COMPUTER SCIENCE (PhD, MS, GRADUATE CERTIFICATE)

The Faculty

Ethan Bolker, PhD, *Harvard University* • Performance Measurement and Modeling

William R Campbell, PhD, *University of St Andrews* • Object-Oriented Software Engineering • Programming Language Design and Implementation

Robert Cohen, PhD, *Brown University* • Information Security • Algorithms and Data Stuctures • Graph Drawing

Richard H Eckhouse, PhD, *State University of New York at Buffalo,* Professor Emeritus

Peter Fejer, PhD, *University of Chicago* • Theoretical Computer Science • Mathematical Logic

Robert A Morris, PhD, *Cornell University* • Digital Typography • Human Vision and Reading

Kenneth Newman, PhD, *Cornell University* • Computer Performance Modeling • Parallel Compilers

Elizabeth J O'Neil, PhD, *Harvard University* • Database Engines • Database Isolation Levels

Patrick O'Neil, PhD, *Rockefeller University* • Access Methods in Database Systems • Concurrency Control • Query Performance in Database Systems

Carl Offner, Industrial Professor, PhD, *Harvard University* • Compilers

Woojin Paik, PhD, *Syracuse University* • Artificial Intelligence • Natural Language Processing

Gregory Piatetsky-Shapiro, Adjunct Professor, PhD, *New York University* • Data Mining

Marc Pomplun, PhD, *University of Bielefeld* • Human Vision • Computer Vision • Human-Computer Interaction

Dan A Simovici, PhD, *University of Bucharest* • Data Mining • Database Design Theory • Multiple-Valued Logic

Richard L Tenney, Professor Emeritus, PhD, *Cornell University*

The Programs

The Department of Computer Science offers programs of graduate study leading to the PhD and the MS in computer science and a graduate certificate in database technology. Faculty interests in computer science include compilers, computability theory, computer communication protocols, database systems, formal languages, office automation systems and electronic publishing, operating systems, programming languages, queuing theory, semantics, and software engineering.

The computer laboratory of the department operates a network of SUN workstations using the UNIX operating system, as well as several other kinds of computing equipment. The department is also a member of the Internet. All graduate courses are scheduled in late afternoon and evening. The program is open to full-time and part-time students as well as those who want to strengthen specific skills by taking single courses.

The PhD Program

The Computer Science PhD Program prepares students for research careers in the software industry and in academia. It combines a commitment to theory with significant experience in software development. The areas in which students may carry out dissertation research are currently programming languages, databases, digital typography, neural networks, computer architecture, computer networks, and operating systems.

Degree Requirements

The program requires 48 credits of course work, a minimum of 15 credits of dissertation research, at least one year of full-time status, several examinations, and a doctoral dissertation containing original results. A GPA of 3.5 must be maintained at all times.

Several courses in theory (CS 720 and CS 724) and software development (CS 680-CS 683) are required.

Other requirements include the following:

A. After 30 credits of graduate work have been completed, each candidate takes a written examination to verify the breadth of his or her knowledge. The examination covers theoretical computer science and two of the following three areas:

   - Programming Languages
   - Databases
   - Systems

   Candidates who leave the program at this point will be awarded a master's degree in computer science if their course work satisfies the requirements specified for the MS degree in Computer Science.

B. After 48 credits of graduate work have been completed, and after passing the oral qualifying examination, each candidate presents a dissertation proposal to the thesis committee. This proposal must be presented within one year of passing the oral qualifying examination. The thesis committee may accept or reject the proposal.

All examinations may be retaken once.

The Major Advisor and the Thesis Committee

Every incoming PhD student will have a temporary advisor appointed by the department, until a major advisor is appointed through the procedure described below.

When a candidate has found a faculty member with whom to work, the faculty member (referred to as the major advisor) applies to the computer science graduate faculty for permission to direct the dissertation. The selection of the major advisor must take place no later than the semester when the student intends to take the oral examination. The major advisor is appointed by the graduate program director after approval by a meeting of the Computer Science graduate faculty. The major advisor is then responsible for monitoring the candidate's progress. After the candidate passes the oral qualifying examination, the major advisor will propose a thesis committee. This committee is appointed immediately after the student passes the oral qualifying examination.

The program must be completed in seven years.

Admission Requirements

Please see the general statement of admission requirements for all graduate studies programs in the "Admissions" section of this publication. All applicants for the PhD program must have the equivalent of a bachelor's degree in computer science and must take the general Graduate Record Examination. Foreign-educated applicants must submit TOEFL scores. The department reviews applications and recommends candidates to the Dean of Graduate Studies for admission to the PhD program.

Computer Science

The Master of Science Program

The MS program is intended as preparation for professional careers in research and commercial applications, although it also provides the background for further graduate work leading to the PhD degree. The program stresses the integration of theoretical knowledge with practical applications. The central course in the curriculum is a two-semester sequence in software engineering.

Degree Requirements

Candidates must complete a minimum of 30 credits, at least 24 of which must be in courses numbered 600 or above.

CS 680, followed by CS 681 and its co-requisite lab (CS 682) and then by CS 683 (another lab course), constitute a required Advanced Software Development sequence that carries a total of 12 credits. Students must take this sequence during the final semesters of their course work. Students are also required to choose two theoretical electives and four applied electives from among the following:

1) Theoretical Electives:

CS 420	(Introduction to Theoretical Computer Science)
CS 620	(Theory of Computation)
CS 622	(Theory of Formal Languages)
CS 624	(Analysis of Algorithms)
CS 720	(Logical Foundations of Computer Science)
CS 724	(Topics in Algorithm Theory and Design)

or

MATH 470 (Mathematical Logic)

2) Applied Electives:

CS 615	(User Interface Design)
CS 630	(Database Management Systems)
CS 634	(Architecture of Database Systems)
CS 637	(Database-backed Web Sites)
CS 639	(XML and Semi-structured Data on the Web)
CS 641	(Computer Architecture)
CS 644	(Operating Systems)
CS 646	(Computer Communications Networks)
CS 647	(Multimedia Networking)
CS 651	(Compilers)
CS 662	(Document Preparation and Text Processing Systems)
CS 670	(Artificial Intelligence)
CS 672	(Neural Networks)
CS 674	(Natural Language Processing)
CS 734	(Database System Internals)
CS 741	(High Performance Computer Architectures)
CS 750	(Implementation of Very High Level Programming Languages)
CS 752	(Parallel Programming)

or the following undergraduate courses:

CS 444	(Operating Systems)
CS 445	(Real-time Systems)
CS 460	(Graphics)
CS 470	(Artificial Intelligence)

Please note: No more than two upper-level (400-level) undergraduate courses may be used for graduate credit in the MS program.

Capstone Requirement

To complete the MS program, students must participate in a software engineering project by taking the required software development labs, CS 681 and CS 683. In general, these labs are completed during the last two semesters in the MS program. The project is approved by a committee that consists of two faculty members (professors who are currently teaching the software engineering course and supervise the software engineering laboratory) and the graduate program director. Students must submit the documentation for the projects and give a final oral presentation. The final project documentation will contain a signatory page containing the signatures of all members of the committee; a copy of the project will be retained by the department.

In exceptional circumstances, students with significant industrial experience may request a waiver of the software engineering requirement by applying to a faculty committee established for this purpose. As a part of the waiver application, the student must present a portfolio demonstrating the nature of this experience. Students who receive a waiver will be required to complete a master's thesis.

Minimum Requirements for Graduate Credit

The minimum grade for graduate credit is C. No more than two grades below B- may count for credit. Students may register for three credits for CS 699 (Research for MS Thesis) in order to write an MS thesis with the approval of the director of the program. This option is open for students whose grade point average is at least 3.5.

Students must maintain a grade point average (GPA) of at least 3.0 during their studies. The computation of the GPA takes into account all graduate courses and all upper-level undergraduate courses that may count toward the program requirements.

Admission Requirements

Please see the general statement of admission requirements for all graduate studies programs in the "Admissions" section of this publication.

Entrance requirements include a good working knowledge of computer science and mathematics. Each applicant's background will be assessed individually with respect to these skills, and specific requirements for making up deficiencies in preparation will be determined at an initial interview. The program requires the GRE test and, for students educated in non-English-speaking countries, the TOEFL test.

The MS program is designed to accommodate students with a wide range of backgrounds. In particular, the undergraduate degree need not be in computer science. Students may make up deficiencies by taking specific undergraduate courses. A typical set of such recommended courses includes

CS 110 and CS 210	(a sequence of introductory programming courses)
CS 240	(Programming in C and Assembler Language)
CS 310	(Advanced Data Structures and Algorithms)
CS 320	(Applied Discrete Mathematics)

Other undergraduate courses may be necessary, depending on the background and experience of the candidate. These courses are described in the Computer Science Information booklet available from the program office.

Graduate students who take courses at UMass Boston to satisfy undergraduate prerequisite course requirements must earn a B or better in each such course.

Computer Science

Graduate Certificate in Database Technology

This program is geared toward programmers and software professionals who wish to acquire a systematic education in database technology. It provides a solid anchoring in state-of-the-art database technology and trains its students to become database application developers or database administrators. A bachelor's degree is required. The candidates should have substantial programming experience; knowledge of databases is not expected.

The certificate requires 12 credit hours (4 courses).

The basic training comprises a sequence of three courses:

CS 630	(Database Management Systems)
CS 615 and	(User Interface Design)
CS 634	(Architecture of Database Systems)

These courses provide a thorough introduction to relational and object-relational databases, SQL, indexing, concurrency and recovery, and the design of user interfaces for database applications. Each student also selects one advanced database elective depending on specific interests.

For application materials or to arrange a personal interview, please contact the Computer Science Graduate Programs at 617.287.6440, or send email to dsim@cs.umb.edu.

Prospective applicants will be invited for an interview with a faculty member to determine whether they have an appropriate background for the certificate. The admission to the certificate program is determined by a computer science faculty committee.

A Note on Courses:

Please note that CS 310 or its equivalent is a general prerequisite for all graduate courses in computer science.

Courses

CS 615
User Interface Design

An introduction to user interface design, which encompasses design of the user interface and the functional design of the whole system. Students read and critique papers and articles, evaluate and critique existing user interfaces, and design interfaces of their own. Working in small groups, students use either interface prototyping tools or conventional rapid prototyping systems to construct an experimental interface.
Prerequisites: CS 320, or permission of instructor.
3 Lect Hrs, 3 Credits

CS 620
Theory of Computation

Functions computable by programs. Recursive functions and Turing machines; simulation and diagonalization. Universality and unsolvable problems. Kleene's hierarchy and the recursion theorem. Gregorczyk's hierarchy and Ackermann's function. Abstract complexity. Formal languages and classes of automata. Inherently difficult combinatorial problems.
Prerequisites: CS 320, or permission of instructor.
3 Lect Hrs, 3 Credits

CS 622
Theory of Formal Languages

This course treats languages from an abstract point of view as defined by formal grammars and by families of abstract machines. The Chomsky hierarchy and associated automata are covered. Emphasis is placed on context-free languages. Careful mathematical definition and proof are stressed throughout. This course does not involve programming. This course is of special interest to students interested in linguistics and in the theory of programming language compilers.
Prerequisite: CS 320, CS 450, or permission of instructor.
3 Lect Hrs, 3 Credits

CS 624
Analysis of Algorithms

Basic techniques for designing algorithms: divide and conquer, the greedy method, dynamic programming, etc. Applications to searching and sorting algorithms. Complexity of parsing. The fast Fourier transform and its applications (evaluation of polynomials and arithmetical problems). Lower bound theory. NP-hard and NP-complete problems. Probabilistic estimates of algorithms.
Prerequisite: CS 320 or permission of instructor.
3 Lect Hrs, 3 Credits

CS 630
Database Management Systems

Databases and database management systems. The entity/relationship model. The relational model. Relational algebra. The query language SQL. The object-relational model and SQL3. Embedded SQL in programs and dynamic SQL. Database administration: creating views and integrity constraints, handling data security. Functional dependencies. Normalization.
Prerequisite: CS 310 or permission of instructor.
3 Lect Hrs, 3 Credits

CS 634
Architecture of Database Systems

Indexing and query optimization in database systems. Writing programs to update a database. ACID properties. Concurrency theory: serializability, 2-phase locking, deadlock detection. Transactional Recovery: REDO and UNDO logging, different checkpoint approaches, media recovery. Examples of recovery utility use in INGRES, ORACLE and DB2. Transactional performance: the TPC-A benchmark, analysis of bottlenecks, and cost-performance considerations. Distributed database systems. Two-phase commit. Database parallelism.
Prerequisite: CS 430/630 or permission of instructor.
3 Lect Hrs, 3 Credits

Computer Science

CS 637
Database-Backed Websites
The design and implementation of database-backed websites. Static sites, dynamic sites, and sites that act as interfaces to relational database systems, providing for web-based collaboration through scalable online communities. Students install and maintain their own web servers, extend existing tool sets, and build their own sites from scratch in a series of intensive programming projects.
Prerequisites: CS 430/CS 630 and either CS 451 or CS 651.
3 Lect Hrs, 3 Credits

CS 639
XML and Semi-Structured Data on the Web
The eXtensible Markup Language (XML) smooths Web programming by providing a clear separation of presentation from structure in documents. This course surveys XML and semi-structure data technologies with the goal of understanding the problems and solutions arising from combining data from multiple sites and on-line databases. Students will learn the fundamentals of XML, the stylesheet and transformating language XSLT, the schema definition language X-Schema, tools that support Java-based XML programming, and some specialized applications such as Wireless Application Protocol. Focus is on manipulation of XML for data exchange, resource discovery, and the building of interactive web applications.
Prerequisites: CS 451/651, fluency in Java, and permission of the instructor.

CS 641
Computer Architecture
An examination of the designs for hierarchical memory systems including caches and virtual memory systems, pipeline design techniques, characteristics of RISC/CISC machines, multi-computer systems including multiprocessors and loosely-coupled computer systems, the micro engine and microprogrammed machines, vector and array processors, and the cost/performance tradeoffs in all of the above designs.
Prerequisite: CS 310 or permission of instructor.
3 Lect Hrs, 3 Credits

CS 644
Operating Systems
Structure and dynamics of operating system software. Operating systems as event driven software: interrupt processing and asynchronous operation. Memory management, scheduling, concurrency, consideration, device drivers. UNIX as a major example.
Prerequisite: CS 641 or permission of the instructor.
3 Lect Hrs, 3 Credits

CS 646
Computer Communications Networks
Need for computer networks. Architectures of networks. Architectures of systems. ISO Reference Model. Standardization efforts. Specification of protocols. Example of protocols and networks.
Prerequisite: CS 641 or permission of instructor.
3 Lect Hrs, 3 Credits

CS 647
Multimedia Networking
Network service requirements for streaming media and interactive media applications are analyzed. Audio and video coding and compression algorithms are surveyed. Challenges and solutions for delivering continuous media over today's best-effort Internet are investigated. Protocols for establishing and controlling multimedia sessions, for transporting continuous media end-to-end, and for wide-area dissemination of multimedia data are also investigated. Evolving Internet services models for establishing and maintaining levels of quality-of-service are evaluated. Students are expected to form groups to complete a significant semester project involving an investigation and development of a prototype.
3 Lect Hrs, 3 Credits

CS 648
Wireless Networks and Mobile Computing
Wireless communications, wireless networking, mobility management technologies, and protocols for wireless LANs and WANs are surveyed. Selected mobile computing models and mobile applications development environments are evaluated. A wireless networks laboratory provides a realistic mobile/wireless computing environment. Students are expected to form groups to complete a semester project involving an investigation and the development of a prototype.
Prerequisite: CS 644 or CS 646.
3 Lect Hrs, 3 Credits

CS 451/651
Compilers
Compiler organization and construction. Programming projects involve scanning input, analyzing program structure, error checking, code translation and interpreting, code generation and optimization. These projects result in a compiler for a reasonably large subset of ALGOL, Pascal, or similar procedural language.
Prerequisite: Either CS 420 or CS 622.
3 Lect Hrs, 3 Credits

CS 662
Document Preparation and Text Processing Systems
An applied course in contemporary document preparation systems. This course varies in content, with topics chosen from among the study of interactive editors, text formatters, typesetting systems, digital font design and production, publication graphics system, and author assistance software. Students participate in a major team project to design and implement a substantial portion of a system appropriate to the topic. Important current systems such as TeX, MetaFONT, TROFF, Scribe, and EMACS are studied and criticized, where possible by a study of their source code.
Prerequisites: At least one course at the level of CS 444 or above.
3 Lect Hrs, 3 Credits

CS 670
Artificial Intelligence
A broad technical introduction to the techniques that enable computers to behave intelligently: problem solving and game playing, knowledge representation and reasoning, planning and decision making, learning, perception and interpretation. The application of these techniques to real-world systems, with some programming in LISP.
Prerequisites: CS 320, MATH 470.
3 Lect Hrs, 3 Credits

Computer Science

CS 672
Neural Networks
An introduction to artificial neural networks. Topics include a survey of natural neural network models, perceptrons and their limitations, multi-layer networks and back propagation, Hebbian learning, unsupervised competitive learning, relations to automata and computability theory, adaptive resonance theory, applications of connectionist models of computing to various domains, including pattern recognition and databases.
Please note: An understanding of multivariable calculus and associated linear algebra, including gradient methods, is required. Some exposure to statistics and probability is advised.
Prerequisite: Permission of instructor.
3 Lect Hrs, 3 Credits

CS 674
Natural Language Processing (NLP)
The course provides the basic principle and theoretical issues underlying Natural Language Processing (NLP). It provides information on techniques and tools used to develop practical, robust systems that can communicate with users in multiple languages. The course will also provide insights into many open research problems in natural language such as information extraction, statistical corpus analysis, machine translation, speech processing, and text summarization.
Prerequisite: CS 420.
3 Lect Hrs, 3 Credits

CS 675
Computer Vision
This course provides both theoretical knowledge and practical experience with fundamental and advanced Computer Vision algorithms. Topics range from basic image processing techniques such as image convolution and region and edge detection to more complex vision algorithms for contour detection, depth perception, dynamic vision, and object recognition. Students will implement vision algorithms in the JAVA programming language. The performance of these programs is evaluated, and the advantages and disadvantages of individual approaches are discussed. The final project is the development by students of their own computer vision program solving a given problem.
Prerequisite: CS 320 or permission of instructor.
3 Lect Hrs, 3 Credits

CS 680
Object-Oriented Design and Programming
Object-oriented techniques for the design and development of software. Students develop a series of moderate sized programs in C++ and Java and then design and implement a more substantial project in small teams. Please note: Completion of this course with a grade of B or better is required for admission to the remainder of the Advanced Software Development sequence (CS 681, CS 682, and CS 683). Please note: This course is offered every fall and spring semester. Anyone receiving less than a B may not re-enroll in CS 680 before taking and passing an intervening advanced programming course. The sequence CS 680-683 was previously numbered CS 610-613.
Prerequisites: CS 651 or equivalent and permission of instructor.
3 Lect Hrs, 3 Credits

CS 681
Object-Oriented Software Development
This course, CS 682, and CS 683 form an integrated one-year sequence. In these courses students work in small teams, each team taking the full year to create a useful, complete, medium-sized software system for real customers. There is a strong emphasis on process: each project makes systematic use of an object-oriented development methodology based on UML models and incremental development. This course is offered every fall semester.
Prerequisites: A grade of at least B in CS 680 and permission of the instructor.
3 Lect Hrs, 3 Credits

CS 682
Software Development Laboratory I
First half of a two-semester laboratory course in which students, working in small groups, specify, design, implement, and document a large software project.
This course is offered every fall semester.
Prerequisite: CS 680.
Corequisite: CS 681.
3 Credits

CS 683
Software Development Laboratory II
A continuation of CS 682.
This course is offered every spring semester and must be taken during the semester following CS 681 and CS 682.
Prerequisites: CS 681, CS 682.
3 Credits

CS 697
Special Topics
Topics of current interest in the field, according to student and faculty areas of specialization. Course content varies according to the topic and will be announced prior to registration.
1 to 3 Credits

CS698
Practicum
This course is intended to enhance academic studies by providing an industrial context for learning new concepts and skills. It will help to prepare the student for the transition from an academic program to eventual employment in the computer industry. This course is not open to graduate students in the PhD track.
Prerequisite: Approval of Graduate Program Director.
1 Credit

CS 699
Research for MS Thesis
A one-semester supervised practicum course to help students complete the required thesis, which must be a substantial piece of research on some aspect of computer science. The master's thesis may take the form of a theoretical paper or a report on a piece of software.
Prerequisite: 15 graduate credits.
Hrs by arrangement, 3 Credits

CS 720
Logical Foundations of Computer Science
The course treats the logical foundations of computer science in a mathematically rigorous way but with emphasis on the applications of logic in computer science. Topics include the syntax and semantics of predicate logic, formal systems for predicate logic, many-sorted logic, and logic programming. Additional topics may include equational logic, algebraic specification, term rewriting, program verification, nonstandard logic, and databases. Students are expected to demonstrate an understanding of theoretical material and the ability to apply it.
Prerequisites: MATH 470 or equivalent, and permission of instructor.
3 Lect Hrs, 3 Credits

Computer Science

CS 724
Topics in Algorithm Theory and Design
An examination of current topics in algorithm analysis and design: complexity classes, abstract complexity theory, generating functions. Topics may include genetic algorithms, string matching algorithms, and circuit complexity.
Prerequisites: CS 624 or equivalent, and permission of instructor.
3 Lect Hrs, 3 Credits

CS 734
Database System Internals
This is a course in database internals design and programming. Students are expected to have already mastered a basic DBA-level understanding of database systems. The course is intended for students who want to program at the system level at a database vendor or go for a PhD in database systems with a practical orientation. Topics include implementation techniques for disk buffering, indexing, and transactional concurrency.
Prerequisites: CS 634 or permission of instructor.
3 Lect Hrs, 3 Credits

CS 738
Data Mining
Data mining is the process of secondary analysis to search for unsuspected relationships that are of interest or value for decision making. It aims at discovering association rules, episode rules, sequential rules, and other knowledge embedded in data and is concerned with efficient data structures and algorithms that have good scaling properties. The course presents essential aspects of data mining as a part of our current offering in databases; it includes a data mining project and prepares students to program applications that use data mining techniques.
Prerequisites: CS 630 and permission of instructor.
3 Lect Hrs, 3 Credits

CS 741
High Performance Computer Architectures
High performance computer architectures achieve an increase in performance with increasing system resources. System resources are scaled by the number of processors used, the memory capacity, the access latency tolerated, the I/O bandwidth required, the performance level desired, and other considerations. Scalable architectures delivering a sustained performance are desirable in both sequential and parallel computers. Parallel architecture has a higher potential to deliver scalable performance. The scalability varies with different architecture/algorithm combinations. Both hardware and software issues need to be studied in building scalable computer systems. The issues are examined in the light of both research and commercial parallel systems.
Prerequisite: CS 641.
3 Lect Hrs, 3 Credits

CS 750
Implementation of Very High Level Programming Languages
This course examines the issues and techniques that apply to the implementation of very high level programming languages—languages whose semantics are determined more at run time than at compile time. Topics include interpreters and their performance, definitional interpreters, storage management, and garbage collection. Participants study implementations of specific languages, for example Java, Smalltalk, Scheme, CLOS, Self, or Prolog.
Prerequisites: CS 651 and permission of instructor.
3 Lect Hrs, 3 Credits

CS 752
Parallel Programming
This course introduces the issues involved in parallel programming systems, including ease of programming, match between programming language and problem domain, and efficiency of the generated code. It explores and compares several parallel programming paradigms and investigates algorithms for scheduling parallel programs and for automatic parallelization of serial programs.
Prerequisite: CS 651 and permission of the instructor.
3 Lect Hrs, 3 Credits

CS 768
Color Science for Computer Graphic Applications
Topics include color spaces; device independent imaging; and international standards for describing, encoding, and transmitting color images. Students learn about spectral power distributions of light sources; models of human color vision; CIE tri-stimulus values; Lab and Luv; RGB and CYMK image models, and the relation between them. Programming exercises are usually in Mathlab.
Prerequisites: Permission of instructor.
3 Lect Hrs, 3 Credits

CS 899
PhD Dissertation Research
Research, conducted under faculty supervision, which leads to the presentation of a doctoral dissertation. This course carries variable credit and can be taken more than once.
Prerequisite: Permission of Graduate Program Director and instructor.
1 to 9 Credits