# Computer Sciences Department
# Graduate Guidebook

**1992 - 1993**

**University of Wisconsin - Madison**
**Madison, Wisconsin**

*July 1992*

## Table of Contents

# Introduction

The Computer Sciences Department at the University of Wisconsin-Madison 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 Bulletin* provides 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 your individual needs and interests, and all degree requirements, are met.

During the 1991 calendar year 66 B.S., 58 M.S., and 17 Ph.D. degrees were conferred by the Department. There are approximately 170 undergraduate majors and 240 graduate students in the Department. Graduate student support currently includes approximately 20 fellowships, 50 research/project assistantships, and 100 teaching assistantships.

The Department currently has 39 faculty members and another 12 affiliated with the Department. Areas of specialization in the Department include artificial intelligence, computer architecture, computer networks, database systems, distributed systems, mathematical programming, modeling and analysis of computer systems, numerical analysis, operating systems, programming languages, and theory of computation. A separate publication, the *Computer Sciences Department Research Review,* lists the faculty in the Department, active research projects, computing facilities, and recent technical reports and colloquia.

## 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 *January 15.*

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

2. **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.

3. **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.

4. **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. All applications, whether or not they involve requests for financial aid, can be for the fall semester only and must be completed by January 15.

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 hundred 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 contact the Academic Computing Center (MACC) and individual departments on campus for information.

# M.S. Degree Requirements

The requirements for the M.S. degree have been altered recently. Students admitted to the Computer Sciences graduate program before January 1989 have the alternative of obtaining the M.S. degree by completing a previously filed plan.

In addition to the requirements of the Graduate School 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. Fifteen of these credits must be received for *core* courses: CS courses numbered 700-990 (excluding 702). No courses – including 838, 880, and 999 – may be counted more than once, though up to six credits of 990 may be counted if a Master's thesis is filed with the Department. A qualifying core course for which a grade of S is received may be counted only for non-core credit. Non-core courses will not be counted if a grade of S is received. Instructors often give only S/U grades for 880 and 838. At its discretion, the Graduate Advising Committee (GAC) may declare an individual section of 838 exempt from the repetition restriction.

During registration week of the first semester or summer session 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. 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 990 (thesis and research) 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 990 may be used to satisfy the M.S. requirements provided that a Master's project report is filed with the Department. Note that 990 can be counted as core credit. 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 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.

# Ph.D. Degree Requirements

The requirements for a Ph.D. in Computer Sciences include satisfying the general University residency and minor field requirements, 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 and II below. A student who has satisfied requirements I, II and III below, and has satisfied the Graduate School's residence requirement is officially a dissertator in Computer Sciences. Details of the Graduate School's requirements can be found in the *Graduate School Bulletin.*

**I. Qualifying Examination**

Each semester, the Department offers a four-hour *depth* exam and a two-hour *breadth* exam in each of the following areas:

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

Each exam is graded on a scale of P+ (high pass), P (pass), P− (low pass), or F (fail). There are two ways to pass the qualifying examination. One way is by taking the depth exam in one area and the breadth exams in two other areas. The second way is to pass the depth exams in the areas of mathematical programming and numerical analysis. In one area, the depth exam must be passed with a grade of P or better; competency in the other two areas is exhibited by passing their associated breadth exams, one with a grade of at least P and the other with a grade of at least P−. Students taking two depth exams in mathematical programming and numerical analysis must pass one with at least a P and the other with at least a P−.

A student is allowed at most two chances to pass each area (including any combination of depth and breadth exams in the area). If a student takes an exam twice, the second grade always supersedes the first.

The student must pass all exams 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 exams, 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 examination process without written approval from the chair of GAC.

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. Breadth exams are intended to demonstrate a basic understanding of an area by students concentrating in another area. A student should be able to pass the breadth exam in any area after completing two courses in that area. In some areas the syllabus will simply list two

500-level courses or one 500-level and one 700-level course, with the understanding that a student who has thoroughly mastered the material in these courses should be able to complete the breadth exam. Other areas may structure their exams to allow a student a choice of two from among three or more courses.

Qualifying examinations are currently offered early in each regular semester. Students are required to register 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. Late registration is permitted only by a departmental vote of the faculty.

One of the breadth exams may be given in a "special area" instead of in one of the areas listed above. The "special area" may be either a specially prepared area for the individual student or an area approved by another department and the associated exam administered by that department. In either case, the area must be approved by the Examination Committee and two faculty members in the Computer Sciences Department. Under normal circumstances a student wishing to take a "special area" examination as part of the qualifying process must petition for this privilege before the end of the 12th week of the semester before the semester proposed for the examination. This petition must include: (1) letters of support from two Computer Sciences faculty members, one of whom asserts that he or she would be willing to be the Ph.D. advisor for the student if the qualifying process is successfully completed, and (2) a description of the student's overall program and goals together with a statement of the relevance of the "special area."

The committee administering a depth area exam may, at its discretion, recommend both depth and breadth grades for a student's performance on the depth exam. For example, the committee might recommend a grade of F for the depth exam, but offer the student the option of counting the sitting as a P− on the corresponding breadth exam. All such recommendations must be approved by the Examination Committee. If the recommendation is approved, the student may choose which grade is to be counted towards the completion of the qualifying process.

### II. 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 exam must have a major professor certify that progress is being made toward the exam. In any case, the preliminary exam may not be taken more than 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.

### III. 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. One option (Option A) is to fulfill the minor field requirement as specified by another 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 majors who require a program not covered by an orthodox choice of courses in some other single department. This option is an interdepartmental minor for students whose program makes such a minor more desirable than the standard minor under the control of one particular department other than the student's major department. For Computer Sciences graduate majors the requirements for an Option B 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:

1. 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.
2. After completing the program, bring the preliminary examination warrant to the secretary in charge of student records in the departmental office. 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.

**IV. Dissertation**

The student must conduct a substantial piece of original research in computer science and report it in a dissertation meeting 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.

**V. 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.

# 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. Note, however, that it is permissible to take four. The beginning graduate student should select 400-, 500- or 700-level courses depending on 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 CS 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 so that USAP is possible should immediately discuss the problem with a member of GAC.

## II. Sources of Advice

1. 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.
2. 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.
3. Particular questions about a course can often best be resolved by contacting the course instructor.
4. 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

1. *Master's degree:* Most full-time students obtain this degree in three or four semesters.
2. *Doctoral qualifying examinations:* A student well prepared at time of admission might attempt at least some of these examinations as soon as the second semester after entry, but many students take the qualifying examinations during their fourth through sixth semesters after entry.
   Students should begin preparing for qualifying examinations well in advance. They are stiff examinations which many students fail. Topics lists for the various areas are available in the Department office. These topics lists contain suggestions for background reading. Also, students may find the file of previous examinations, available at Bob's Copy Shop, 1314 W. Johnson St., useful in determining what will be expected of them on the examinations. In general, each 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

examinations with appropriate faculty members once the areas of specialization have been decided.

3. *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.

4. *Oral defense of Ph.D. dissertation:* In ideal circumstances, the student will be at this final step four years (eight semesters) after entry; the student should certainly be here by the ninth or tenth semester as a graduate student.

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

## IV. Scheduling Courses in Particular Areas

1. *Artificial Intelligence*
   The Artificial Intelligence Qualifying Exam for Fall 1992 will cover materials and courses under the current policy. However, this is the last semester during which that policy will be valid. For Spring 1993 and later semesters, the new policy will hold. For the Fall 1992 examination students should use the existing reading lists. For examinations in Spring 1993 and later, extensively revised reading lists will be prepared. The Spring 1993 examination will be considered transitional and will consist of questions and question selection options suitable for students proceeding under either the old or new policies.

   CURRENT POLICY FOR ARTIFICIAL INTELLIGENCE QUALIFYING EXAMINATIONS
   Students should first take 540 if they have not had a comparable course as an undergraduate. 540 and 731 cover the material included in the AI breadth qualifying exam. Advanced work in AI divides into four main fields — computer vision; deductive problem solving, logic programming and expert systems; learning; and natural language processing. Vision is covered in 765 and 766. Problem solving, logic programming and expert systems are covered in 562, 761 and 762. Learning is covered in 760. Natural language processing is covered in 545, 771, 773 and 774. The AI depth exam is in two parts: a set of questions covering 540, 731 and associated readings, and a set of questions covering material in the four advanced fields. Students are allowed to select any combination of questions from the four fields.

   NEW POLICY FOR ARTIFICIAL INTELLIGENCE QUALIFYING EXAMINATIONS
   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:
   - Machine learning, CS 760.
   - Deductive problem solving and logic programming, CS 761.
   - Knowledge representation and expert systems, CS 762.
   - Computer vision, CS 766.

   Each of the 700-level courses has CS 540 (or a comparable course, e.g., CS731) 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 CS 540. If CS 731 is not available to them when they need it, they should take CS 540.

Students taking the AI Breadth Qualifying Examination will be prepared for it if they have mastered the material presented in CS 540 (or 731) and in one of the primary 700-level sub-area courses, and if they have assimilated the publications contained on the AI Breadth Reading List for AI in general and for their chosen sub-area.

The Breadth Examination will consist of three general AI questions (B540/731), two machine learning questions (B760), two deductive problem solving questions (B761), two knowledge representation questions (B762), and two computer vision questions (B766). To pass the examination, students will have to satisfactorily answer two of the three questions in the set B540/731 and both of the questions in one of the sets B760, B761, B762, and B766.

Students taking the AI Depth Qualifying Examination will be prepared for it if they have mastered the material presented in CS 540 (or 731) and in the primary 700-level courses for two or more of the separate sub-areas, and if they have assimilated the publications contained on the AI Breadth Reading List for AI in general and on both the AI Breadth and AI Depth reading lists for their chosen sub-areas, with special concentration in one of them.

The Depth Examination will contain the same B540/731, B760, B761, B762, and B766 question sets as does the Breadth Examination and in addition will contain two advanced questions in each of the sets D760, D761, D762, and D766. To pass the examination, students will have to satisfactorily answer two of the three questions in the set B540/731; both of the questions in one of the B sets (B760, B761, B762, and B766); both of the questions in the D set (D760, D761, D762, or D766) for the same sub-area; and two additional questions from any of the other B or D sets except B540/731. The two additional questions can be but do not have to be from the same set.

2. *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. 552 is the basic course. 752 and 757 cover advanced topics in computer architecture, 755 covers advance topics in VLSI design, and 756 covers design tools for VLSI. 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. 752 is usually offered by the Computer Sciences Department in the fall and by the ECE Department in the spring. 756 is offered in the fall semester alternately by the two departments. Though the same basic course, the emphasis in 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 breadth qualifying exam requires knowledge of material covered in 552 and 752. The depth qualifying exam requires knowledge of material covered in 552 and 752 and some extra readings, most of which are covered in 757.

3. *Database Systems*
   At the present time two courses are offered in the area of database systems. Students interested in a general introduction to database systems and their implementation should take 564. Students with an interest in more advanced topics and current research directions in the area, such as students preparing for either of the database qualifying exams, should take 764 after completing 564. Those students who are planning on taking the depth database 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.

4. *Mathematical Programming*
   Students should first take 525 unless they have had a thorough introductory course in linear programming. After completing a good linear programming course, students may take 719 or 720 (which emphasize combinatorial optimization) without further prerequisites, and students with a background in basic mathematical analysis may take 726, 727, or 730 (which emphasize continuous, nonlinear optimization). (Note that 525 is not a prerequisite for 726, 727 or 730.) Some background in probability is recommended for 723, which is not taught every year. All of the 700-level courses may be taken independently of each other. Ph.D. students planning on taking the depth qualifying examination in this area should be familiar with the material in 525 and 719, 720, 726, 727, 730 and optionally 723. For the breadth exam familiarity with the material in 525 and one of the 700-level courses is sufficient. Master's and Ph.D. students minoring in math programming should take 525 and at least one of the 700-level courses in the area.

5. *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, CS 737, and CS 747 are currently offered in this area. CS 547, which introduces analytic modeling methods and some elementary queueing 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 CS 747 are 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 CS 747. CS 547 and either CS 737 or CS 747 are required for the MA breadth exam. CS 547 and CS 747, plus additional reading on the Reading List, are required for the MA depth qualifying exam. Whenever possible, a CS 838 seminar is offered which covers papers on the reading list for the depth exam, advanced topics in performance modeling, and provides students with the opportunity to apply modeling and analysis techniques to problems in their field of interest.

6. *Numerical Analysis*
   The course sequence 513, 514 provides the basic introduction to numerical analysis and is a prerequisite for the breadth qualifying exam. Students planning to take the depth qualifying exam in numerical analysis should take the course sequence: 717, 712 and 713. This sequence also serves as the starting point for the more advanced courses, 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.

7. *Operating Systems*
   Three courses are offered in the area of operating systems. 537 introduces the subject, providing hands-on experience with building parts of operating systems in simple environments. There are several large programming assignments. 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. 739 takes up where 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. Both 537 and 736 are needed for the OS breadth qualifying examination. The depth exam covers material in these two courses and advanced material from 739 and 740 (computer networks).

8. *Programming Languages and Compilers*
   After finishing 367, students interested in the area of programming languages should take 536. This course provides an introduction to compilers and to programming-language implementation techniques. The graduate courses in the area are 701/2, 703, and 704. 701 is an extremely time-consuming course, during which each student builds a working compiler for a non-trivial language. Either 701 alone or 701/702 may be taken; the latter is a way to get 3 more credits at the expense of a relatively small amount of extra work. Ph.D. students usually take 701 alone; master's students usually take the combination. 703 covers a selection of advanced topics, such as alternative parsing methods and code optimization. 704 provides an introduction to the theory of programming languages, including the study of functional languages and formal language semantics. Occasionally, a section of 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 breadth qualifying exam, 536 and 701 suffice; for the depth qualifying exam, 704 is required and 703 is recommended.

9. *Theory of Computing*
   The breadth exam in theory covers concepts that are introduced in the undergraduate theory courses, 520 and 577. Graduate students who wish to take theory courses in preparation for the breadth exam should take the graduate theory courses 810 and 787, which cover some of the same concepts at a level suitable for graduate students. Students preparing for the depth exam in theory should take 810 and 787, and should cover topics from at least two courses in the following list: 709, 767, 812, 820, 830. Courses 709, 767, 787 and 812 focus on algorithms for specific problems, and courses 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 880, a special topics course, when offered.

## V. Transfer of Credits Toward the Master's Degree

1. *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:
   (a) 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.
   (b) 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.

2. *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 available for inspection and verification by the student. At the beginning of each registration week, they are placed in the GAC advising room, and at other times they are in the Department office. It is the responsibility of the student to see that these records are correct and up-to-date.

1. *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 doctoral qualifying examinations.

2. *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 which 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. For example, a mathematics course in advanced algebra would be approved for a student in hardware and software systems but not for a student in numerical analysis. In doubtful cases, GAC will require a note from a Department faculty member supporting the request for approval.

# 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. We will concentrate here primarily on the rules of the Department.

## I. Definitions

1. *Full-load and Part-load*
   Part-load students don't 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 or 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 9 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 9 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.

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

3. *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:

   (a) Department courses numbered 400 or above.

   (b) 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.

   (c) CS 302, 352, 354, and 367, and Math 221, 222, and 223, provided that the student has been admitted with deficiencies being removed by taking these courses.

(d) 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) before being admitted as a degree candidate are automatically counted toward our Master's program as approved courses.

4. *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.

5. *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 (if full load) or one (if part-load) 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 3 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.

### 132 Using Computers — 4 cr.

Introduction to use of computers. Some programs in Basic, but emphasis on productivity tools such as word processing, spreadsheets, graphics, and telecommunications. Consideration of societal impacts of computers. Not intended for Computer Sciences majors. Prereq: Minimum mathematical competency (two years of high school math). Open to Fr.

### 302 Algebraic Language Programming — 3 cr.

Construction of algorithms; problem solving; instruction and experience in the use of at least one procedure-oriented language (e.g., Pascal or Fortran); survey of other such languages, advanced programming techniques. Prereq: Advanced high school mathematical preparation or some college work in mathematics, statistics or logic; or consent of instructor. Open to Fr.

### 312 Elementary Numerical Methods — 3 cr.

Basic material in the numerical solution of various problems, including, but not limited to, interpolation and approximation of functions, root finding for non-linear equations, solution of linear systems of equations, numerical integration and differentiation, and the numerical solution of ordinary differential equations. Prereq: Math 223 and CS 302, or equivalent.

### 332 Using Computers — 4 cr.

Introduction to use of computers. Some programs in Basic, but emphasis on productivity tools such as word processing, spreadsheets, graphics, and telecommunications. Consideration of societal impacts of computers. Each student will attend two lectures and three pre-scheduled one hour laboratory sessions per week. This course is not intended for students majoring in computer science. It is intended for students majoring in Education; it meets with CS132 but special projects are assigned. Prereq: Junior standing in School of Education or Graduate standing in any college. Consent of instructor is mandatory.

### 352 Digital Systems Fundamentals — 4 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 302 or equivalent. Not open to students with EGR classification.

### 354 Machine Organization and Basic Systems — 4 cr. (ALSO ECE)

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 or consent of instructor. Open to Fr.

### 367 Introduction to Data Structures — 3 cr.

Study of data structures – specification, application and implementation. Stacks, queues, trees and other important structures. Application to garbage collection, dynamic storage allocation, sorting and searching, symbol tables, and arithmetic expressions. Emphasis on development and analysis of efficient algorithms, including use of structured programming methodology. Prereq: CS 302; Pascal or equiv.

**371 Technology of Computer-Based Business Systems** **3 cr. (ALSO BUS)**

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: Business 370, CS 303 and CS 367, or equivalent experience with consent of instructor.

**374 Introduction to Linear Programming and Business Applications1 cr. (ALSO BUS)**

Introduction to linear programming with applications in accounting, finance, marketing, production, and logistics. Prereq: Math 222. Not open for credit to students who receive credit for Math 210.

**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 223 and Math 340 or equivalent, CS 302 or equivalent. Not open for credit to students who receive credit for CS 312.

**458 Computer Graphics** **3 cr. (ALSO ME & ECE)**

Principles of computational geometry and computer graphics and their application; homogeneous coordinates, projective geometry, 3-D curve and surface representations; data structures and geometric databases, wire-frame and solid geometric representations; graphic I/O device characteristics and cost considerations. Programming exercises and projects based on applications. Prereq: CS 302 and 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.

**471 Introduction to Statistical Data Processing** **3 cr. (ALSO STAT)**

Organization and application of computers and statistical techniques to data processing. Data handling in terms of coding, preparation, acquisition (with and without computers), file organization and retrieval, screening and reduction summarization and tabulation, and analysis; computer-oriented sampling; statistical simulation; output and display of data; linear models and data processing (regression, correlation, covariance analysis, analysis of variance); nonlinear models in data processing and iterative analysis; survey of available programs and applications. Prereq: CS 302 and Stat. 310, 312 or 314.

**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.

**509 Mathematics for Computer Science** **3 cr.**

Program correctness and termination, invariants, pre- and post-conditions, axiomatic semantics. Representing time and space requirements of programs by summations, recurrences, and generating functions. Exact and asymptotic solutions. Analysis of algorithms for sorting, searching, and data structure traversal. Prereq: CS 367 and Math 222, or consent of the instructor.

### 513 Numerical Linear Algebra 3 cr. (ALSO MATH)

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. (ALSO MATH)

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.

### 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 and Math 222, or consent of the instructor.

### 525 Linear Programming Methods 3 cr. (ALSO IE, MATH, & STAT)

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. undergr, 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.

### 532 Theory and Applications of Pattern Recognition 3 cr. (ALSO ECE)

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 430 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 and ECE 333 or consent of instructor.

### 536 Introduction to Programming Languages and Compilers4 cr. undergr, 3 cr. grad.

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 CS 354.

**537 Introduction to Operating Systems** **4 cr. undergr, 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 367 and CS 354.

**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.

**540 Introduction to Artificial Intelligence** **4 cr. undergr, 3 cr. grad.**

Principles of knowledge-based search techniques; automatic deduction; knowledge representation using predicate logic, semantic networks, connectionist networks, frames, rules; Applications in problem solving, expert systems, game playing, vision, natural language understanding, learning, robotics; Lisp programming.
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 CS 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.

**552 Introduction to Computer Architecture** **3 cr. (ALSO ECE)**

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.

**562 Expert Systems: Design and Implementation** **3 cr.**

Design of expert knowledge bases: choice of appropriate representation, knowledge base structure, connection with data bases. Knowledge acquisition: possible knowledge sources, logical analysis and formalization, consistency and adequate checking. Inference engine construction and tailoring: choice and control of inferencing strategies, handling uncertain knowledge, achieving efficiency. Display and dialog design. Implementation of explanation capabilities. Evaluation of knowledge engineering environments and languages.
Prereq: CS 460 or CS 540.

**564 Database Management Systems: Design and Implementation** **4 cr. undergr, 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 CS 354.

**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 and Math 222, or consent of instructor.

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

Prereq: Consent of instructor.

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

Architecture and components of computer communications networks; protocol concepts and standards; OSI Reference Model; network/protocol architecture examples: Internet, ISO/CCITT, SNA, DECNET, public data networks; local area networks-gateways, internetworking, transport and application protocols.
Prereq: CS 537.

**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.

**702 Compiler Construction** **3 cr.**

Techniques for the implementation of compilers for sophisticated programming languages. Prereq: CS 536. Coreq: CS 701 or consent of instructor.

**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 equiv., or consent of 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.

### 712 Finite Difference Methods — 3 cr.

Development of finite difference methods for initial and boundary value problems for hyperbolic, parabolic, and elliptic partial differential equations. Analysis of accuracy and stability of difference schemes. Direct and iterative methods for solving elliptic difference schemes. Applications from science and engineering.
Prereq: CS 302, 412, Math 419 or equivalent, or consent of instructor.

### 713 Numerical Analysis of Differential Equations — 3 cr.

Analysis of numerical methods for ordinary differential equations. Single step and multistep methods. Stiff equations. Introduction to Galerkin methods; collocation, least squares, etc. Analysis of methods for the solution of large sparse systems of boundary value problems.
Prereq: Graduate standing and consent of instructor.

### 717(–718) Numerical Functional Analysis — 3 cr. per sem. (ALSO MATH)

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, 514 and Math 223 or consent of instructor.

### 719 Network Flows — 3 cr. (ALSO IE)

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.

### 726 Nonlinear Programming Theory and Applications — 3 cr. (ALSO IE & STAT)

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.

### 730 Nonlinear Programming Algorithms — 3 cr. (ALSO IE)

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.

### 732 Topics in Artificial Intelligence — 3 cr.

Advanced topics in artificial intelligence. A variable content course which may be repeated any number of times for credit. Prereq: Consent of instructor.

### 733 Computational Methods for Large Sparse Systems — 3 cr. (ALSO 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: CS367 and CS412, 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 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 536 or consent of instructor.

**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.

**752 Advanced Computer Architecture** **3 cr. (ALSO ECE)**

Advanced techniques of computer design. Parallel processing and pipelining; multiprocessors, multi--computers 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. (ALSO ECE)**

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. (ALSO ECE)**

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.

**757 Advanced Computer Architecture** **3 cr. (ALSO ECE)**

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.

**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 and either CS 731 or consent of instructor.

**761–762 Deduction and Problem Solving by Computer** **3 cr. per sem.**

Critical study and evaluation of programs and of man-computer interactive systems that play games such as chess, bridge, and GO; solve problems and puzzles; and prove theorems in various mathematical domains. Different kinds of languages, including natural language, usable for problem formulation and for solution planning and testing.
Prereq: CS 731 or consent of instructor.

**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 the 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.

**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 731 or consent of instructor.

**767 Graph Theory** **3 cr.**

Theory of graphs, including adjacency and incidence matrices, planarity, hamiltonian circuits, Euler's formula, directed graphs, and trees. The efficiency of the known algorithms for performing various operations on graphs. Prereq: CS 367 and either CS 475 or CS 577, or consent of the instructor.

**771 Computational Linguistics** **3 cr. (ALSO LING)**

Tools, techniques and literature of computational linguistics with applications in artificial intelligence and cognitive science. Topics include syntactic and semantic parsing, natural language understanding, text generation, machine translation, and speech recognition and production.
Prereq: CS 540 or CS 545 or Linguistics 530 or consent of instructor.

**773–774 Problems in Computational Linguistics** **3 cr. per sem. (ALSO LING)**

Current research in computational linguistics, coupled with original directed research.
Prereq: CS 545 or 771 or consent of instructor.

**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 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 or 509.

**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.

**813–814 Algebraic Algorithms** **3 cr. per sem. (ALSO MATH)**

Algorithms for arithmetic operations on multivariate polynomials with integral, rational and finite field coefficients. Polynomials remainder sequences and subresultants. Algorithms for multivariate polynomial resultant and greatest common divisor calculation. Arithmetic operations on multivariate rational functions. Algorithms for exact linear algebra of systems with polynomial coefficients. Exact calculation of the real and complex zeros of polynomials. Algorithms for factorization of polynomials into irreducibles. Calculations with real algebraic numbers. Quantifier elimination for real closed fields. Operations on formal power series.
Prereq: CS 812 and Math 541, 542, or consent of instructor.

**815 Transcendental Function Algorithms** **3 cr. (ALSO MATH)**

Formalism for function class representations; canonical and normal forms; diophantine sets and relation to recursively enumerable sets; Richardson's theorem; Lindemann theorems; canonical and normal form algorithms; fundamental operations on polynomials and rational functions with Gaussian integer coefficients; fundamental operations on various subclasses of the elementary transcendental functions; indefinite integration of rational functions; differential fields; Liouville integration theory; Risch integration algorithm.
Prereq: CS 510 and 814, or consent of instructor.

**820 Theory of Automata and Formal Languages** **3 cr.**

Advanced treatment of results concerning finite automata, regular sets, context-free and context-sensitive languages, Turing machines. Extensions of the models to multi-head, multi-tape, and probabilistic machines. Comparison of models. Applications to programming language design. Prereq: CS 520.

**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.

**837 Topics in Numerical Analysis** **3 cr. (ALSO MATH)**

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

**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.

**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 the instructor.

### 881–882 Numerical Methods for Ordinary Differential Equations
**3 cr. per sem. (ALSO MATH)**

Initial-value problems and boundary value problems for systems of nonlinear ordinary differential equations; convergence and stability of computational procedures; some topics chosen from the design of general computer programs, variational problems. Prereq: CS 514 and Math 417, or consent of instructor.

### 883–884 Numerical Methods for Partial Differential Equations
**3 cr. per sem. (ALSO MATH)**

Initial-value problems for systems of partial differential equations; boundary value problems, stability and convergence; iterative methods; applications to the classical problems of mathematical physics.
Prereq: CS 712 and 717, or consent of instructor.

### 885 Matrix Theory in Numerical Analysis — 3 cr. (ALSO MATH)

Methods for determining exclusion and inclusion regions for the eigenvalues of an $N \times N$ matrix. Variational methods for eigenvalues. The QR and LR algorithms. The Perron-Frobenius theory of positive matrices. Error analysis of numerical methods for computations in linear algebra.
Prereq: CS 717 or consent of instructor.

### 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.

### 990 Thesis and Research — 1–6 cr.

Prereq: Consent of instructor.

### 999 Independent Study and Research — 1–6 cr.

Prereq: Consent of instructor.

# Additional Information

**Seminars and Special Topics Courses**

Each semester a number of advanced seminars are held. The topics covered depend upon the current interests of permanent and visiting faculty members. Currently, weekly seminars are held in the areas of artificial intelligence, computer architecture, databases, distributed systems, performance modeling, programming languages, theoretical computer science, and numerical analysis. Seminars give graduate students the opportunity to learn about current research problems and to get feedback on their own research.

Advanced research topics are also covered in special topics courses which are offered each semester in CS 838. In particular, an 838 seminar is held in conjunction with the annual Distinguished Lecture Series presented each fall semester. Distinguished scholars have given lectures in various areas of computer science, including architecture, artificial intelligence, databases, operating systems, programming languages, theory, and optimization. The 1991-92 series was entitled "Machine Learning." Other recent 838 seminar topics include "Database Support for Advanced Applications," "Logic Programming and Expert Systems," "Parallel and Distributed Algorithms," "Advanced Programming Languages," "Distributed Systems and Performance," "Advanced Topics in Performance Modeling," "Mathemtical Programming in the Computer Sciences," and "Parallel Computer Architectures."

**Requirements for a 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 367, all courses must be numbered 400 and above.
3. At least one of the courses must involve a significant amount of programming in a structured language, such as C, Pascal, or Lisp. 367 will meet this requirement, as will many of the Systems and AI courses which have 367 has a prerequisite.
4. At least one of the courses must be numbered 700 or above, and passed with a grade of at least a B.
5. The average grade in all Computer Sciences courses presented for a minor must be at least B.

The student planning to minor in Computer Sciences should consult with the Department's external advisor and file a form with the Department and the Graduate School.

**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 given in the *Computer Sciences Department Research Review.*