *warrant* from the Graduate School. The Department's Graduate Coordinator will also help students obtain this document.

[11] http://info.gradsch.wisc.edu/admin/academicservices/pguide.html
[12] http://info.gradsch.wisc.edu/admin/academicservices/ddd.html

Program Planning

I. Starting Out Right and Maintaining Acceptable Progress

Graduate students who have half-time jobs (or less) are generally advised to take three courses per regular semester. However, it is permissible to take four. Beginning graduate students should select 400-, 500- or 700-level courses depending on their undergraduate background. Students with little or no programming background are encouraged to acquire programming skills prior to beginning their regular graduate studies. Most graduate courses offered by the Department require at least a minimal level of programming skills. Students who are accepted for admission, but who lack basic computer science prerequisites (CS 354 and 367), should contact the Graduate Advising Committee (GAC) well in advance of beginning their first semester.

All entering students should consult with GAC no later than the beginning of registration week to discuss the planning of their schedules. GAC must formally approve all graduate schedules each semester. Students who have completed the Master's degree requirements or the qualifying process and who have an official doctoral major professor may have their schedules approved by that professor.

It is important that the student maintain satisfactory academic progress (SAP). Unsatisfactory academic progress (USAP) leads to ineligibility for financial support through teaching assistantships and, if continued for a second semester, results in dismissal from the Department. Initially, failure to complete an adequate course load is the only reason for USAP. Students who are doing badly enough that USAP is possible should immediately discuss the problem with a member of GAC.

II. Sources of Advice

- Members of GAC are available during regular schedule office hours, with special hours during registration periods. They will give advice and must also formally approve proposed courses of study.
- An informal advising service is run by graduate students during registration periods. This is an excellent source of information about the pros and cons of courses and professors.
- Particular questions about a course can often best be resolved by contacting the course instructor.
- The University provides counseling services for help with personal problems. See the handbook *The Wheat and the Chaff.*

III. The Normal Schedule for a Computer Sciences Graduate Student

Master's degree

Most full-time students obtain this degree in three or four semesters.

Breadth-courses requirement

Upon arrival, students should plan a course of study that will satisfy their breadth-courses requirement during the period allotted. It is important to check the prerequisites of 700-level classes, as well as the schedule of course offerings. Many 700-level classes are offered only once a year, and some are only offered every third or fourth semester.

Doctoral qualifying examination

Students generally take their qualifying examination during their second or third year as a graduate student, but should begin preparing for their qualifying examination well in advance. Qualifying examinations are difficult; failure is quite possible without thorough preparation. Topic and reading lists for the various areas are available from the Graduate Coordinator. Also, students may find the on-line archive of previous examinations[13] useful in determining what will be expected of them. In general, the qualifying examination (a) requires a broad and unified knowledge of an area, (b) is closed-book, (c) is written under time constraints, and (d) often contains questions that require essay answers. It is a good idea for a student to discuss preparation for the qualifying examination with appropriate faculty members once the area of specialization has been decided.

Doctoral preliminary examination

This should be taken within two semesters after the qualifying examination deadline has been passed. Before taking it, the student should have satisfied the minor requirement and should have worked on some project for his or her major professor.

Oral defense of Ph.D. dissertation

Students may reach this final step as early as four years (eight semesters) after entry or as late as the tenth or eleventh semester of graduate study.

Students who do not maintain a schedule reasonably close to these recommendations may not receive continued financial support from the Department.

IV. Scheduling Courses in Particular Areas

Artificial Intelligence

The Department's standard course offerings in Artificial Intelligence include a senior-level survey of the field, CS 540; a graduate-level survey of the field, CS 731; and a series of 700-level courses each covering one sub-area of the field. The sub-areas and the primary courses covering them are Machine Learning (CS 760), Computer Vision (CS 766), and Natural Language Processing (CS 769). Each of the 700-level courses has CS 540 (or 731) as a prerequisite. Staff shortages may make it impossible for the Department to regularly offer CS 731. When it is offered in a semester appropriate for their course scheduling, graduate students should take CS 731 instead of 540. Students will be prepared for the Ph.D. qualifying exam in AI if they have mastered the material presented in CS 540 (or 731) and in two of CS 760, 766, and 769.

The AI qualifying exam is designed to test for both a general "breadth" knowledge of AI, plus a deeper specialized knowledge of one particular sub-area within AI. *Students are required to specify their sub-area when they sign up for the exam, and then answer corresponding questions in that sub-area.* Specifically, the exam will consist of questions in nine separate sections. There will be two "general" AI questions (in section G540/731), two "basic" questions in each of the three sub-areas (in sections B760, B766, and B769), and two "advanced" questions in each of the three sub-areas (in sections A760, A766, and A769). To pass the examination,

[13] http://www.cs.wisc.edu/includes/archive.html

students will have to answer satisfactorily one of the two questions in section G540/731; both of the questions in one of the A sections; both of the questions in the B section for the same sub-area; and two additional questions from any of the other B sections. The two additional questions can be but do not have to be from the same section. The A section chosen must correspond to the sub-area for which the student is signed up.

Computer Architecture and VLSI

There are three main courses in computer architecture, one in VLSI design, and one in design tools for VLSI. All are cross-listed as Electrical and Computer Engineering courses with the same number. CS 552 is the basic course. CS 752 and 757 cover advanced topics in computer architecture, CS 755 covers advance topics in VLSI design, and CS 756 covers design tools for VLSI. CS 752 is usually offered by the Computer Sciences Department in the fall and by the ECE Department in the spring. CS 755 and 757 are usually offered in the fall under the auspices of the ECE Department and in the spring under the Computer Sciences Department. Though the same basic course, the emphasis in CS 755 varies considerably, and CS graduate students who do not have an undergraduate engineering degree usually experience difficulty with the course as offered by ECE. The Ph.D. qualifying exam requires knowledge of material covered in CS 552 and 752 and some extra readings, most of which are covered in CS 757.

Computer Graphics

Students interested in graphics research should begin with CS 559, the primary advanced undergraduate graphics course. Students who already have an undergraduate graphics background should consult with faculty. Students may then take graduate courses in sub-areas of graphics: CS 777 covering animation, and CS 779, which discusses rendering. Students should also consider CS 679, which covers interactive virtual environments in the context of game development. All of these courses assume knowledge of linear algebra and calculus, preferably multi-variate for CS 779. CS 559 is taught at least once a year, while the other courses may be taught less frequently. Other courses of relevance to graphics include CS 533 (image processing), CS 540 (artificial intelligence), CS 558 (computational geometry) and CS 766 (computer vision). Several courses taught in the Department of Mechanical Engineering are also related to graphics research. Material for the qualifying exam in graphics is covered in CS 559, CS 777 and CS 779, although additional reading is also required.

Database Systems

At the present time three courses for graduate students are offered in the area of database systems (CS 364, which provides a general introduction to database systems, is for undergraduates only). CS 564 provides a systems-oriented introduction to databases and their implementation. Students with an interest in more advanced topics and current research directions in the area, such as students preparing for the database Ph.D. qualifying exam, should take CS 764 and 784 after completing CS 564. Those students who are planning on taking the database Ph.D. qualifying exam are expected to complete their preparation for the exam independently. In addition, students interested in pursuing research in the database systems area are advised to consider taking courses in the related areas of operating systems, performance evaluation, and artificial intelligence to prepare for their work in this relatively broad area.

Optimization

Students interested in optimization should first take CS 525 and CS 635. After completing a good linear programming and optimization modeling course, students may take CS 719 or CS 720 (which emphasize combinatorial optimization) without further prerequisites, and students with a background in basic mathematical analysis may take CS 726, CS 727, or CS 730 (which emphasize continuous, nonlinear optimization). (Note that CS 525 and CS 635 are not prerequisites for CS 726, 727 or 730.) All of the 700-level courses may be taken independently of each other. Ph.D. students planning on taking their qualifying examination in this area should be familiar with the material in CS 525 and CS 635 and at least three courses from 719, 720, 726, 727, 730. Master's and Ph.D. students minoring in optimization should take CS 525 or 635 and at least one of the 700-level courses in the area.

Modeling and Analysis of Computer Systems

The various techniques for performance modeling and analysis of computer systems constitute the topics in the MA area. Three courses, CS 547, 737, and 747, are currently offered in this area. CS 547, which introduces analytic modeling methods and some elementary queuing theory results, is offered in the Fall semester each year. Students enrolling in CS 547 are expected to have some background in operating systems, databases, or computer architecture, which provide some of the systems analysis questions that are used as examples in the course. Students with little or no background in probability theory should audit or enroll in Math 431 prior to or concurrently with CS 547. Both CS 737 and 747 are usually offered in the spring semester each year. CS 747 covers additional and more advanced analytical modeling methods, while CS 737 covers performance-simulation methods. CS 547 is a pre-requisite for CS 737 and 747. CS 547 and 747, plus additional reading on the reading list, are required for the Ph.D. qualifying exam in MA.

Numerical Analysis

The two-course sequence CS 513-514 provides the basic introduction to numerical analysis. Students planning to take the Ph.D. qualifying exam in numerical analysis should also take the course sequence: CS 717, 712 and 713. This sequence also serves as the starting point for the more advanced courses, CS 883, 884, 887, 718, 881, 882, and 885. Numerical analysis is a broad field and Ph.D. students usually take several courses in mathematics as well as independent reading courses. Prospective Ph.D. students should consult with numerical analysis faculty about their course of study.

Operating Systems

Three courses are offered in the area of operating systems. CS 537 introduces the subject, providing hands-on experience with building parts of operating systems in simple environments; there are several large programming assignments. CS 736 is an advanced course, often run as a seminar, which discusses a selection of advanced topics. This course often has a large project associated with it. CS 739 takes up where CS 736 leaves off. It covers distributed systems in greater depth, studying a wide variety of systems and examining issues such as replication, fault tolerance, load balancing, and security. The Ph.D. qualifying exam covers material in CS 537 and 736 and advanced material from CS 739.

Programming Languages and Compilers

Students interested in the area of programming languages should first take CS 536 and 538. The former provides an introduction to compilers and to programming-language implementation techniques. The latter provides an introduction to the theory and design of programming languages. The graduate courses in the area are CS 701, 703, and 704. CS 701 is the graduate compiler course. It covers program analysis, optimization, and code generation. CS 703 covers a selection of advanced topics. CS 704 covers graduate-level topics in the theory of programming languages, including the study of functional languages and formal language semantics. Occasionally, a section of CS 838 deals with issues in programming language design; this course is extremely valuable for students intending to do Ph.D. work in programming languages. For the Ph.D. qualifying exam, CS 536, 538, 701, and 704 are required and CS 703 is recommended.

Theory of Computing

The basic concepts in the theory of computation are introduced in the undergraduate theory courses, CS 520 and 577. Students preparing for the Ph.D. exam in theory should take CS 810 and 787, and should study topics from at least two courses in the following list: CS 709, 767, 812, 820, 830. CS 709, 767, 787 and 812 focus on algorithms for specific problems, and CS 810, 820 and 830 study models of computation and complexity classes; students are advised to take both types of courses in their program. All students interested in theory are encouraged to take CS 880, a special-topics course, when offered.

V. Transfer of Credits Toward the Master's Degree

Other UW Departments and UW Special Students

Computer Sciences students who were previously special students or students in other departments may apply some or all of their previous courses at the University toward a Computer Sciences Master's degree, subject to the following restrictions:

1. A course must either be a Computer Sciences course or it must satisfy the criteria for "approved" courses (see part VII below). In either case, the course must be entered on the student's GAC file, and be initialed as approved by a GAC member.
2. At most, one-fourth of the courses applied toward one graduate degree may be applied toward a second graduate degree. Courses applied toward an undergraduate degree may not be applied toward a graduate degree.

Other Universities

Courses taken at other universities may not be counted toward the Master's degree.

VI. Maintenance of Student Records

GAC maintains two types of records about each student. These records are stored in the Graduate Coordinator's offices, and are available for inspection and verification by the student. It is the responsibility of the student to see that these records are correct and up-to-date.

The Master's Degree/Qualifying Examination Course Plan

The course plan must be prepared, in consultation with GAC and the student's personal faculty adviser, during registration week of the student's first semester as a graduate student. The purpose of the course plan is to ensure that the student understands Department regulations and is pursuing a suitable course of study. The course plan may be changed at any time in consultation with GAC and the student's personal faculty adviser. For most students the course plan represents a course of study leading to a Master's degree. For graduate students not pursuing a Master's degree, the course plan represents a course of study preparing for the doctoral qualifying examination.

The Course-History Record

The course history record card contains a record of all courses completed or being taken. This record must be brought up to date during each registration week.

In addition to the above records, a student wishing to obtain a Master's degree should submit a completed application form, available from the Department office, not less than one month before the end of the semester when the degree is desired.

VII. Approved Courses

Courses taken outside the Department that are to be counted, either toward satisfactory academic progress or toward a Master's degree, must be approved by GAC. Approval must be obtained before the course has been taken. Usually approval is given during the registration period, but approval can be requested at an earlier stage if a student wishes to plan ahead.

To be approved, a course must materially contribute to the specific computer science education for which the student is working. In doubtful cases, GAC will require a note from a Department faculty member supporting the request for approval. Only CS courses are approved for students from other departments enrolled in the "Masters only" degree program.

Satisfactory Academic Progress

Each graduate student is responsible for planning and carrying out a program of study that continually meets with the approval of the Department. Students should meet with a member of the Graduate Advising Committee (GAC) every semester, usually during the registration period, to get approval for the plan for the following semester. The record of approvals and future plans is maintained in a file that is available at any time for the student to inspect. Any exceptions to rules that have been granted to the student are also recorded in this file. A student who wants to drop or add a course during the semester should get the change approved by a member of GAC.

The rules enforced by the Department for satisfactory progress are distinct from the rules used by the Graduate School. The student must satisfy both sets of rules. Here we concentrate on the rules of the Department.

I. Definitions

Full-load and Part-load

Part-load students do not have to take as many credits during a semester as full-load students. (The specific rule is discussed below.) All students are considered to be full-load students unless they have been granted part-load status by GAC. Part-load status is granted semester by semester to students who have full-time jobs, non-academic duties, or substantial family responsibilities. Such students should apply in writing to GAC at the beginning of each semester for which they want part-load status. They will be notified in writing whether their request has been approved.

Full-load status is distinct from full-time status as determined by the Graduate School for residence credit. Because of Graduate School rules, students often need to take 8 instead of 6 credits during a semester. The combined effect of the various rules is this:

> Part-load dissertators take 1 credit. Part-load students and dissertators take 3 credits. Full-load students who are TAs, PAs, or unsupported by the Department take at least 6 credits. Research assistants and fellows take at least 8 credits.

It is essential for foreign students and students receiving V.A. benefits to be full-load students. Foreign students should check with the Graduate School and the Office for Foreign Students to ensure that they are satisfying residence, visa, and other requirements.

Regular Semester

In the following requirements, *regular semester* denotes either fall or spring semester of an academic year; it does not include summer session.

Approved Courses

Approved courses are courses that have been approved by GAC as appropriate for a student's studies. For students who have neither obtained a Master's degree in Computer Science nor passed the qualifying process, GAC usually approves only courses leading toward the Master's degree.

Courses are approved only if they fall into one of the following categories:

1. Courses from other departments that GAC considers to be an important part of the student's program. These courses will usually be numbered 400 or above. To comply with current Graduate School requirements, except as noted below in point 2, no course numbered less than 300 will be approved.
2. CS 302, 352, 354, and 367, and Math 221, 222, and 223, provided that the student has been admitted with deficiencies that are being removed by taking these courses.
3. All University of Wisconsin-Madison CS courses numbered 400 and above (and courses cross-listed with such CS courses) taken by a student and not applied to any other degree.

Satisfactory Completion of Courses

Courses taken for credit and passed with letter grades A, AB, B, BC, C, S, and P are satisfactorily completed. These criteria determine satisfactory completion for the Department. The Graduate School has its own rules, and they should be consulted if any question arises.

Examinations

The terms *qualifying examination process, preliminary examination* and *final oral examination* designate the procedures and/or examinations of those names supervised by the Department of Computer Sciences. The requirements for the Ph.D. include time limits for their completion.

II. Satisfactory Academic Progress Criteria

A graduate student in Computer Sciences shall be considered to have made satisfactory academic progress (SAP) at the end of any regular semester only if, at the end of the semester, the following conditions are all satisfied:

1. (a) Before achieving dissertator status: the student has satisfactorily completed at least six (if full load) or three (if part load) credits of approved courses during the semester.

   (b) After achieving dissertator status: the student has satisfactorily completed at least three credits of courses approved by the student's major professor.

2. The student has removed all incomplete grades from any previous regular semester or summer session.

3. The student has passed any required exams and procedures within designated time limits.

III. Dismissal

Any graduate student who fails to make SAP during two consecutive regular semesters (fall and spring, or spring and fall) will be dismissed from the Department at the end of the subsequent summer session. Any student who fails to make SAP because of criterion II.3 above will be dismissed from the Department at the end of the subsequent summer session.

IV. Financial Support

To be eligible for financial support from sources controlled by the Computer Sciences Department a student must be making SAP in the Department.

V. Appeal Procedure

Any graduate student may appeal any aspect of the SAP rules, provided that the appeal is made in a timely way. In particular, appealing a decision that a student did not make SAP must be initiated not later than the end of the fourth week of the subsequent regular semester.

To appeal, the student should write a letter to the Chair of GAC stating the basis for the appeal. This letter should explain clearly the reasons for the appeal, and should be accompanied by appropriate documents such as a medical certificate if the appeal is on the grounds of ill health or such as a supporting letter from a Computer Sciences faculty member if the appeal concerns an unusual combination of courses. It will often be useful for the student to discuss the problem with a member of GAC or with the student's personal faculty advisor before putting the appeal into writing.

GAC will consider every such written appeal and will notify the student of its decision at the earliest opportunity, normally within four working weeks. A student who is not satisfied with the decision by GAC may submit a further appeal in writing to the Chair of the Department. The Chair will place the appeal on the agenda of a regular faculty meeting, will circulate the letter of appeal and accompanying documentation, and will give the student written notification of the meeting. The meeting will be scheduled at the earliest opportunity, normally within four working weeks after receipt of the letter to the Chair. The student and any of the student's advisors may attend the meeting to present the appeal, provided that the Chair of the Department is advised in writing before the start of the meeting. In accordance with Wisconsin law, the meeting will begin in open session, but the Chair will move that the meeting convene in closed session before the appeal is considered.

Computer Sciences Courses

General Information about Courses

Courses numbered 399 and below may be taken for undergraduate credit only. Courses numbered 400 through 699 may be taken by either undergraduate or graduate students. Courses numbered 700 or above are intended only for graduate students. Undergraduates are allowed to take courses numbered 700 or above, but only if permission is obtained from the dean's office.

Courses offered less than once every two years are marked as "infrequently offered;" students should not count on taking these classes when planning their schedules. Tentative timetables[14] for upcoming semesters are available.

World-Wide Web pages for the current semester's offerings[15] of many Computer Sciences courses are available. Additional information about many cross-listed courses can be found via the College of Engineering's[16] and the Department of Mathematics'[17] WWW home pages.

240 Introduction to Discrete Mathematics — 3 cr. (also Math)

Basic concepts of logic, sets, partial order and other relations, and functions. Fundamental principles of counting. Basic algebraic structures: modulo arithmetic, group, ring, and field structures, Boolean algebra. Introduction to graph theory: trees, depth first search, matching, max-flow min-cut, and other optimization algorithms. Applications. Prereq: Math 221.

250 Digital Society: The Impact of Computers and Computer Technology — 3 cr.

Introduction to computers in the digital society; social changes they influence, and choices they present. Topics include: digital divide, role of computers in improving quality of life, electronic voting and governance, digital intellectual property rights, privacy, computers and the environment.

252 Introduction to Computer Engineering — 2 cr. (also ECE)

Logic components built with transistors, rudimentary Boolean algebra, basic combinational logic design, basic synchronous sequential logic design, basic computer organization and design, introductory machine-and assembly-language programming.

302 Introduction to Programming — 3 cr.

Instruction and experience in the use of an object-oriented programming language. Program design; development of good programming style; preparation for other Computer Science courses. Prereq: Problem solving skills such as those acquired in a statistics, logic, or advanced high school algebra course; or consent of instructor. Open to Fr.

---

[14] http://www.cs.wisc.edu/tentative_future_timetables.html
[15] http://timetable.doit.wisc.edu/cgi-bin/TTW3.navigate.cgi?20072+crses/d266.html
[16] http://www.engr.wisc.edu
[17] http://math.wisc.edu/courses/

304 WES-CS Group Meeting

Small group meetings for Wisconsin Emerging Scholars—Computer Science (WES-CS) students. Meet for two hours each week in small groups to work together on problems related to the CS 302 course material. Co-req: CS 302 and WES-CS membership. Open to Fr. Prereq: No prerequisites. Co-requisites include enrollment in CS 302 and membership in the WES-CS (WSCS) student group.

310 Problem Solving using Computers 3 cr.

Gives engineering students an introduction to computer and analytical skills to use in their subsequent course work and professional development. Discusses several methods of using computers to solve problems, including elementary Fortran and C programming techniques, the use of spreadsheets, symbolic manipulation languages, and software packages. Techniques will be illustrated using sample problems drawn from elementary engineering. Emphasis on introduction of algorithms with the use of specific tools to illustrate the methods. Prereq: Math 222.

352 Digital Systems Fundamentals 3 cr. (also ECE)

Logic components, Boolean algebra, combinational logic analysis and synthesis, synchronous and asynchronous sequential logic analysis and design, digital subsystems, computer organization and design. Prereq: CS 252 or equivalent. Not open to students with EGR classification.

354 Machine Organization and Basic Systems 3 cr.

An introduction to current system structures of control, communication, memories, processors and I-O devices. Projects involve detailed study and use of a specific small computer hardware and software system. Prereq: CS 302 and ECE/CS 252 or consent of instructor. Open to Fr.

367 Introduction to Data Structures 3 cr.

Study of data structures (including stacks, queues, trees, graphs, and hash tables) and their applications. Development, implementation, and analysis of efficient data structures and algorithms (including sorting and searching). Experience in use of an object-oriented programming language. Prereq: CS 302 or consent of instructor. Students are strongly encouraged to take CS 367 within two semesters of having taken CS 302.

368 Learning a New Programming Language 1 cr.

This course is for students who are familiar with `Java` programming language and are interested in learning `C++`. Prereq: CS 302 or consent of instructor.

369 Web Programming 3 cr.

Covers web application development end-to-end: languages and frameworks for client- and server-side programming, database access, and other topics. Involves hands-on programming assignments. Students attain a thorough understanding of and experience with writing web applications using tools popular in industry. Prereq: CS 367 or substantial programming experience and consent of instructor.

371 Technology of Computer-Based Business Systems 3 cr. (also Info Sys)

Overview of computers, their attendant technology, and the implications of this technology for large-scale, computer-based information systems. Topics include hardware, system software, program development, files, and data communications. Prereq: Bus 370 and CS 302, or equivalent experience with consent of instructor.

412 Introduction to Numerical Methods 3 cr.

Interpolation, solution of linear and nonlinear systems of equations, approximate integration and differentiation, numerical solution of ordinary differential equations. Prereq: Math 222 and either CS 240 or Math 234, and CS 302, or equivalent, and knowledge of matrix algebra.

416 Foundations of Scientific Computing 3 cr.

Basic techniques for scientific computing, including fundamentals of linear algebra and numerical linear algebra, rootfinding, floating-point arithmetic, interpolations and splines, linear and quadratic programming. Prereq: Math 222 and either CS 240 or Math 234, and CS 302, or equivalent.

425 Introduction to Combinatorial Optimization 3 cr. (also Math & IE)

Exact and heuristic methods for key combinatorial optimization problems such as: shortest path, maximum flow problems, and the traveling salesman problem. Techniques include problem-specific methods and general approaches such as branch-and-bound, genetic algorithms, simulated annealing, and neural networks. Prereq: Math 221 or CS 302 or consent of instructor.

435 Introduction to Cryptography 3 cr. (also Math & ECE)

Cryptography is the art and science of transmitting digital information in a secure manner. This course will provide an introduction to its technical aspects. Prereq: Math 320 or 340 or consent of instructor.

460 Artificial Intelligence Programming Languages and Tools 3 cr.

Symbolic computation; Lisp programming; Prolog programming; knowledge representation languages based on logic, objects, frames, rules; symbolic pattern matching; automatic inferencing and reasoning techniques; special-purpose languages and computer architectures for artificial intelligence applications. Prereq: CS 367. (Last taught: Spring 92)

475 Introduction to Combinatorics 3 cr. (also Math & Stat)

Problems of enumeration, distribution and arrangement. Inclusion-exclusion principle. Generating functions and linear recurrence relations. Combinatorial identities. Graph coloring problems. Finite designs. Systems of distinct representatives and matching problems in graphs. Potential applications in the social, biological, and physical sciences. Puzzles. Emphasis on problem solving. Prereq: Math 320 or 340 and consent of instructor.

513 Numerical Linear Algebra 3 cr.

Direct and iterative solution of linear and nonlinear systems and of eigenproblems. LU and symmetric LU factorization. Complexity, stability, and conditioning. Nonlinear systems. Iterative methods for linear systems. QR-factorization and least squares. Eigenproblems: local and global methods. Prereq: Math 340 or equivalent; CS 302 or equivalent.

514 Numerical Analysis 3 cr.

Polynomial forms, divided differences. Polynomial interpolation. Polynomial approximation: uniform approximation and Chebyshev polynomials, least-squares approximation and orthogonal polynomials. Splines, B-splines and spline approximation. Numerical differentiation and integration. Numerical methods for solving initial and boundary value problems for ordinary differential equations. Prereq: Math 340 or equivalent; CS 302 or equivalent.

515 Introduction to Splines and Wavelets 3 cr. (also Math)

Introduction to Fourier series and Fourier transform; time-frequency localization; wavelets and frames; applications: denoising and compression of signals and images. Interpolation and approximation by splines: interpolation, least-squares approximation, smoothing, knot insertion and subdivision; splines in CAGD. Prereq: Math 340 or equivalent; CS 302 or equivalent.

520 Introduction to Theoretical Computer Science 3 cr.

Survey of the basic concepts of theory, including context-free and context-sensitive languages, regular sets, finite and pushdown automata, Turing machines, undecidable problems, complexity with respect to time and space, NP-completeness, and reducibilities. Prereq: CS 367, Math 222, and CS 240, or consent of instructor.

525 Linear Programming Methods 3 cr.

Real linear algebra over polyhedral cones, theorems of the alternative for matrices. Formulation of linear programs. Duality theory and solvability. The simplex method and related methods for efficient computer solution. Perturbation and sensitivity analysis. Applications and extensions, such as game theory, linear economic models and quadratic programming. Prereq: Math 443 or 320 or 340 or consent of instructor.

526 Advanced Linear Programming 4 cr. ugrad, 3 cr. grad (also IE)

Review of linear programming. Polynomial time methods for linear programming. Quadratic programs and linear complementarity problems and related solution techniques. Solution sets and their continuity properties. Error bounds for linear inequalities and programs. Parallel algorithms for linear and quadratic programs. Prereq: CS 525 or equivalent, CS 302 or equivalent, or consent of instructor. (Last taught: Spring 92)

532 Theory and Applications of Pattern Recognition 3 cr. (also ECE & ME)

Pattern recognition systems and components; decision theories and classification; discriminant functions; supervised and unsupervised training; clustering; feature extraction and dimensional reduction; sequential and hierarchical classification; applications of training, feature extraction, and decision rules to engineering problems. Prereq: ECE 331 or Math 431 or consent of instructor.

533 Image Processing 3 cr. (also ECE)

Mathematical representation of continuous and digital images; models of image degradation; picture enhancement, restoration, segmentation, and coding; pattern recognition, tomography. Prereq: ECE 330 or consent of instructor; Math 320 or 340 or equiv. recommended.

536 Introduction to Programming Languages and Compilers 3 cr.

Introduction to the theory and practice of compiler design. Comparison of features of several programming languages and their implications for implementation techniques. Several programming projects required. Prereq: CS 367 and either CS 354 or 552.

537 Introduction to Operating Systems 4 cr. ugrad, 3 cr. grad.

Input-output hardware, interrupt handling, properties of magnetic tapes, discs and drums, associative memories and virtual address translation techniques. Batch processing, time sharing and real-time systems, scheduling resource allocation, modular software systems, performance measurement and system evaluation. Prereq: CS 354 and CS 367.

538 Introduction to the Theory and Design of Programming Languages 3 cr.

Design and theory of programming languages: procedural, object-oriented, functional and logic paradigms. Serial and concurrent programming. Execution models and formal specification techniques. Prereq: CS 354 and CS 367.

539 Introduction to Artificial Neural Network and Fuzzy Systems 3 cr. (also ECE, ME)

Theory and applications of artificial neural networks and fuzzy logic: multi-layer perceptrons, self-organizing maps, radial basis networks, Hopfield networks, recurrent networks, fuzzy-set theory, fuzzy logic control, adaptive fuzzy neural networks, genetic algorithms, and evolutionary computing. Applications to control, pattern recognition, nonlinear system modeling, speech and image processing. Prereq: CS 302, or CS 310, or knowledge of C.

540 Introduction to Artificial Intelligence 3 cr.

Principles of knowledge-based search techniques; automatic deduction, knowledge representation using predicate logic, machine learning, probabilistic reasoning. Applications in tasks such as problem solving, data mining, game playing, natural language understanding, computer vision, speech recognition, and robotics. Prereq: CS 367.

545 Natural Language and the Computer 3 cr.

The course covers basic techniques and tools in natural language processing: generative grammars, parsing, dictionary construction, semantic networks, generation of text from a knowledge base, natural language interfaces, and machine translation. Prereq: CS 536 or CS 537 or 564 or consent of instructor.

547 Computer Systems Modeling Fundamentals 3 cr.

An introduction to basic tools and applications for modeling and analysis of computer systems. Fundamentals of network flow graphs, graph models of computation and stochastic models of computer system performance. Network delay analysis and capacity planning, reachability analysis for deadlock detection in distributed systems, Markov chains, elementary queueing theory, basic concepts of queueing network models and associated analyses. Prereq: Math 223, CS 367 and CS 354.

550 Computers and Society 3 cr. (also Social Studies)

The effect of scientific and technological change on social and economic organization. Historical examples. Comparison, with these examples, of the computer and its effect. Consideration of possible uses of computer systems, social change which they would influence, and the choices they present. Prereq: Junior standing. (Last taught: Fall 90)

552 Introduction to Computer Architecture 3 cr.

The design of computer systems and components. Processor design, instruction set design, and addressing; control structures and microprogramming; memory management, caches, and memory hierarchies; interrupts and I/O structures. Prereq: ECE/CS 352 and CS/ECE 354; co-req: CS 367.

558 Introduction to Computational Geometry 3 cr.

Introduction to fundamental geometric computations and algorithms, and their use for solving engineering and scientific problems. Computer representations of simple geometric objects and paradigms for algorithm design. Applications from areas of engineering analysis, design and manufacturing, biology, statistics, and other sciences. Prereq: CS 367 or equivalent, Math 223 or equivalent, or consent of instructor.

559 Computer Graphics 3 cr.

Survey of computer graphics. Image representation, formation, presentation, composition and manipulation. Modeling, transformation, and display of geometric objects in 2 and 3 dimensions. Representation of curves and surfaces. Rendering, animation, multi-media and visualization. Prereq: Math 320 or 340 (linear algebra), and CS 367.

564 Database Management Systems: Design and Implementation **4 cr. ugrad, 3 cr. grad.**

What a database management system is; different data models currently used to structure the logical view of the database: relational, hierarchical, and network. Hands-on experience with relational and network-based database systems. Implementation techniques for database systems. File organization, query processing, concurrency control, rollback and recovery, integrity and consistency, and view implementation. Prereq: CS 367 and 354.

576 Introduction to Bioinformatics **3 cr (also BMI).**

Algorithms for computational problems in molecular biology. The course will study algorithms for problems such as: genome sequencing and mapping, pairwise and multiple sequence alignment, modeling sequence classes and features, phylogenetic tree construction, and gene-expression data analysis. Prereq: CS 367 and Math 222.

577 Introduction to Algorithms **3 cr.**

Survey of important and useful algorithms for sorting, searching, pattern-matching, graph manipulation, geometry, and cryptography. Paradigms for algorithm design. Techniques for efficient implementation. Prereq: CS 367, Math 222, and CS 240, or consent of instructor.

635 Tools and Environments for Optimization **3 cr.**

Formulation and modeling of applications from computer sciences, operations research, business, science and engineering involving optimization and equilibrium models. Survey and appropriate usage of software tools for solving such problems, including modeling language use, automatic differentiation, subroutine libraries and web-based optimization tools and environments. Prereq: CS 302, Math 340 or equivalent.

638 Undergraduate Topics in Computing **3 cr.**

Prereq: Consent of instructor.

640 Introduction to Computer Networks **3 cr.**

Architecture of computer networks and network protocols, protocol layering, reliable transmission, congestion control, flow control, naming and addressing, unicast and multicast routing, network security, network performance, widely used protocols such as Ethernet, wireless LANs, IP, and HTTP. Prereq: CS 537.

642 Introduction to Information Security **3 cr.**

This is a senior level undergraduate course covering various topics on information security. the course will cover a wide range of topics, such as, cryptographic primitives, security protocols, system security, and emerging topics. Prereq: CS 537 or consent of instructor. Elementary knowledge of mathematical logic and discrete probability theory is also required.

679 Computer Game Technology 3 cr.

Survey of software technology important to computer games and other forms of interactive technology: Real-time image generation, managing complex geometric models, creating virtual characters, simulating physical phenomenon, networking technology for distributed virtual environments. Prereq: CS 559.

681–682 Senior Honors Thesis 3 cr. per sem.

Prereq: Honors candidacy and consent of instructor.

691–692 Senior Thesis 2–3 cr. per sem.

(A year's course must be taken to get credit.) Prereq: Consent of instructor.

699 Directed Study 1–6 cr.

Prereq: Junior or senior standing and consent of instructor.

701 Programming Languages and Compilers 3 cr.

Design and implementation of compilers for modern programming languages. Emphasis on tools for compiler construction. Prereq: CS 536.

703 Advanced Topics in Programming Languages and Compilers 3 cr.

Advanced topics in compiling and programming languages design. Advanced parsing techniques; automatic syntactic error correction; local and global code optimization; attribute grammars; programming language design issues (data and control abstractions, specification and verification of high level languages). Prereq: CS 701.

704 Principles of Programming Languages 3 cr.

Introduction to principles of advanced programming languages and programming-language theory. Topics include: lambda-calculus, functional languages, polymorphic functions, type inference, structural induction, lazy evaluation, operational semantics, denotational semantics, and axiomatic semantics. Prereq: CS 536 or consent of instructor.

706 Analysis of Software Artifacts 3 cr.

Advanced course covering various analysis techniques used in software engineering. This course will cover techniques for analyzing various software artifacts. Some of the topics that will be covered are: model checking, testing, program analysis, requirements analysis, and safety analysis. Prereq: CS 536 or consent of instructor. A basic knowledge of mathematical logic is also required.

707 Mobile and Wireless Networking — 3 cr. (also ECE)

Design and implementation of protocols, systems, and applications for mobile and wireless networking, particularly at the media access control, network, transport, and application layers. Focus is on the unique problems and challenges presented by the properties of wireless transmission, various device constraints such as limited battery power, and node mobility. Prereq: CS 640 or CS 537 or equivalent, or permission of the instructor.

709 Mathematical Techniques for Analysis of Algorithms — 3 cr.

Techniques for quantitative analysis of algorithms. Charging arguments, amortization, probabilistic methods. Adversary and information lower bounds. Use of methods from combinatorics, complex analysis, and asymptotics in obtaining precise analyses of quicksort, chained hashing, and other algorithms. Prereq: CS 577, knowledge of complex variables at the level of Math 321. (Last taught: Spring 96)

714 Methods of Computational Mathematics I — 3 cr. (also Math)

Development of finite difference methods for hyperbolic, parabolic, and elliptic partial differential equations. Analysis of accuracy and stability of difference schemes. Direct and iterative methods for solving linear systems. Introduction to finite volume methods. Applications from science and engineering. Prereq: CS 302, CS 412, Math 322, 340, 521 or equivalent, or consent of instructor.

715 Methods of Computational Mathematics II — 3 cr. (also Math)

Introduction to spectral methods (Fourier; Chebyshev; Fast Fourier Transform), finite element methods (Galerkin methods; energy estimates and error analysis), and mesh-free methods (Monte Carlo; smoothed-particle hydrodynamics) for solving partial differential equations. Applications from science and engineering. Prereq: CS 302, CS 412, Math 322, 340, 521 or equivalent, or consent of instructor.

717 Numerical Functional Analysis — 3 cr.

Fundamentals of normed spaces and linear operators; analysis of nonlinear operators; existence of, and iterative methods for, solutions of linear and nonlinear operator equations, error estimation; variational theory and minimization problems; monotonicity theory. Development of abstract tools and application of them to the general analysis of numerical methods for such problems as differential and integral equations. Prereq: CS 513, CS 514 and Math 234 or consent of instructor.

719 Network Flows — 3 cr.

Optimization problems and techniques for networks, including single and multi-commodity network flow, critical path, and facilities location problems. The theory of totally unimodular matrices and its relationship to network optimization. Prereq: CS 525 or consent of instructor.

720 Integer Programming — 3 cr. (also IE)

Formulation of integer programming problems and the characterization of optimization problems representable as integer and mixed-integer programs. The degree of difficulty of classes of integer programs and its relation to the structure of their feasible sets. Optimality conditions. Branch-and-bound, cutting plane, and decomposition methods for obtaining solutions or approximating solutions. Prereq: CS 525 or consent of instructor.

723 Dynamic Programming and Associated Topics — 3 cr. (also IE)

A generalized optimization model; discrete and continuous state spaces; deterministic and stochastic transition functions. Multistage decision processes. Functional equations and successive approximation in function and policy spaces. Relationship to linear programming and acyclic networks. Markovian decision processes. Solution methods and computational problems. Associated topics and applications such as calculus of variations; feedback control processes; and optimal trajectories, inventory and maintenance policies, and stopping rules. Prereq: CS 525 or IE 623; Math 521 or CS 726; Math 431 and computer programming, or consent of instructor. (Last taught: Spring 83)

726 Nonlinear Programming Theory and Applications — 3 cr.

Separation theorems and other properties of convex sets in finite-dimensional spaces. Formulation of nonlinear programming problems. Saddle-point (Lagrangian) optimality criteria for convex nonlinear programs. Duality theorems for convex programs. First, and second-order Kuhn-Tucker stationary-point theory for differentiable non-convex programs. Perturbation and sensitivity analysis. Applications and extensions. Prereq: Familiarity with basic mathematical analysis (e.g., Math 521) and either Math. 443 or 320, or consent of instructor.

727 Advanced Nonlinear Programming — 3 cr. (also IE)

Conjugate convex functions and Fenchel-Rockafellar duality. Monotone operators and subdifferentials. Advanced methods for nonconvex problems, such as variational principles, generalized gradients, degree and index arguments, and multivalued ordinary differential equations. Applications to economics and operations research. Prereq: CS 726 or consent of instructor. (Infrequently offered.)

730 Nonlinear Programming Algorithms — 3 cr.

Rigorous description, and convergence proofs of various nonlinear programming algorithms. Emphasis on algorithms that are important, can be proved to converge and are practical. Unification of classes of algorithms and convergence rates. Each student will code and test one of the algorithms described in the course. Prereq: Consent of instructor.

731 Advanced Artificial Intelligence — 3 cr.

Learning and hypothesis formation; knowledge acquisition; deductive and inductive inference systems; reasoning techniques involving time, nonmonotonic reasoning, spatial reasoning, truth maintenance systems; planning strategies. Prereq: CS 540.

733 Computational Methods for Large Sparse Systems 3 cr. (also Math & ECE)

Sparse matrices in engineering and science. Sparsity preservation. Numerical error control. Transversal algorithms, Tarjan's algorithm, Tinney's algorithms, minimum degree, banding, nested dissection, frontal methods. Linear and nonlinear equation solving. Compensation. Sparse vector methods. Iterative methods. ODE and PDE applications. Prereq: CS 367 and (ECE 334 or (CS 412 and Math 340)); or consent of instructor.

736 Advanced Operating Systems 3 cr.

Advanced topics in operating systems, including process communication, resource allocation, multiprocess and network operating systems, kernel philosophies, fault-tolerant systems, virtual machines, high-level language systems, verifiability and proof techniques. Prereq: CS 537 or consent of instructor.

737 Computer System Performance Evaluation and Modeling 3 cr.

Statistical techniques of computer system performance evaluation and measurement. System selection and tuning strategies. Deterministic and probabilistic models of process scheduling and resource allocation. Analytic and simulation models of computer systems. Systematic study of system architectures. Prereq: Math 222, CS 537 or CS 736, or consent of instructor.

739 Distributed Systems 3 cr.

Basic concepts, distributed programming; distributed file systems; atomic actions; fault tolerance, transactions, program & data replication, recovery; distributed machine architectures; security and authentication; load balancing and process migration; distributed debugging; distributed performance measurement; distributed simulation techniques; distributed applications; correctness considerations and proof systems. Prereq: CS 736 or consent of instructor. (Infrequently offered.)

740 Advanced Computer Networks 3 cr.

Advanced topics in computer communications networks: Congestion and flow control; Routing; Rate-based protocols; High-speed interfaces and technologies; Metropolitan area networks; Fast packet switching technologies; Advanced applications; Network services: name service, authentication, resource location. Prereq: CS 640.

747 Advanced Computer Systems Analysis Techniques 3 cr.

Advanced analytical modeling techniques for performance analysis of computer systems, including discrete-parameter (embedded) Markov Chains, M/G/1 queues, stochastic Petri nets, queueing networks, renewal theory, and sample path analysis. Application areas include high performance computer architectures, databases, and operating system resource allocation policies. Prereq: CS 547 or consent of instructor.

750 Real-Time Computing Systems 3cr. (also ECE)

Introduction to the unique issues in the design and analysis of computer systems for real-time applications. Hardware and software support for guaranteeing timeliness with and without failures. Resource management, time-constrained communication, scheduling and imprecise computations, real-time kernels and case studies. Prereq: CS 552 and 537 or consent of instructor.

752 Advanced Computer Architecture 3 cr.

Advanced techniques of computer design. Parallel processing and pipelining; multiprocessors, multicomputers and networks; high performance machines and special purpose processors; data flow architecture. Prereq: ECE/CS 552 and CS 537.

755 VLSI Systems Design 3 cr.

Overview of MOS devices and circuits; introduction to integrated circuit fabrication; topological design of data flow and control; interactive graphics layout; circuit simulation; system timing; organizational and architectural considerations; alternative implementation approaches; design project. Prereq: ECE 340, ECE/CS 352, and CS/ECE 552 or consent of instructor.

756 Computer-Aided Design for VLSI 3 cr.

Broad introduction to computer-aided design tools for VLSI, emphasizing implementation algorithms and data structures. Topics covered: design styles, layout editors, symbolic compaction, module generators, placement and routing, automatic synthesis, design-rule checking, circuit extraction, simulation and verification. Prereq: CS 367, good programming skills, CS 352; CS 755 strongly recommended. (Last taught: Spring 94)

757 Advanced Computer Architecture 3 cr.

Parallel algorithms, principles of parallelism detection and vectorizing compilers, interconnection networks, SIMD/MIMD machines, processor synchronization, data coherence, multis, dataflow machines, special purpose processors. Prereq: CS 752 or consent of instructor.

758 Advanced Topics in Computer Architecture 3 cr.

Advanced topics in computer architecture that explore the implications to architecture of forthcoming evolutionary and revolutionary changes in application demands, software paradigms, and hardware implementation technologies. Prereq: CS 752 and CS/ECEr 757 required. Alternatively, consent of instructor.

760 Machine Learning 3 cr.

Computational approaches to learning: including inductive inference, explanation-based learning, analogical learning, connectionism, and formal models. What it means to learn. Algorithms for learning. Comparison and evaluation of learning algorithms. Cognitive modeling and relevant psychological results. Prereq: CS 540.

764 Topics in Database Management Systems 3 cr.

Implementation of database management systems, the impact of new technology on database management systems, back-end database computers, distributed database management systems, concurrency control and query execution in both distributed and centralized systems, implementation of multiple user views, roll-back and recovery mechanisms, database translation. Prereq: CS 564, CS 537, and CS 536 or consent of instructor.

765 Perceptual Recognition 3 cr.

High-level perceptual processing by computer; recognition of complex objects and scenes; advanced computer vision systems; relation to the living visual system; algorithm-structured multi-computer architectures for perception; binocular and multi-modal vision; recognition and tracking of moving objects; learning in perceptual systems; perceptual-motor control of robots. Prereq: CS 731 or consent of instructor. (Last taught: Fall 91)

766 Computer Vision 3 cr.

Fundamentals of image analysis and computer vision; image acquisition and geometry; image enhancement; recovery of physical scene characteristics; shape-from techniques; segmentation and perceptual organization; representation and description of two-dimensional and three-dimensional objects; shape analysis; texture analysis; goal-directed and model-based systems; parallel algorithms and special-purpose architectures. Prereq: CS 540.

769 Advanced Natural Language Processing 3 cr.

Develop algorithms and mathematical models for natural language processing tasks, including text categorization, information retrieval, speech recognition, machine translation, and information extraction. Focus is on the state-of-the-art computational techniques as they are applied to natural language tasks. Prereq: CS 540 or the equivalent.

776 Advanced Bioinformatics 3 cr.

Advanced course covering computational problems in molecular biology. The course will study algorithms for problems such as: modeling sequence classes and features, phylogenetic tree construction, gene-expression data protein and RNA structure prediction, and whole-genome analysis and comparisons. Prereq: CS 576.

777 Computer Animation 3 cr.

Survey of technical issues in the creation of moving and dynamic computer imagery. Principles of animation. Manual motion specification and keyframing. Procedural and simulation-based motion synthesis. Motion capture processing, editing and use. Animation systems. Modeling, rendering and video issues relating to animation. Image-based animation methods and warping. Applications of animation such as games and virtual environments. Basic introduction to artistic issues in animation, such as cinematography. Special Effects for Film and Video. Prereq: CS 559.

779 Rendering Images with Computers 3 cr.

Survey of models and algorithms used in the computer generation of images. The physics of global illumination, the global illumination equation, approximations and techniques for solving them. Large database rendering. Image-based methods. Stylized rendering. Point-based (splatting) algorithms. Prereq: CS 559.

780 Robot Motion Planning 3 cr. (also ECE & ME)

A unified view on geometric, algorithmic, and computational issues of automatic motion planning of motion for mobile robots and arm manipulators in a complex environment. Planning with complete information - configuration space, connectivity graphs, computational complexity; with partial information - algorithm convergence, topological issues. Effect of system kinematics. Relation between sensing media and algorithm efficiency. Prereq: Math 340 or equivalent and consent of instructor. (Infrequently offered.)

784 Data Models and Languages 3 cr.

Study of database programming languages. Topics include: Logic based languages, embedded query languages, object-oriented languages. There will be coverage of types, persistence, inheritance, object identity, data models, implementation issues, and case studies of actual systems and languages. Prereq: CS 564 and CS 536 or consent of instructor.

787 Advanced Algorithms and Data Structures 3 cr.

Algorithms for graph manipulation, geometry, matrix multiplication, string processing, information retrieval, etc. Mathematical models and analyses. Lower bounds. Probabilistic, distributed, and parallel algorithms. Advanced data structures. Prereq: CS 577.

790 Master's Thesis 1–9 cr.

For students writing a Master's thesis or project. Prereq: Master's candidate.

799 Master's Research 1–9 cr.

For pre-Master's students doing research projects. Prereq: Master's candidate.

810 Models and Formalisms for Computation 3 cr.

Models of computation, Turing machines, recursive functions, Church's thesis, undecidable problems, degrees of unsolvability. Denotational semantics, logic of programs. Applications to automata, formal languages, program verification, programming languages, and complexity. Prereq: CS 520.

812 Arithmetic Algorithms 3 cr.

Design, implementation, and analysis of algorithms for exact arithmetic on arbitrarily large integers, Gaussian integers and rational numbers. Algorithms for modular arithmetic and internal arithmetic. Algorithms for integer greatest common divisor calculation and factorization. Classical and modern algorithms. Prereq: Math 541 and CS 367, or consent of instructor.

830 Abstract and Concrete Complexity Theory 3 cr.

Study of computation with limited resources (time, space, etc.). Complexity hierarchies, structure of P, NP, PSPACE, co-NP. Strong NP-completeness, isomorphism completeness, relativized complexity. Abstract complexity, probabilistic complexity. Lower bounds, time-space tradeoffs, pebbling, alternation. Prereq: CS 520. (Infrequently offered.)

837 Topics in Numerical Analysis 3 cr. (also Math)

Topic selected from advanced areas. A variable content course which may be repeated any number of times for credit. Prereq: Consent of instructor. (Infrequently offered.)

838 Topics in Computing 3 cr.

Topics selected from advanced areas. A variable content course which may be repeated any number of times for credit. Prereq: Consent of instructor. (Infrequently offered.)

880 Topics in Theoretical Computer Science 3 cr.

Advanced topics in algorithms, complexity, and models of computation, discussed in a seminar format. The exact topic varies. Prereq: consent of instructor. (Infrequently offered.)

887 Approximation Theory 3 cr. (also Math)

Interpolation and approximation by means of interpolation; uniform approximation; best approximation; approximation in normed linear spaces; spline functions; orthogonal polynomials; degree of approximation; computational procedures. Prereq: Consent of instructor.

899 Pre-Dissertator Research 1–9 cr.

Prereq: Post-Master's, pre-dissertator status.

915 Computation and Informatics in Biology and Medicine 1 cr. (also BMI, Gen, Biochem, CBE, and BME)

This seminar course brings together trainees, trainers, and other interested faculty and students for cross-disciplinary exposure to current research in computer science, biostatistics, engineering, biological sciences, and biomedical research problems related to bioinformatics and computational biology. Prereq: Consent of instructor.

990 Dissertation 1–6 cr.

Prereq: Dissertator status.

999 Independent Study and Research **1–6 cr.**

Prereq: Dissertator status.

Non-Credit Seminars

The nine research areas in the Department each run an advanced, non-credit seminar where graduate students, visitors, and faculty members from within and outside the Department present their latest research or discuss recently published papers. These seminars give graduate students the opportunity to learn about current research problems and to get valuable feedback on their own research.

Also, each year the Department runs a Distinguished Lecturer Series where 6-8 leading researchers in a subfield of computer science visit. The visitors give two lectures – one to a general computer science audience and a second, more specialized, talk targeted toward researchers in the given

Computing Facilities

The Department contains a large array of sophisticated computer hardware which supports both our research and instructional missions. This equipment is maintained by a central facility, the Computer Systems Laboratory.[18] We are continuously upgrading and enhancing our systems to offer the most up-to-date computing resources possible. Much of the equipment was donated by our industrial affiliates; their support has been invaluable in enabling us to develop a first-rate computing environment.

All faculty, supported graduate students (i.e., TA's, RA's, and Fellows), and staff have high-performance workstations on their desks. These machines include various models of Sun, Intel based and AMD based workstations. Desktop workstations run one of Windows 2000, RedHat Linux or Solaris.

The Department is recognized as a national leader in research on parallel and distributed computing. Current work involves experimental design of both parallel algorithms and computer architectures in support of a wide range of projects, including mathematical programming, parallel-program debugging tools, performance modeling and analysis, computer vision, databases and many others. The Topaz and PRISM projects, funded by NSF Institutional Infrastructure grants, have enabled the Department to acquire parallel hardware to enhance this work. Components of our parallel computing environment include clusters of 100 dual processor Xeon CPU's, 150 dual processor PIII's, 70 one processor Pentium 4's, a 16 node SP3, a 32 node Netra cluster and 6 SUN Enterprise servers used to perform simulations and experiments on large-scale parallel architectures and large-scale parallel computations.

A locally developed software package called Condor provides additional computing power for compute-bound tasks such as simulations. Condor automatically locates workstations which are idle and transfers jobs to them. The jobs are periodically checkpointed and migrate from machine to machine until completion. Studies of Condor showed that jobs submitted to Condor made use of over 180 CPU-days per week of otherwise wasted machine cycles.

All of our research and instructional facilities are connected to local area networks, each of which is connected to every other and to the Internet by routers. The network allows remote and automated use of departmental resources and information sharing. There are currently about 3 Terabytes of storage available to most of our machines through the AFS distributed file system. Much of the research information produced by the Department is made freely available to the world through our Web server.

In addition to the research facilities, the Department has a number of workstations to support work in undergraduate and graduate courses. These include 60 UltraSPARC 10 workstations, 100 Intel PIII based workstations running Windows 2000, 40 Intel PIII based workstations running RedHat Linux. Our instructional laboratories support more than 2000 students each semester.

[18] http://www.cs.wisc.edu/csl/

Additional Information

Requirements for a Ph.D. Minor in Computer Sciences

To obtain a graduate minor in Computer Sciences, a student must earn at least 12 credit hours in Computer Sciences courses, meeting the following requirements:

1. The courses presented for a minor must form a coherent plan of study, approved by the Computer Sciences Department's external advisor, who should be consulted for further details.

2. With the exception of CS 367, all courses must be numbered 400 and above. "Topics" courses, CS 837, 838, and 880, must be graded in the usual A-F manner.

3. At least one of the courses must involve a significant amount of programming in a structured language, such as C++, C, or Java. CS 367 will meet this requirement, as will many of the Systems and AI courses which have CS 367 has a prerequisite.

4. At least one of the courses must be numbered 700 or above (not including CS 900), and passed with a grade of at least a B. If this course is cross-listed with another department, the course must be taught by a regular CS Department faculty member. If an exception is necessary, advance approval is required.

5. The average grade in all Computer Sciences courses presented for a minor must be at least B.

Students planning to minor in Computer Sciences should consult with the Department's external advisor early in their graduate program to ensure acceptance of the minor program. A Minor Agreement form must be filed with their home Department.

Adding Computer Sciences as a Second Major (M.S. only)

The Computer Sciences Department will consider applications from graduate students, with uniformly excellent graduate records in CS courses, for addition of the CS major – *for the Master's Degree only*. (Students intending to get a Ph.D. in Computer Sciences or seeking long-term TA support must submit a standard admissions application, which requires a subject GRE test and, for non-native speakers, a TSE score.)

Students who wish to apply for admission to this program must meet the following conditions:

1. Be very near completion of a Graduate Degree from the U.W. in their major department.

   We require a letter from the major department attesting to the fact that the student will receive a Master's degree shortly or has attained dissertator status.

   Note: A student in dissertator status who is admitted to this program **loses** his or her dissertator status. This is a Graduate School rule that is enforced by the Graduate School.

2. Have obtained grades of AB or better in at least three CS courses numbered 500 or above in three different areas of computer science (see pp 6-7). At least one of these courses must be at the 700 level (not including CS 900). All of these courses must have been taught by a member of the CS Department faculty.

3. Have two letters of recommendation from CS Department faculty members.

4. At least one of the courses meeting requirement 2 must have a significant amount of programming in a structured language. This requirement is waived for those students who have also taken CS 367 (which cannot be counted toward the Master's degree in CS).

Note: Meeting these requirements guarantees that your application will be considered for admission to the program. However, it does not guarantee admission.

Students should also check the Graduate School publication *Guidelines for Joint, Double, and Dual Master's Degrees*.

Technical Reports

Over 1000 technical reports, dealing with various aspects of computer science, and written by faculty members and graduate students, have been published by the Department. A listing of the most recent reports is available via the World-Wide Web.[19]

Alumni Group

The Department has an Alumni Group, which publishes a newsletter, *Badger Bytes*, and an annual alumni directory. More information is available by emailing `alumni@cs.wisc.edu` or via the Alumni Group's World-Wide Web page.[20]

SACM – Student ACM Chapter

The student chapter of the Association of Computing Machinery (SACM)[21] provides numerous services for members of the Department, including: running an orientation for new graduate students; providing partial financial support to students attending conferences; organizing picnics each semester and a potluck dinner in the spring; organizing trips to the theater and sporting events; maintaining the Department's photo board; and running the Department's Coffee Club.

[19] http://www.cs.wisc.edu/tech_reports.html
[20] http://www.cs.wisc.edu/~alumni/
[21] http://www.cs.wisc.edu/~sacmuse/

Key Contacts (2006-2007)

The following people direct departmental activities that are relevant to CS graduate students:

Department Chair
Guri Sohi (sohi@cs.wisc.edu)

Associate Chair
Susan Horwitz (horwitz@cs.wisc.edu)

Graduate Admissions
Lisa Louden (admissions@cs.wisc.edu)

Graduate Coordinator
Lisa Louden (louden@cs.wisc.edu)

Assistant to the Chair (supervises TAs)
Perry Kivolowitz (perryk@cs.wisc.edu)

Graduate Advising Committee (GAC) Chairs
Robert Meyer (rrm@cs.wisc.edu) and Thomas Reps (reps@cs.wisc.edu)

SACM President
Ameet Soni (sacm@cs.wisc.edu)

Department Manager
Melody Bakken (melody@cs.wisc.edu)

Payroll and Benefits Coordinator (has copies of insurance forms, etc.)
Virginia Werner (ginny@cs.wisc.edu)

Director of the Computer Systems Laboratory
Paul Beebe (lab@cs.wisc.edu)

Technical Reports
Cathy Richard (tech-reports@cs.wisc.edu)

Travel Reimbursement
Cathy Richard (cathy@cs.wisc.edu)

General Information
email: cs@cs.wisc.edu
phone: (608) 262-1204
fax: (608) 262-9777
URL: http://www.cs.wisc.edu/
mailing address:

Computer Sciences Department
University of Wisconsin
1210 W. Dayton Street
Madison, WI 53706