UNDERGRADUATE PROGRAM INFORMATION
Department of Computer Science
The UMass Boston Undergraduate Computer Science Program

The Department of Computer Science offers a BS degree and a BA degree. Our BS program is accredited by ABET which assures quality for degree programs in applied science, computing, engineering, and technology. The BS provides a rigorous education in theory and practice of computer science in addition to outside science courses. Our BA program provides a similar education in the theory and practice of computer science with more of an emphasis on outside humanities. Either degree can be used as a basis for many jobs in industry or for graduate work in computer science and other fields. We have an Honors Program for students who complete an Honors Thesis.

We offer an undergraduate Computer Science certificate program for non-degree seeking students or for matriculated students who are majoring in other fields. The CS certificate may be beneficial for students having undergraduate degrees in other fields who want to demonstrate their competence in computer science. The CS certificate also provides a fundamental core education in computer science that may be appropriate for students who need a computer science education in their career field. We also provide CS105 “Computer Concepts” introducing computer literacy for non-majors.

We offer a curriculum stressing software development. It provides training appropriate for students with interests in areas such as systems programming, compiler development, artificial intelligence, database management, and software engineering. An aptitude for logical reasoning and mathematics is needed to complete the major successfully, partly because designing programs to solve problems requires good problem solving skills and partly because some of the requirements are relatively advanced theoretical computer science and mathematics courses.

The Computer Science Department and College of Science and Math, jointly with the College of Management, offers a BS degree in Information Technology. Students interested in the BSIT degree should contact us about this program and get the Undergraduate IT Program Information Booklet.
The College of Management also provides a Masters’ Program in Information Technology. For more information go to http://www.umb.edu/academics/cm/masters_programs/msit

This booklet contains descriptions of our computer science courses and related mathematics courses, lists of the requirements for the major, the honors option, the certificate program, and other related information. Anyone considering majoring in computer science will want to read this document. Non-majors interested in computer course work will also find this booklet helpful in choosing appropriate courses. Prospective graduate students should consult the booklet describing the Master’s program. Everyone should consult the UMass Boston website, www.umb.edu, for general information about UMass Boston. Tentative course schedules for the coming semester (which courses will be offered and when) are usually available on the department’s website, www.cs.umb.edu. These are subject to change until (and sometimes after) they are included in the official UMass Boston schedule listed on WISER during the pre-registration period.

The material in this booklet provides only a general description of the computer science program and of the individual courses. Requirements and course content may change from time to time. It is important that all students receive individual advice from the faculty and students should come to the department in order to be assigned a faculty advisor. You can obtain further information by coming to the department office (Room S-3-132, on the third floor of the Science Building), by calling us at (617) 287-6441, or by visiting our web site at www.cs.umb.edu.

Learning Objectives in the BS in Computer Science Program

As part of the accreditation process, the department has established the following set of learning objectives for students enrolled in the Bachelor of Science Program in Computer Science.

1. To provide our students with a working knowledge of basic CS principles and computing practices
Students will:

a. demonstrate proficiency in problem solving, software design and development, data structures, algorithms, computer organization and computer architecture;
b. apply this proficiency to mid-sized applications like compilers;
c. write, document and test programs, in a variety of programming languages, supporting different programming paradigms;
d. be proficient in at least one (in-demand) programming language; and
e. learn to deal with different operating systems and computer architectures.

2. To expose our students to the theoretical underpinnings of computing and to mathematical and scientific thought and practices.

Students will:

a. understand and work with some of the important theoretical underpinnings of computer science in discrete mathematics and formal languages;
b. demonstrate an understanding of differential and integral calculus, linear algebra, probability and statistics; and
c. demonstrate the ability to do laboratory work in at least one of the natural sciences.

3. To give our students an understanding of the human and social issues, enabling them to be informed and active members of their communities and thoughtful CS professionals.

Students will:

a. become familiar with the basic concepts and issues in the social sciences and the humanities; and
b. demonstrate the ability to understand, write about and speak about the social and ethical issues of computing;

4. To provide our students with appropriate organizational and communication skills

Students will be able to:

a. write about their programs and issues related to computing;

b. orally present material related to computing; and

5. To prepare our students for rewarding employment and (in some cases) for graduate programs

Graduates of the BS Program in Computer Science will

a. be recruited for rewarding jobs in high-tech companies in Massachusetts and elsewhere;

b. be prepared to pursue graduate work in computer science; and

c. be prepared to learn new skills as required in a continually changing discipline.

These objectives were either inspired by, or come directly from, those of Ohio State’s objectives for their CS Engineering students; see http://www.cis.ohio-state.edu/ugrad/ugsc/programs/cseobjectives.html

Registration

Currently enrolled students who wish to major in computer science should file a Declaration of Major form with the Registrar. This form, which may be obtained from the Registrar’s Office, the Computer Science Office (S-3-132), or here, requires the signature of the department chair before returning to the Registrar.

New students who wish to work towards a B.A. or B.S. at UMass Boston must file the appropriate application with the Admissions Office. New students, who are not seeking a degree, but wish to
earn the certificate in computer science, must file a Certificate Program Application form with the One-Stop Center. The form, which is available online or in the Computer Science Office (S-3-132), must first be signed by the department chair. New students who wish merely to take courses may enroll as non-degree students. Transfer students should contact the Undergraduate Program Director, Ming Ouyang, who can answer any questions about transfer credit for computer science course work done at another institution.

The Undergraduate Computer Science Program

Part 1: Computer Science Course Descriptions

The computer science courses offered by the Department of Computer Science are listed below. Each has a number of the form CS xyz. The first digit x indicates the level of the course. The second digit y indicates the technical area, using the following scheme:

0  Non-technical
1  Programming/software engineering
2  Theoretical
3  Database
4  Operating systems/architecture/networks
5  Programming language design and implementation
6  Special applications
7  Artificial intelligence/independent study
8  Other
9  Special topics and readings

Thus, for example, the three basic programming courses are CS 110, CS 210 and CS 310.

CS 105: An Introduction to Computer Concepts
This course presents an overview of the role of computers in society— their application and misapplication, their capabilities and limitations. Applications may include artificial intelligence, medical, aerospace and business uses of computers. Computer hardware and associated technologies will be discussed. Computer programming will be taught from a non-Mathematical, problem-solving point of view, the objective being an understanding of the programming process, rather than the development of complex or extended computer programs. This survey course is not part of the computer science major sequence. Students planning to major in computer science should start with CS 110.

Note: No student will receive graduation credits for CS 105, if it is taken after successful completion of CS 110 or a higher level computer course. Students who enroll in CS 110 after taking CS 105 may receive fewer than the normal number of credits for CS 110.

Prerequisite: Math 115 or Math placement exam

CS 108: An Introduction to Python

An introduction to some of the basic issues in computation through exercises in Python programming. Students will write relatively simple programs in several application areas, e.g. mathematics, graphics and biology. This is a good course for those who have no programming experience and who want preparation for taking CS 110; students learn about using an editor, program design, implementation, and testing. This is also a good course for scientists who wish to learn a popular scientific scripting language.

CS 109: Computer Programming for Engineering

An introduction to computer programming for engineering students. This course is not intended for computer science majors. Credit will not be given for both CS 109 and CS 110.

Prerequisite: MATH 130
CS 110: An Introduction to Computing

An introduction to computer programming—the concepts involved in using a high-level language and the program development process. The goal of this course is proficiency in the design and implementation of programs of significant size and complexity. This course is quite demanding because of the length of the programming exercises assigned. This is the first course in the computer science major sequence.

Note: Credit toward an UMass Boston degree may be awarded for only one elementary programming course. Thus, no credit will be given for CS 110 to a student who has already completed (or is concurrently taking) a similar course. For instance, if you received the equivalent of CS 110 for transfer credit you cannot take CS 110 for credit here. Students who are barred from receiving credit for CS 110, but who are not fully prepared for CS 210L, should request permission to take CS 119 (below) as a means of remedying this deficiency.

Prerequisite: Math placement exam (ALEKS). Students who have passed Calculus I at UMass Boston or an equivalent calculus course elsewhere need not take the placement exam.

CS 119: Computer Language Supplement

This course addresses the dilemma of students who studied the equivalent of CS 110 in a language other than the one currently used in our CS 110 and CS 210L courses, but now need to learn the current language in order to take CS 210L. Such students can, with the permission of the department, register for CS 119 for two credits. They then make arrangements to attend a section of CS 110. They are required do all the assignments, examinations, etc., of regular CS 110 students. Thus, in virtually all respects, students who register for CS 119 are really CS 110 students, except that CS 119 meets no core curriculum requirement and does not count towards any Computer Science major requirement. Students receive two credits for this course.

Prerequisite: Permission of the CS Department.
CS 210L: Intermediate Computing with Data Structures

The design and implementation of computer programs in a high-level language, with emphasis on proper design principles and advanced programming concepts, including dynamic data structures and recursion. Efficient design, implementation and debugging techniques are stressed. The assignments are designed to introduce the student to a variety of topics in computing: data structures and ADTs, Lists, Stacks, Queues, Ordered Lists, Binary Trees, and searching and sorting techniques. The language of instruction is Java.

Prerequisite: CS 110 or permission of the CS Department

CS 240: Programming in C

C programming for programmers already knowledgeable in some high-level language (e.g., having taken one semester of introductory programming in Pascal, Java, Smalltalk, Lisp, etc.). C is presented as both a general-purpose and machine-level language. Topics covered include representation of integer and character data, bitwise operations, masking, memory allocation methods, pointers, dynamic data structures, file I/O, separate compilation, program development tools and use of debuggers.

Prerequisite: CS 110 or CS 115L plus CS210L as a co-requisite; or permission of the CS Department

CS 271L: Introduction to Cognitive Science

Cognitive Science is an interdisciplinary field fundamentally concerned with furthering our understanding of the development, underlying processes, and implementation of language, perception, problem-solving, learning, memory, and other intelligent capacities. This course offers an introduction to this science for all levels of undergraduate majors in psychology, computer science, biology and other related fields.
Prerequisite: None

CS 285L: Research Topics in Computer Issues: Ethics and Societal Impact

This course will teach and apply teamwork, research and presentation skills. Students will work in teams, each team investigating a broad topic in a selected area of computer application from an ethical and/or societal point of view.

Prerequisites: None

CS 310: Advanced Data Structures and Algorithms

A systematic study of the methods of structuring and manipulating data in computing. Abstract data types. The design and analysis of algorithms. Advanced techniques for program development and organization.

Prerequisites: CS 210L, CS 240, and MATH 140

CS 320L: Applied Discrete Mathematics

An introduction to the Mathematical structures and concepts used in computing: sets, Mathematical induction, ordered sets, Boolean algebras, predicate calculus, trees, relations and lattice theory, formal theory of languages and automata theory. Formal and informal theories and corresponding Mathematical proofs are taught.

Prerequisites: CS 110 or CS 115L and MATH 260; or permission of the instructor.

CS 341 Computer Organization and Architecture

Computer hardware concepts and hardware-level programming for C programmers. Topics are: digital logic circuits, computer organization of a microprocessor system (i.e., how the CPU, memory,
and I/O interface chips are interconnected to make a system), serial and parallel port interfacing, hardware programming in C and assembler, interrupt programming, device drivers. The course includes a hands-on lab meeting one hour per week.

**Prerequisite:** CS 240

**CS 410: An Introduction to Software Engineering**

This course covers all aspects of the software development process from initial specification to final validation of completed software design. Implementation methodologies are discussed in the context of a major team project, to be chosen according to student and instructor interest. Oral presentations by students are an important part of the course.

**Prerequisites:** CS 310, CS 320L, plus one other CS400 level course, or permission of the instructor.

**CS 420: An Introduction to the Theory of Computation**

Introduction to theoretical aspects of computing including models of computation, inherent limits on computation, and feasible computation. Topics studied will include definition of computable functions (recursive functions, functions computable by Turing machines, functions computable in a programming language), insolvability of the halting problem and related problems, the classes \( P \) and \( NP \), finite automata, and context-free grammars.

This course is required for all computer science majors who took CS 320L during fall 1989 or later. It may be used as a theoretical elective by other computer science majors.

**Prerequisites:** CS 320L

**CS 430: Database Management Systems**
Introduction to database systems, including database programming. The course covers relational algebra, SQL, object-relational systems, embedded programming, and basic transaction concepts. It covers database design, both entity-relationship modeling and normalization.

**Prerequisites:** CS 240 and CS 310 or permission of the instructor

**CS 436: Database Application Development**

A study of database applications, that is, software systems that solve a particular real-world problem and hold their data in a relational database. The systems under study will also have a realistic user interface. Students will work in small groups on a real-world project specified and implemented during the term. Topics include system specification from user needs, analysis of dataflow and workflow, object design, database design, and client-server techniques.

**Prerequisites:** CS310 and CS430

**CS 437: Database-Backed Web Sites and Web Services**

Today, much programming is web-based. Web-based programs serve up information from a Web site in a form that can be either read by a browser or processed by another program. This course introduces the student to the design and implementation of such web-based programs.

**Prerequisites:** CS 310 and CS430, or permission of the instructor

**CS 438: Applied Machine Learning**

This course presents the practical side of machine learning for applications, such as pattern recognition from images or building predictive classifiers. Topics will include linear models for regression, decision trees, rule based classification, support vector machines, Bayesian networks, and clustering. The emphasis of the course will be on the hands-on application of machine learning
to a variety of problems. This course does not assume any prior exposure to machine learning theory or practice.

**Prerequisites:** CS 310

**CS 444: An Introduction to Operating Systems** (Fall Semester Only)

Description of current operating systems, focusing on one or two in particular. Topics include defining the operating system as distinct from hardware on one side and software systems on the other, process concepts, memory management, CPU scheduling, device management, file systems, and network support. Note: this course was previously numbered as CS 440.

**Prerequisites:** CS 341 and CS 310

**CS 445: Real-Time Systems**

Covers the entire development process of a real-time system, from the requirements specification and design to implementation and testing, including real-time programming, low-level software issues, and hardware architectures. Emphasizes timing constraints and scheduling principles. Subtopics include the four layers of real-time system development: real-time design methodologies, real-time design programming language constructs, real-time kernels, and real-time hardware architectures.

**Prerequisites:** CS 341 and CS 310

**CS 446: Introduction to Internetworking** (Fall Semester Only)

The objective of this course is to provide a practical understanding of computer networks with emphasis on the Internet. The course starts with an overview of the Internet, its protocol layers, edge and core networks, access networks and physical media. The course then focuses on fundamental design and implementation concepts of the application, transport, and network layers of the Internet.
Prerequisites: CS 341 and CS 310. CS 444 is a co-requisite.

CS 447: Introduction to Multimedia Systems (Spring Semester Only)

Multimedia applications are ubiquitous, evidenced by their vast presence on the Internet. This course covers the basic fundamentals in the design and development of multimedia systems and applications. Three key elements are discussed at the introductory level: multimedia computing, multimedia databases, and multimedia networking. The topics include, but are not limited to, multimedia processing/compression/representations, multimedia content management and retrieval, and multimedia content streaming and distribution. Students are expected to work in groups to complete a semester-long project developing a system prototype.

Prerequisites: CS310 and CS341

CS449: Introduction to Computer Security

The course will provide an introduction to the fundamentals of computer security, and will cover both general theoretical aspects as well as applied methods of computer security. The course will address the general concepts of confidentiality, integrity and availability of digital information, and will focus on aspects such as: risks and vulnerabilities; models and policies for access control; program security - buffer overflow attacks, malware, viruses; browser security; authentication and authorization; encryption; and an overview of applied data, operating system and network security (with emphasis on Internet security). The course will also address the aspect of privacy, which is tightly related to security and is becoming increasingly important in today’s digital society.

Prerequisites: CS310 and CS341

CS 450: The Structure of Higher Level Languages
The syntax and semantics of higher-level languages are treated. Formal specifications of syntax and models of semantics will be used. Important topics include mechanisms for parameter passing, scoping, dynamic storage allocation and systems interfacing. Both compiled and interpreted languages will be used as examples. The current language of instruction is Scheme, a dialect of LISP, which is taught in the course.

Prerequisites: CS 31O and CS 320L

CS 451: Compilers I

Introduction to compiler organization and implementation, including formal specifications and algorithms for lexical and syntactic analysis, internal representation of the source program, semantic analysis, run-time environment issues and code generation. Students will write a compiler for a reasonably large subset of a contemporary language, targeted to a virtual machine.

Prerequisites: CS 31O and CS 42O or CS 622

CS 460: Graphics

Topics include segmentation, windows and viewports, clipping, hidden lines, geometric transforms and data structures for memory management and device independent graphics specifications. The course also considers raster graphics and the GKS and ACM Core. It covers both the practice of, and underlying Mathematical foundation for, interactive graphics programming. Students need good programming skills and a mastery of linear algebra.

Prerequisites: MATH 260 and CS 31O; or permission of the instructor

CS 470: An Introduction to Artificial Intelligence

An introduction to the main techniques of artificial intelligence: state-space search methods, semantic networks, theorem-proving and production rule systems. Important applications of these
techniques will be presented. Students will be expected to write programs exemplifying some of these techniques using the LISP language.

*Prerequisites: CS 310 and CS 320L or permission of the instructor*

**CS 478: Independent Study**

Work done by a student or group of students under faculty supervision on material not currently offered in a regularly scheduled course. Students wishing to undertake such work must first find a faculty member willing to supervise it; the work to be completed must be approved by the department’s chairperson. Students can receive between one and three credits for this course.

*Prerequisite: Permission of the instructor*

**CS 480: Special Topics**

An advanced Course offering intensive study of selected topics in computer science. Course content varies and will be announced prior to registration.

*Prerequisites: Permission of the department chair*

**CS 495: Practicum in Computer Science**

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. Students receive one credit for this course.

*Prerequisites: Permission of instructor*

**CS 498: Honors Thesis**
The design and execution of a significant research project under the guidance of a faculty advisor. Successful completion and oral defense of the Honors Thesis is necessary to qualify for Departmental Honors in Computer Science. Both the instructor and the project must be chosen and approved prior to the start of the course.

Prerequisites: Overall of GPA of 3.0 and GPA of 3.5 in the major; CS 310, CS 420, permission of the instructor and department chair. Students are encouraged to take CS 478 (Independent Study) the semester before 498 in order to undertake background research, formulate a successful project proposal, and present the proposal to the department. Forms for enrolling in CS 498 are available in the department’s office.

Part 2: Undergraduate Courses in Mathematics

The following Mathematics courses are required or can be used as an elective for some computer science majors.

MATH 140: Calculus I

Basic concepts of functions and limits, derivatives and their applications, definite and indefinite integrals with applications to geometrical and physical problems, discussion of algebraic and transcendental functions.

Note: A student who has received credit for MATH 135 may not take Math 140 for credit without the explicit permission of the department.

Prerequisite: Valid Math Placement Test (ALEKS) or passing MATH 130 in the previous term with a grade of B or better

MATH 141: Calculus II
Continuation of MATH 140. Topics include: transcendental functions, techniques of integration, applications of L'Hôpital’s rule, sequences, series.

Note: Because MATH 141 is the second part of a three-semester calculus sequence, it should be taken as soon as possible after MATH 140.

**Prerequisite:** MATH 140.

**MATH 260: Linear Algebra I**

Elementary theory of abstract vector spaces. Topics include: linear independence, bases, dimension, linear maps and matrices, determinants, orthogonality, eigenvalues and eigenvectors.

**Prerequisite:** MATH 140.

**MATH 345: Probability and Statistics I**

The mathematical laws of random phenomena, including discrete and continuous random variables, expectation and variance, and common probability distributions such as the binomial, Poisson, and normal. Basic principles of combinatorics are introduced to solve problems in discrete probability spaces. Topics also include basic ideas and techniques of statistical analysis.

**Prerequisite:** MATH 141 or permission of the instructor.

**MATH 346: Probability and Statistics II**

A statistics course for students with a firm mastery of calculus, emphasizing the mathematical and conceptual basis of statistics, with a view to understanding the proper application of standard methods. Thorough treatment of the Central Limit Theorem, theory of estimation, hypothesis testing and regression.
Prerequisite: MATH 345

MATH 360: Abstract Algebra I


Prerequisite: MATH 260 or permission of instructor

MATH 458: Theory of Numbers


Prerequisites: MATH 260 and CS 320L or equivalent

MATH 470: Mathematical Logic

Syntax and semantics of propositional and first order predicate logic. Axiomatic theories and completeness. Brief discussion of incompleteness results.

Prerequisite: MATH 360 or CS 320L

Part 3: Graduate Courses in Computer Science

The following courses are open to qualified undergraduates. The permission of the instructor is required for enrollment; satisfaction of the prerequisites is not sufficient. All graduate courses are 3 credits unless otherwise specified.

A note on courses: Please note that CS 310 or its equivalent is a general prerequisite for all graduate courses in computer science.
CS 612: Algorithms in Bioinformatics

The course will introduce students to bioinformatics - the area concerning the development and application of computational methods to address key problems in biology. It will introduce the students to a variety of methods and skills required to conduct research in this popular field. The emphasis of the course is algorithmic methods in structural bioinformatics with a focus on various computational methods to simulate, analyze, and model protein structure, dynamic, and function.

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 will read and critique papers and articles, evaluate and critique existing user interfaces and design interfaces of their own. Working in small groups, students will use either interface prototyping tools or conventional rapid prototyping systems to construct an experimental interface.

Prerequisites: CS 310, CS 320L or permission of the instructor.

CS 620: Theory of Computation


Prerequisites: CS 320L or permission of the instructor.

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 is 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.

**Prerequisites:** CS 320L and CS 450 or permission of the instructor.

**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.

**Prerequisites:** CS 320L or permission of the instructor.

**CS 630: Database Management Systems**


**Prerequisites:** CS240 and CS 310 or permission of the instructor

**CS 632: Database Models** (Fall Semester Only)

This course covers material required for the understanding and design of current state-of-the-art databases, relational and beyond; comparison of various relational query systems; functional and multivalued dependencies, normal forms, tableaux; object-relational and object-oriented databases.

**Prerequisites:** CS630
CS 634: Architecture of Database Systems


Prerequisite: CS 430/CS 630 or permission of the instructor.

CS 636: Database Application Development

Study of database applications that is software systems that solve a particular real-world problem and hold their data in a relational database. The systems under study also have a realistic user interface. Students work in small groups on a real-world project specified and implemented during the term. Topics include systems specification from user needs, analysis of data flow and work flow, object design, database design, client-server techniques and rapid prototyping systems.

Prerequisite: CS 615 and CS 630

CS 637: Database-Backed Web Sites

The design and implementation of database-backed web sites. 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.

Prerequisite: CS 430 or CS 630 and CS 451 or CS 651

CS 638: Applied Machine Learning
This course presents the practical side of machine learning for applications, such as pattern recognition from images or building predictive classifiers. Topics will include linear models for regression, decision trees, rule based classification, support vector machines, Bayesian networks, and clustering. The emphasis of the course will be on the hands-on application of machine learning to a variety of problems. This course does not assume any prior exposure to machine learning theory or practice.

**CS 639: Semi-structured Data and XML Documents on the Web**

A study of the international standard eXtensible Markup Language (XML) and related semi-structured data technologies for application to web programming. Special attention will be given to combining data from multiple sites and on-line data bases, and to the transformation, display, and extraction of data from XML documents for data exchange, resource discovery and the building of interactive web applications.

**Prerequisite:** CS 451/CS 651 or CS 636 or CS 637 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 trade-offs in these designs. Note: this course was previously numbered CS 644.

**Prerequisites:** CS 310 or permission of the instructor

**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 considerations, device drivers. UNIX as a major example. Note: this course was previously numbered CS 640.

**Prerequisites:** CS 641 or permission of the instructor.

**CS 646: Computer Communications Networks**


**Prerequisite:** CS 444 or permission of the instructor

**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 the development of a prototype.

**Prerequisite:** CS 446 or CS 646

**CS 648: Wireless Networks & Mobile Computing**

Wireless communications, wireless networking, mobility management technologies and protocols for wireless LANs and WANs are surveyed. Selected mobile computing models and mobile application
development environments are evaluated. A wireless networks laboratory provides a realistic mobile/wireless computing environment. Students are expected to form groups to complete a significant semester project involving an investigation and the development of a prototype.

**Prerequisite:** CS 446 or CS 646

CS 651: Compilers I

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.

**Prerequisites:** CS 310; CS 420 or CS 622

CS 662: Document Preparation and Text Processing Systems

An applied course in contemporary document preparation systems. This course will vary 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 will 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 will be studied and criticized, where possible by a study of their source code.

**Prerequisites:** At least one course at the level of CS444 or above

CS 664: Image Processing

Techniques of digital image processing. Topics include processing in the image and spatial frequency domains; Fourier and other transforms, wavelets, continuous and discrete convolution and
filtering; gray-level transforms, feature identification, image encoding, image enhancement; applications to models of human and machine vision.

**Prerequisites:** *Calculus III and one semester of linear algebra, or permission of the instructor*

**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 310 and CS 320L*

**CS 671: Machine Learning**

Machine Learning is a foundational discipline for data mining and artificial intelligence which explores the limits and capacities of automated learning of abstract concepts. The course will focus on the probably approximately correct (PAC) learning model and will cover topics like the Vapnik-Chervonenkis dimension, the weak and strong learning paradigms, inherent unpredictability, reducibility in PAC learning, and learning finite automata.

**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, databases, etc.
Prerequisites: permission of the instructor

CS 674: Natural Language Processing (NLP)

The course provides the basic principles 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 or permission of the instructor

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. Their final project is the development of their own computer vision program that solves a given problem.

Prerequisites: CS 310, CS 320L or permission of the instructor

Part 4: Degrees and Certificates

No course taken to satisfy a requirement for a degree or certificate may be taken Pass/Fail.

The Bachelor of Science Degree
The department requirements for a Bachelor of Science degree with a major in computer science are given below.

Degree-seeking students must also satisfy the other requirements of the college such as the Core requirements, the English requirement, and the Writing Proficiency Examination requirement. These are not listed here.

**BS in Computer Science**

The current department requirements are listed below:

1. *CS 110; CS 210L; CS 240; CS 285L; CS 310; CS 320L; CS 341; CS 410*(this will be the capstone course for the major); CS 420; CS 444; CS 450 and CS 451 or CS 651.*

   (Students who declared their major prior to 01/27/03 may replace CS 285L with an additional computer science elective).

2.  *MATH 140; MATH 141; MATH 260 and MATH 345.*

3.  *Physics 113; Physics 181; Physics 114 and Physics 182.*

4.  One science elective. Information on allowable electives may be obtained in the department office.

5.  Two computer science electives chosen from: CS 260; CS 430; CS 437; CS 445; CS 446; CS 460; CS 470; CS 615; CS 620; CS 622; CS 624; CS 630; CS 634; CS 636; CS 637; CS 639; CS 641; CS 644; CS 646; CS 647; CS 648; CS 664; CS 670; CS 672; CS 674; CS 675. With prior permission it may be possible to take an independent study course in place of one of the above.

6.  At least four 300; 400; or 600 level computer science or mathematics courses must be taken at UMass Boston. This limits the number of transfer courses that can be applied towards a degree.

7.  A major must maintain a C average (2.0 GPA) in all of the above. Only courses taken at UMass Boston are averaged.
Sample Program for the Major

We present below a possible sequence of mathematics and computer science courses leading to satisfaction of the department requirements in four years. We do not show courses in other departments.

This schedule is only an example; not a prescription or a recommendation. Programs of study will vary depending on students’ interests and abilities. Stronger students may wish to do more than meet the minimal requirements.

1. CS 110, MATH 140
2. CS 210L, MATH 141, Physics 113, Physics 181
3. CS 240, MATH 260, Physics 114, Physics 182
4. CS 310, CS 341, CS 320L
5. CS 285L, CS 450, MATH 345
6. CS 420, CS 451, science elective
7. CS 444, computer science elective
8. CS 410, computer science elective

Bachelor of Arts in Computer Science

The department requirements for a Bachelor of Arts degree with a major in computer science are given below.

Degree-seeking students must also satisfy the other requirements of the college such as the Core requirements, the English requirement, and the Writing Proficiency Examination requirement. These are not listed here.

Some students may find that their scores on the department’s placement test indicate that they are not yet qualified to take CS 110 or MATH 140. Such students will begin their course work with
MATH 130 (Pre-calculus) or possibly MATH 115 (College Algebra). Credits earned in those courses count toward the degree, but not toward the major.

BA in Computer Science

The current department requirements are listed below:

1. **CS 110; CS 210L; CS 240; CS 310; CS 320L; CS 341; CS 420; and CS 450.**

2. **MATH 140; MATH 141; and MATH 260.**

3. **Two applied computer science electives chosen from the following list. (One of these choices must be CS 444; or CS 451 which will be the capstone course for the major.) CS 410; CS 430; CS 437; CS 444; CS 445; CS 446; CS 451; CS 460; CS 470; CS 615; CS 630; CS 634; CS 636; CS 637; CS 639; CS 641; CS 644; CS 646; CS 647; CS 648; CS 651; CS 664; CS 670; CS 672; CS 674; CS 675.** With prior permission it may be possible to take an independent study course in place of one of the above.

4. **One theoretical elective from among: MATH 345; MATH 346; MATH 360; MATH 425; MATH 458; MATH 470; CS 620; CS 622; and CS 624**

5. **One additional elective; either theoretical; applied or CS 260.**

6. **At least four 300; 400; or 600 level computer science or mathematics courses must be taken at UMass Boston.** This limits the number of transfer courses that can be applied towards a degree.

7. **A major must maintain a C average (2.0 GPA) in all of the above.** Only courses taken at UMass Boston are averaged.

Sample Program for the Major

We present below a possible sequence of mathematics and computer science courses leading to satisfaction of the department requirements in four years. We do not show courses in other departments.
This schedule is only an example, not a prescription or a recommendation. Programs of study will vary depending on students’ interests and abilities. Stronger students may wish to do more than meet the minimal requirements.

1. MA 130 (if necessary)
2. CS 110, MATH 140
3. CS 210L, MATH 141
4. CS 240, MATH 260
5. CS 310, CS 320L
6. CS 341, CS 420
7. CS 450, CS 444 or CS 451, and 1 theoretical elective
8. 1 applied elective, and 1 additional elective

**Honors program in Computer Science**

The department offers an Honors option for those who qualify. To graduate with Department Honors in Computer Science, a Computer Science major must satisfy the following requirements:

1. Maintain a GPA of 3.0 or above in his/her University studies.
2. Maintain a GPA of 3.5 or above in his/her major courses.
3. Successfully complete CS 498, including an oral defense of the Honors Thesis.

For the defense of the Honors Thesis, the faculty supervisor will form a committee consisting of himself/herself and two colleagues whose backgrounds are relevant to the project. The membership of the committee is subject to approval by the undergraduate program director or chair. All members of the committee must be present for the defense of the thesis and must deem it worthy of honors recognition. Whether or not the defense is successful, credit for CS 498 will be granted based on a passing grade by the faculty advisor. Honors theses will be published by the department as Technical Reports, so they will be accessible from both inside and outside the department.
CS Undergraduate Certificate

The Computer Science Undergraduate Certificate is available for matriculated and non-matriculated students. All students who wish to start the Certificate Program must fill out a form available at the CS main office, S-3-132, or here.


The form must be signed and returned to One Stop in the Campus Center.

For matriculated students, the Certificate is in lieu of a Minor or Program of Study in Computer Science. Students who want to add the Certificate in Computer Science need to complete a Major Declaration form. Non-matriculated students in the Certificate program are non-degree students and would need to register for classes with the Non-degree Registration form at One-Stop, but not apply to UMB for admission.

http://www.umb.edu/registrar/registration_courses/registered_courses/non_degree_students/

Students whose preparation in Computer Science is insufficient for admission to the Master’s program should consider seeking the Undergraduate Certificate. Because of the prerequisite structure of the courses it may take two years to complete the Certificate (the exact time will depend on the student’s background), and it may not be possible to do course work full-time. In particular, the listed mathematics requirements begin with calculus. However, many students may need a review of high school algebra. On the other hand, students who enter this program with previous mathematics or computer programming experience will be given appropriate credit.

The Undergraduate Certificate is intended for students who are mainly interested in computer applications. The requirements are:

- All five of CS110, CS210, Math140, CS240, CS310
- One of CS260, CS341, or Computer Science elective
• One application elective

The CS Dept. does not offer winter courses; we only offer classes during Fall, Spring, and Summer semesters. We currently do not offer online courses. The Undergrad Certificate typically requires 4 semesters due to prerequisites. We do not allow students to take requisites out of order. The schedule is as follows:

1st semester – CS 110 & Math 140
2nd semester – CS 210 & CS 240
3rd semester – CS 310 & CS 341 (or CS elective)
4th semester – applications elective

Summer courses only include CS 110, CS 210, CS 240, [CS 320L not required] & CS 310.

Applications elective: The applications requirement may be satisfied by any Computer Science course numbered higher than (not including) CS320L, or by a course taught outside of the department of Computer Science in computer applications to another discipline. Students wishing to use such a course as the applications elective for a Certificate must have their choice approved in advance by the Undergraduate Program Director and must meet the prerequisites for the course set by the department in which it is offered.

Interested students who would like to speak to an advisor about our Undergrad Certificate should contact the Undergraduate Program Director, Ming Ouyang, at ming.ouyang@umb.edu

Non-matriculated students need to contact Computer Science Administrative Assistant, Allison Christiansen allison.christiansen@umb.edu, when they feel they’ve fulfilled all requirements. She will print off a transcript and send it to Ming Ouyang for evaluation and then send the student the Certificate. Matriculated students will be evaluated when they declare their intent to graduate and receive their certificate upon graduation. Students who wish a more thorough training in computer science should consider doing a double major.
Placement Test and Registration

To enroll in any mathematics course below Calculus II (MATH 141) you must either perform at the required level on the Math Placement Test (ALEKS) or have taken the appropriate prerequisites at UMass Boston. ALEKS is an online, adaptive assessment and learning system that is tailored to your ability level. ALEKS includes a six-week personalized learning module to allow you to be well prepared for the class into which you placed, or to help you review and re-test into a higher level class. In most cases, ALEKS can be taken off-campus or anywhere you have internet access.

More information can be found here

http://www.umb.edu/academics/vpass/uac/testing_services/math

The specific requirements for each course are listed in the table below. In particular, note the following:

1. **New students are required to take the Math Placement Test (ALEKS) as part of their orientation**

2. **Permission to enroll in MATH 130 is by Math Placement Test (ALEKS) result only.**
   
   Completion of MATH 115 is not sufficient—a student who has passed this course must still receive the appropriate score on the Math Placement Test to enroll in MATH 130.

3. **Permission to enroll in MATH 140 is by Math Placement Test (ALEKS) or passing MATH 130 in the previous term with a grade of B or better**

4. **Permission to enroll in CS 110 is by Math Placement Test (ALEKS) or successful completion of a course in calculus**

5. **CS 105 is not sufficient preparation for any other computer course.** To take CS 110, a student with too low a placement score may need to take MATH 115 in order to improve his or her subsequent placement exam performance to the level required for MATH 130 and CS 110.
If you have already learned some computer programming in high school, in another college, or on the job, and it is possible, please bring with you evidence of your computer experience. Examples are: copies of transcripts, certificates, grade reports, programs you have written, etc. If you have taken and scored high enough on the Computer Science A or A/B Advance Placement exam, you will be given credit for the equivalent computer science courses here.

**Advanced Placement Course Credits**

The College Board Advanced Placement Exams will receive credit as follows:

- **Computer Science Test A**  
  Score 3 or better  
  CS110

- **Computer Science Test AB**  
  Score 3 or better  
  CS110 and CS210

**Computer Resources**

Students in our introductory courses may use the Computing Services PC Labs in the Healey Library. We arrange to have all software applications needed for these CS courses installed on those systems. We also usually provide students URL’s for free software that they can download from the Internet and install on their own PCs to do their homework assignments for these courses at home. The department provides access to a turn-in server for students to upload their homework assignments for grading.

The department operates a large network of UNIX/LINUX servers for CS students in upper division courses. The UNIX servers are accessible 24 hours a day via the internet. We provide a UNIX/PC lab for local on campus access to UNIX servers and use with specific PC software required for some courses. The local lab is accessible 24 hours a day based on an access control list. We provide a Weblab, based on 18 PC’s, that is available for teaching, presentations, or research based on instructors’ requirements for their courses. The Weblab may be accessible 24 hours a day based on a key coded door lock if an instructor has made arrangements for this.
We also maintain specialized laboratories for teaching and research in machine organization and architecture, computer networks, human vision, and software engineering. Computing Services operates an Adaptive Computing Lab for computer users with special needs, such as impaired vision or a physical handicap. A description of Computing Services resources is provided on their website http://www.umb.edu/it. The University library offers an on-line catalog service, accessible from the department’s and Computing Service’s computers, as well as at terminals in the library itself. Other libraries’ catalogs are also reachable over the Internet.

Work/Study

Many UMass Boston students work while studying; some find jobs on campus related to their major. In addition to the money these jobs pay, students gain useful, practical experience. Acquiring relevant practical experience through industrial internships or cooperative education is strongly recommended.

The Computer Science Department hires students to grade papers and to assist in running its UNIX machines. The computer center hires some students as part-time computer operators and consultants. The operators help to run the computers and the consultants help to solve the problems of novice computer users in the Healey Labs. Academic Support Services hires students as tutors for computer and other mathematics courses.

Contact Information

Ming Ouyang
Assistant Professor, Undergraduate Program Director
Department of Computer Science
College of Science & Mathematics
Ming.Ouyang@umb.edu
S-3-070