Microcomputers are used widely in all areas of modern life. For this reason it is important for all students to understand how computers work and how computers can be used as a problem-solving tool. The focus of this course is on computer applications. The course stresses the ways in which computers can help you solve problems efficiently and effectively. The course provides a broad introduction to hardware, software, and mathematical aspects of computers. Four application areas are discussed: Internet tools (including Web page design), word processing, spreadsheets, and databases. Weekly lab assignments are an integral part of the course, and it is expected that students have access to their own computing equipment. There are optional lab times set up for students who do not have the proper equipment or software available to them. This course is a "Foundations" course for the Information Technology minor. Students who are more interested in computer programming should take a course such as CMPSCI 191P or CMPSCI 121. Prerequisites: reasonable high school math skills. Typing ability is also an important asset for the course. Some previous computer experience, while not absolutely required, will prove helpful. Not for CMPSCI majors. 3 credits.

Introduction to Programming

The Internet has transformed computers from machines that calculate to machines that communicate. This introduction to computer programming with Python emphasizes multimedia (graphics and sound) applications that are relevant for Web designers, graphic artists, and anyone who just wants to have more fun with their computer. Students will explore basic concepts in computer science and computer programming by manipulating digital images and sound files. No prior programming experience is needed. Not for CMPSCI majors. 3 credits.

Introduction to Problem Solving with the Internet (R2)

The Internet is a goldmine of information and software resources for those who know how to plug in and navigate it. Originally designed by computer scientists for computer scientists, the net is now a driving force behind life in the information age and a new global economy. This course will provide non-CMPSCI majors with timely skills needed to tap the net as well as an introduction to basic networking, client-side web programming in HTML, CSS, and Javascript, and server-side programming in Python. In addition to static and dynamic web page and web site design and implementation, we will cover strategies for finding information, managing e-mail, and ensuring privacy. We will survey current social, technical, and political topics that are relevant to the Internet such as spam and malware, net neutrality, censorship, copyright laws, and public key cryptography. Prerequisites: some hands-on experience with PCs or MACs or UNIX (programming experience is NOT required). Not for CMPSCI majors. 3 credits.

Introduction to Problem Solving with Computers (R2)

CMPSCI 121 provides an introduction to problem solving and computer programming using the programming language Java; it also provides an integrated introduction to some of the wonderful innovations to modern science and indeed modern life that can be attributed to computer science. The course teaches how real-world problems can be solved computationally using the object-oriented metaphor that underlies Java. Concepts and techniques covered include data types, expressions, objects, methods, top-down program design, program testing and debugging, state representation, interactive programs, data abstraction, conditionals, iteration, interfaces, inheritance, arrays, graphics, and GUIs. No previous programming experience required. A companion introduction to programming class, CMPSCI 191P is also offered. If you are fairly sure you only want to do just one programming class, take that course; if you think it likely that you will do more than one programming course, take 121. Use of computer is required. Prerequisite: R1. 4 credits.

Programming with Data Structures (R2)

The course introduces and develops methods for designing and implementing abstract data types using the Java programming language. The main focus is on how to build and encapsulate data objects and their associated operations. Specific topics include linked structures, recursive structures and algorithms, binary trees, balanced trees, and hash tables. These topics are fundamental to programming and are essential to other courses in computer science. There will be weekly assignments and assignments in discussion sections consisting of programming and written exercises. There will also be several exams. Prerequisites: CMPSCI 121 (or equivalent Java experience) and Basic Math Skills (R1). Basic Java language concepts are introduced quickly; if unsure of background, contact instructor. 4 credits.
Mathematical techniques useful in the study of computing and information processing. The mathematical method of definition and proof. Sets, functions, and relations. Combinatorics, probability and probabilistic reasoning. Graphs and trees as models of data and of computational processes. Prerequisite: R1 math skills recommended. Not intended for Computer Science majors – students interested in a majors-level treatment of this material should see CMPSCI 240 and 250, or MATH 455. 3 credits.

**CMPSCI 190DM**  
*A Mathematical Foundation for Informatics*  
*David Barrington*  
Mathematical techniques useful in the study of computing and information processing. The mathematical method of definition and proof. Sets, functions, and relations. Combinatorics, probability and probabilistic reasoning. Graphs and trees as models of data and of computational processes. Prerequisite: R1 math skills recommended. Not intended for Computer Science majors – students interested in a majors-level treatment of this material should see CMPSCI 240 and 250, or MATH 455. 3 credits.

**CMPSCI 191A**  
*Computer Science Majors RAP Seminar*  
*David Barrington, Robert Moll*  
Conversations with members of the UMass CMPSCI community, to explore issues in and subareas of computer science. Open only to first-year students in the CMPSCI RAP residential program. Mandatory pass/fail. This course does not count toward requirements for the CMPSCI major or minor. 1 credit.

**CMPSCI 197B**  
*Special Topics - Advanced Assignments for CMPSCI 121*  
*Robert Moll*  
This add on to CMPSCI 121 consists entirely of five additional more advanced programs that students write to extend the coding experience available in the base 121 class. The course is intended primarily for students in 121 who have some previous programming experience. Must be enrolled in CMPSCI 121 concurrently. 1 credit.

**CMPSCI 197C**  
*Special Topics - Programming in C*  
*Nicolas Scarcci*  
A brief introduction to the C programming language for students with a good working knowledge of Java and data structures. This course is good preparation for CMPSCI 230 and courses that use C and C++. Prerequisites: CMPSCI 121 and 187 or permission of instructor. Runs for 6 weeks. This course is for CMPSCI minors and majors only, but it does not count towards either degree. 1 credit.

**CMPSCI 197U**  
*Special Topics - A Hands-on Introduction to UNIX*  
*Jarred Devaughn-Brown*  
This course offers a 6-week introduction to working with Unix, and it is intended to help students work with tools commonly used in CS courses. The class is comprised of both discussion and hands-on exercises in the EdLab. Topics covered include working with the command line, installing and maintaining the OS and software packages, version control systems, compiling programs, and more. No previous experience with Unix is required. This course is for CS minors and majors only, but it does not count towards either degree. 1 credit.

**CMPSCI 220**  
*Programming Methodology*  
*Jack Wileden, John Altidor*  
Development of individual skills necessary for designing, implementing, testing and modifying larger programs, including: use of integrated design environments, design strategies and patterns, testing, working with large code bases and libraries, code refactoring, and use of debuggers and tools for version control. There will be significant programming and a mid-term and final examination. Prerequisite: CMPSCI 187 or ECE 242. 4 credits.

**CMPSCI 230**  
*Computer Systems Principles*  
*Timothy Richards*  
Large-scale software systems like Google - deployed over a world-wide network of hundreds of thousands of computers - have become a part of our lives. These are systems success stories - they are reliable, available (“up” nearly all the time), handle an unbelievable amount of load from users around the world, yet provide virtually instantaneous results. On the other hand, many computer systems don’t perform nearly as well as Google - hence the now-cliché “the system is down.” In this class, we study the scientific principles behind the construction of high-performance, scalable systems. The course begins with a discussion of the relevant features of modern architectures, and moves up the stack from there to operating system services such as programming language runtime systems, concurrency and synchronization, with a focus on key operating system features, I/O and networking, and distributed services. This course can be used as a “core” course for students in the CMPSCI minor. Prerequisites: CMPSCI 187 or ECE 242. 4 credits.
Development of mathematical reasoning skills for problems that involve uncertainty. Each concept will be illustrated by real-world examples and demonstrated through in-class and homework exercises, some of which will involve Java programming. Counting and probability -- basic counting problems, probability definitions, mean, variance, binomial distribution, Markov and Chebyshev bounds. Probabilistic reasoning -- conditional probability and odds, Bayes' Law, Naive Bayes classifiers, Monte Carlo simulation. Markov chains, Markov decision processes, classical game theory, introduction to information theory. This is a core course for the new CMPSCI curriculum and may be used as a math elective for the old curriculum. Prerequisites: CMPSCI 187 (or ECE 242) and MATH 132 or consent of instructor. 4 credits.

**CMPSCI 250**  
**Introduction to Computation**  
David Barrington  

**CMPSCI 290NW**  
**A Networked World**  
James Kurose  
The course will cover the technical foundations of today's communication networks, particularly the Internet. It will also address key social, policy, economic and legal aspects of these networks, their use (and abuse), and their regulation. This course covers computer science topics, but all material will be presented in a way that is accessible to an educated audience with or without a strong technical background. Not intended for Computer Science majors – students interested in a majors-level treatment of this material should see CMPSCI 453. 3 credits.

**CMPSCI 305**  
**Social Issues in Computing**  
Michelle Trim  
Designed to satisfy the Junior Year writing requirement, CMPSCI 305 introduces the student to technical writing and editing, scientific journalism and the social essay. The course combines practical, scientific writing as found in industry and business with explorative essays that focus attention upon the technological and humanistic concerns inherent in society. Ten written assignments—two longer papers, eight shorter ones and one oral presentation. Prerequisite: ENGLWP 112 or CW. 3 credits.

**CMPSCI 311**  
**Introduction to Algorithms**  
Philip Thomas  
This course will introduce you to algorithms in a variety of areas of interest, such as sorting, searching, string-processing, and graph algorithms. You will learn to study the performance of various algorithms within a formal, mathematical framework. You will also learn how to design very efficient algorithms for many kinds of problems. There will be one or more programming assignments as well to help you relate the empirical performance of an algorithm to theoretical predictions. Mathematical experience (as provided by CMPSCI 250) is required. You should also be able to program in Java, C, or some other closely related language. Prerequisite: CMPSCI 250 or MATH 455. 4 credits.

**CMPSCI 320**  
**Introduction to Software Engineering**  
Leon Osterweil  
In this course, students learn and gain practical experience with software engineering principles and techniques. The practical experience centers on a semester-long team project in which a software development project is carried through all the stages of the software life cycle. Topics in this course include requirements analysis, specification, design, abstraction, programming style, testing, maintenance, communication, teamwork, and software project management. Particular emphasis is placed on communication and negotiation skills and on designing and developing maintainable software. Use of computer required. Several written assignments, in-class presentations, exams, and a term project. Prerequisite: CMPSCI 220. 4 credits.

**CMPSCI 325**  
**Usability**  
Eva Hudlicka  
In this course we examine the important problems in Usability, Human Computer Interaction, User Interfaces, and Human Centered Computing. We will examine elements of HCI history, understanding human capabilities, HCI design, several methods for prototyping user interfaces, and new applications and paradigms in human computer interaction. This is not a course in how to make dialog boxes, but rather a much more general approach to interacting with human beings and evaluating designs. Some elementary programming (or user interface prototyping tools) may be required, but people without prior programming experience should feel right at home in this class. The course is available to any undergraduate student, not just computer scientists. IT-minor students are especially encouraged to participate. Several group projects and exams will be required. No prerequisites. 3 credits.
The World Wide Web was proposed originally as a collection of static documents inter-connected by hyperlinks. Today, the web has grown into a rich platform, built on a variety of protocols, standards, and programming languages, that aims to replace many of the services traditionally provided by a desktop operating system. Topics will include: producing dynamic content using a server-based language, content serving databases and XML documents, session state management, multi-tier web-based architectures, web security, and core technologies including HTTP, HTML5, CSS, JavaScript, and SQL will be emphasized. This course will also study concepts and technologies including AJAX, social networking, mashups, JavaScript libraries (e.g., jQuery), and web security. This course is hands-on and project-based; students will construct a substantial dynamic web application based on the concepts, technologies, and techniques presented during lecture. Prerequisites: CMPSCI 187 or ECE 242. 3 credits.

In this course we examine the important problems in operating system design and implementation. The operating system provides a well-known, convenient, and efficient interface between user programs and the bare hardware of the computer on which they run. The operating system is responsible for allowing resources (e.g., disks, networks, and processors) to be shared, providing common services needed by many different programs (e.g., file service, the ability to start or stop processes, and access to the printer), and protecting individual programs from one another. The course will start with a brief historical perspective of the evolution of operating systems over the last fifty years, and then cover the major components of most operating systems. This discussion will cover the tradeoffs that can be made between performance and functionality during the design and implementation of an operating system. Particular emphasis will be given to three major OS subsystems: process management (processes, threads, CPU scheduling, synchronization, and deadlock), memory management (segmentation, paging, swapping), file systems, and operating system support for distributed systems. Prerequisites: CMPSCI 230 with a grade of 'C' or better. 4 credits.

The course explores key concepts underlying intelligent systems, which are increasingly deployed in consumer products and online services. Topics include problem solving, state-space representation, heuristic search techniques, game playing, knowledge representation, logical reasoning, automated planning, reasoning under uncertainty, decision theory and machine learning. We will examine how these concepts are applied in the context of several applications. Prerequisites: (CMPSCI 220 or CMPSCI 230) and CMPSCI 240. 3 credits.

Computing has become data-driven, and databases are now at the heart of commercial applications. The purpose of this course is to provide a comprehensive introduction to the use of data management systems within the context of various applications. Some of the covered topics include application-driven database design, schema refinement, implementation of basic transactions, data on the web, and data visualization. The class will alternate between lecture and practice, and the students will experience the covered topics through a semester-long collaborative mini-project. This course counts as a CS Elective toward the CMPSCI major (BA/BS). Students who have taken CMPSCI 445 are not eligible to take this course. 3 credits.

In recent years, the ability to continuously monitor activities, health, and lifestyles of individuals using sensor technologies has reached unprecedented levels. The typical smartphone comes equipped with a plethora of sensors for monitoring activity, speech patterns, social interactions, and location. In addition, mobile accessories such as wearable wristbands and chestbands now enable routine and continuous monitoring of a host of physiological signals (e.g., heart rate, respiratory rate, skin Conductance, and others.). In conjunction, these sensors can enable higher-order inferences about more complex human activities/behavioral states (e.g., activity patterns, stress, sleep, etc.). Such ubiquitous sensing in daily life, referred to as mobile health sensing and monitoring, promises to revolutionize our understanding of human activities and health conditions. This course is an introduction to personal health sensing and monitoring through mobile phones and on-body sensors and addresses several aspects including mobile devices and applications for health, sensor data quality and reliability challenges, inference of key health assessments from sensor data including such as activity patterns, sleep patterns, or stress, sensor data visualization and feedback, and practical considerations such as battery lifetime. This course is approved as a CS elective for CMPSCI major. Prerequisite: CMPSCI 187 (or equivalent). 3 credits.
How does the computer actually work? In this course we peel away the layers of abstraction and look at how switches become logic circuits, how logic circuits do math, and how programs really execute. We will wire up some simple examples of logic, move on to programming a minimalist simulation of a computer, and gradually build up to the point where we can appreciate how a C program is translated into machine code and what really happens when it executes. We will also see the impact of hidden acceleration mechanisms like caches, pipelines, and branch predictors. This course is approved as a CS elective for CMPSCI major. Prerequisite: CmpSci 220 or equivalent. 3 credits.

**CMPSCI 391IB**  
**Seminar - Inside the Box: How Computers Really Work**  
Charles Weems  
Honors Colloq

This course explores the basic problems in the translation of programming languages focusing on theory and common implementation techniques for compiling traditional block structured programming languages to produce assembly or object code for typical machines. The course involves a substantial laboratory project in which the student constructs a working compiler for a considerable subset of a realistic programming language, within a provided skeleton. The lectures are augmented by a discussion section that covers details of the programming language used to build the compiler, the operating system, the source language, and various tools. Use of computer required. Text: Engineering a Compiler, Cooper and Torczon. Prerequisites: CMPSCI 250 (or MATH 455) and CMPSCI 377. 3 credits.

**CMPSCI 410**  
**Compiler Techniques**  
J. Eliot Moss

This course explores the basic problems in the translation of programming languages focusing on theory and common implementation techniques for compiling traditional block structured programming languages to produce assembly or object code for typical machines. The course involves a substantial laboratory project in which the student constructs a working compiler for a considerable subset of a realistic programming language, within a provided skeleton. The lectures are augmented by a discussion section that covers details of the programming language used to build the compiler, the operating system, the source language, and various tools. Use of computer required. Text: Engineering a Compiler, Cooper and Torczon. Prerequisites: CMPSCI 250 (or MATH 455) and CMPSCI 377. 3 credits.

**CMPSCI 453**  
**Computer Networks**  
James Kurose  
Honors Colloq

This course provides an introduction to fundamental concepts in computer networks, including their design and implementation. Topics covered include the Web and other applications, transport protocols (providing reliability and congestion control), routing, and link access. Special attention is also paid to wireless networks and security. Homework assignments involve programming and written tasks. Prerequisites: Experience programming; CMPSCI 230 (or CMPSCI 377) and CMPSCI 240 (or STATS 515). 3 credits.

**CMPSCI 460**  
**Introduction to Computer and Network Security**  
Christopher Misra

This course provides an introduction to the principles and practice of system and network security with a focus on both fundamentals and practical information. The three key topics of this course are cryptography, system and network security, and protecting information technology resources. Subtopics include ciphers, key exchange, security services (integrity, availability, confidentiality, etc.), system, network, and web based security attacks, vulnerabilities, exploits, countermeasures and responses. Students will complete several lab assignments experimenting with security countermeasures. Grades will be determined by class participation, lab work, homework, quizzes and exams. Prerequisites include CMPSCI 377 and a familiarity with Unix. 3 credits.

**CMPSCI 474**  
**Image Synthesis**  
**DELETED**

This course provides a broad overview of the theory and practice of image synthesis. Topics include photorealistic rendering, global illumination techniques, Monte Carlo ray tracing, texture synthesis, image editing and compositing. Previous experience with introductory computer graphics (such as CMPSCI 473) is preferred. Prerequisites are: CMPSCI 187 or equivalent, MATH 235. 3 credits.

**CMPSCI 503**  
**Embedded Computing Systems**  
Roderic Grupen

This course introduces tools for embedded computational applications in a class focused on team-oriented design applications. Your team will build an integrated robot to perform a challenge task and will compete against other teams for the best system. The course is heavily project-oriented (with a required lab) and discussions will include topics such as; (1) mechanisms, sensors, actuators and feedback systems, (2) analog and digital circuits, power amplifiers, signal processing, operational amplifiers, multiplexing, (3) I/O - A/D, D/A, and latching, serial and parallel interfaces, (4) signal processing/conditioning and (5) an introduction to real-time programming. 3 credits.

**CMPSCI 513**  
**Logic In Computer Science**  
Neil Immerman

Rigorous introduction to mathematical logic from an algorithmic perspective. Topics include: Propositional logic: Horn clause satisfiability and SAT solvers; First Order Logic: soundness and completeness of resolution, compactness theorem. We will use the Coq theorem prover and Datalog. Prerequisites: CMPSCI 250 and CMPSCI 311. 3 credits.
Software has become ubiquitous in our society. It controls life-critical applications, such as air traffic control and medical devices, and is of central importance in telecommunication and electronic commerce. In this course, we will examine state-of-the-art practices for software testing and analysis to verify software quality. We will initially look at techniques for testing and analyzing sequential programs, and then examine the complexity that arises from distributed programs. The students will be required to complete regular homework assignments and exams, and carry out a group research project extending techniques described in class and/or applying them to new domains. No required text. Papers from the open literature will be assigned and made available. Prerequisites: CMPSCI 320, Introduction to Software Engineering (or equivalent course). 3 credits.

The purpose of this course is to provide students with practical experience in the management of software development projects. Students in this course will gain this experience by serving as software development team technical managers for teams of software engineering students in CMPSCI 320. As project managers, the students in CMPSCI 529 will be responsible for: supervising and managing the work of teams of CMPSCI 320 students; interfacing with the other CMPSCI 529 students managing other teams in the course; interfacing with the course instructor, course TA, and course customer. CMPSCI 529 students will be assigned readings in software engineering project management to provide a theoretical basis for their work in this course. But the majority of work in the course will be related to the actual management of assigned development teams. As team managers, CMPSCI 529 students will set goals and schedules for their teams, track and report team progress, negotiate with leaders of other teams and the course customer, and evaluate the work of members of their teams. CMPSCI 529 course assignments may include: written team goals, plans and schedules; periodic reports on team progress; documentation of agreements reached with other team leaders and customers; evaluations of the applicability of theoretical papers to the work of this course. This course will meet at the same times and places as CMPSCI 320. Additional meetings with team members and other students in CMPSCI 529 are also expected to be arranged by mutual agreement. An additional one hour weekly meeting of all of the students in CMPSCI 529 is required. Enrollment in this course is only by permission of the instructor, and is restricted to students who have previously taken CMPSCI 320, and received a grade of A or A-. 3 credits.

Introduction to Natural Language Processing
Andrew McCallum
Natural Language Processing addresses fundamental questions at the intersection of human languages and computer science. How can computers acquire, comprehend and produce English? How can computational methods give us insight into observed human language phenomena? How can you get a job at Google? In this interdisciplinary introductory course, you will learn how computers can do useful things with human languages, such as translate from French into English, filter junk email, extract social networks from the web, and find the main topics in the day's news. You will also learn about how computational methods can help linguists explain language phenomena, including automatic discovery of different word senses and phrase structure. Over the past decade, natural language processing has been revolutionized by statistical and probabilistic methods; you will learn about robust approaches to parameter estimation and inference. Our work will include learning new methods, discussions, and hands-on laboratories modifying provided programs. Prerequisites: CMPSCI 220 (or CMPSCI 230, or equivalent). 3 credits.

Compiler Techniques
J. Eliot Moss
This course explores the basic problems in the translation of programming languages focusing on theory and common implementation techniques for compiling traditional block structured programming languages to produce assembly or object code for typical machines. The course involves a substantial laboratory project in which the student constructs a working compiler for a considerable subset of a realistic programming language, within a provided skeleton. The lectures are augmented by a discussion section that covers details of the programming language used to build the compiler, the operating system, the source language, and various tools. Use of computer required. Text: Engineering a Compiler, Cooper and Torczon. 3 credits.

Advanced Algorithms
Ramesh Sitaraman
Principles underlying the design and analysis of efficient algorithms. Topics to be covered include: divide-and-conquer algorithms, graph algorithms, matroids and greedy algorithms, randomized algorithms, NP-completeness, approximation algorithms, linear programming. Prerequisites: The mathematical maturity expected of incoming Computer Science graduate students, knowledge of algorithms at the level of CMPSCI 311. 3 credits.
Software has become ubiquitous in our society. It controls life-critical applications, such as air traffic control and medical devices, and is of central importance in telecommunication and electronic commerce. In this course, we will examine state-of-the-art practices for software testing and analysis to verify software quality. We will initially look at techniques for testing and analyzing sequential programs, and then examine the complexity that arises from distributed programs. The students will be required to complete regular homework assignments and exams, and carry out a group research project extending techniques described in class and/or applying them to new domains. No required text. Papers from the open literature will be assigned and made available.

Prerequisites: CMPSCI 320, Introduction to Software Engineering (or equivalent course). 3 credits.

**CMPSCI 621**  
*Advanced Software Engineering: Analysis and Evaluation*  
*Yuriy Brun*

This class is an in-depth introduction to systems, focusing on principles of system design that cross-cut numerous systems artifacts, including operating systems, databases, runtime systems, and architecture. We will cover all levels of the "system stack", from chips to distributed systems. This class may be used to satisfy systems core requirements. 3 credits.

**CMPSCI 630**  
*Systems*  
*Emery Berger*

This course covers advanced fundamental principles of computer networks, studying foundational material in the field. Topics include advanced network architecture, network algorithmics, network control, network measurement, and wireless networks. The goal of this course is to teach networking fundamentals/techniques that will be useful for years to come. Prerequisites: Introductory (undergraduate level) courses in computer networks (e.g., CMPSCI 453), and algorithms (e.g., CMPSCI 311). Some familiarity with probability and with optimization theory will be helpful. 3 credits.

**CMPSCI 653**  
*Computer Networking*  
*Donald Towsley*

People are able to infer the characteristics of a scene or object from an image of it. In this course, we will study what is involved in building artificial systems which try to infer such characteristics from an image. Topics include: Basics of image formation - the effect of geometry, viewpoint, lighting and albedo on image formation. Basic image operations such as filtering, convolution and correlation. Frequency representations of images. The importance of scale in images. Measurements of image properties such as color, texture, appearance and shape. Inference of motion and structure from moving objects and images. Detecting and recognizing objects in images. Statistical methods in computer vision. Prerequisite: Linear algebra, calculus, basic knowledge of probability, and an ability to program. 3 credits.

**CMPSCI 683**  
*Artificial Intelligence*  
*Shlomo Zilberstein*

In-depth introduction to Artificial Intelligence focusing on techniques that allow intelligent systems to reason effectively with uncertain information and cope limited computational resources. Topics include: problem-solving using search, heuristic search techniques, constraint satisfaction, local search, abstraction and hierarchical search, resource-bounded search techniques, principles of knowledge representation and reasoning, logical inference, reasoning under uncertainty, belief networks, decision theoretic reasoning, representing and reasoning about preferences, planning under uncertainty using Markov decision processes, multi-agent systems, and computational models of bounded rationality. 3 credits.

**CMPSCI 689**  
*Machine Learning*  
*Sridhar Mahadevan*

Machine learning is the computational study of artificial systems that can adapt to novel situations, discover patterns from data, and improve performance with practice. This course will cover the popular frameworks for learning, including supervised learning, reinforcement learning, and unsupervised learning. The course will provide a state-of-the-art overview of the field, emphasizing the core statistical foundations. Detailed course topics: overview of different learning frameworks such as supervised learning, reinforcement learning, and unsupervised learning; mathematical foundations of statistical estimation; maximum likelihood and maximum a posteriori (MAP) estimation; missing data and expectation maximization (EM); graphical models including mixture models, hidden-Markov models; logistic regression and generalized linear models; maximum entropy and undirected graphical models; nonparametric models including nearest neighbor methods and kernel-based methods; dimensionality reduction methods (PCA and LDA); computational learning theory and VC-dimension; reinforcement learning; state-of-the-art applications including bioinformatics, information retrieval, robotics, sensor networks and vision. Prerequisites: undergraduate level probability and statistics, linear algebra, calculus, AI; computer programming in some high level language. 3 credits.
This course will teach students algorithms that enable users to create and edit digital content for computer graphics, such as animation, 3D geometry, textures and sound. The course will start by covering the basics of 3D computer animation and digital shape representation. Then it will cover algorithms that help artists to create, rig, and animate the body and face of a 3D character. The course will proceed with methods that incorporate physics to realistically simulate particles, bodies, deformations, cloth, liquid, fire and sound. The course will also cover methods for acquiring shape and animation data from the real-world based on 3D scanning and motion capture techniques. Finally, it will discuss methods for interactive shape and motion editing. Students will be asked to complete three programming assignments related to physics-based animation. There will also be a final project that involves the implementation of a research idea related to 3D modeling and animation. There are no prerequisites for CMPSCI graduate students, although familiarity with computer graphics, linear algebra and differential equations is important. For undergrads, the course has the following prerequisites: computer graphics (CMPSCI 473, grade of A- or better), Introduction to Linear Algebra (Math 235, grade of A- or better), and Ordinary Differential Equations (Math 331, grade of B or better). 3 credits.

**CMPSCI 690LG**  
**Advanced Logic in Computer Science**  
Neil Immelman  
Rigorous introduction to mathematical logic from an algorithmic perspective. Topics include: Propositional logic: Horn clause satisfiability and SAT solvers; First Order Logic: soundness and completeness of resolution, compactness theorem, automatic theorem proving, model checking. We will learn about and use the Coq theorem prover, Datalog, a Model Checker, and SAT and SMT solvers. Prerequisites: Students taking this course should have undergraduate preparation in discrete math and algorithms. Requirements will include readings, class participation, weekly problem sets, a midterm and a final project. 3 credits.

**CMPSCI 691BL**  
**Seminar - The R Programming Language**  
Brian Levine  
This six-week seminar introduces the R programming language. The focus is on manipulating and visualizing large datasets. We’ll also cover the fundamentals of R and its data structures, using R for calculating statistics, connecting R to a database, and other useful features. Coursework will be comprised of many graded programming assignments. Students are expected to have programming skills at the level of a first year graduate student in computer science. 1 credit.

**CMPSCI 691F**  
**Programming Languages**  
Arjun Guha  
This course surveys the principles of modern programming languages by implementation. Topics include building interpreters for several kinds of programming languages, type-inference algorithms, program analyses, and program verification. We will conclude with topics that bridge to compilers and runtime systems, including program transformations and garbage collection. 3 credits.

**CMPSCI 691O**  
**Seminar - Tools for Explanatory and Tutoring Systems**  
**DELETED**  
This seminar examines recent work in explanatory and tutoring systems. Participants study artificial intelligence in education, a young field that explores theories about learning, and explores how to build software that delivers differential teaching as it adapt its response to student needs and domain knowledge. Such software supports people who work alone or in collaborative inquiry, students who question their own knowledge, and students who rapidly access and integrate global information. This course describes how to build these tutors and how to produce the best possible learning environment, whether for classroom instruction or lifelong learning. The objective of the course is to stimulate awareness of research issues and to promote sound analytic and design skills as they pertain to building knowledge representations and control strategies. Specific topics include collaboration, inquiry, dialogue systems, machine learning, simulators, authoring tools and user models. The course is appropriate for students from many disciplines (computer science, linguistics, education, and psychology), researchers, and practitioners from academia, industry, and government. No programming is required. Students read and critique tools, methods, and ideas, learn how artificial intelligence is applied (e.g., vision, natural language), and study the complexity of human learning through advances in cognitive science. Weekly assignments invite students to critique the literature and a final project requires a detailed specification (not a program) for a tutor about a chosen topic. Students present book chapter material weekly in parallel with readings from the research literature. Several working systems will be available for hands-on critique. 3 credits.

**CMPSCI 701**  
**Advanced Topics in Computer Science**  
**STAFF**  
This is a 6 credit reading course corresponding to the master’s project. The official instructor is the GPD although the student does the work with and is evaluated by the readers of his or her master’s project.
This course covers advanced data management systems, including high-performance database systems and recent big data systems. The course first covers the design and implementation of high-performance database systems including data warehouses, decision support systems, column stores, and parallel databases. It also covers advanced databases to support new data models, including temporal databases, sequence databases, and data stream systems. Then a special focus of the course will be on recent big data systems, including MapReduce-based parallel processing, bigtable-style storage systems, and many new processing algorithms and techniques developed in industry and academia in this setting. The prerequisite is a graduate course on the principles and implementations of traditional database systems, an equivalent of CMPSCI 645. Students with other backgrounds are asked to contact the instructor for approval for enrollment. 3 credits.

Advanced Topics in Database Systems
Yanlei Diao

This course is an introduction to the main tools and techniques used in modern natural language processing research and systems. It will cover the main components of a modern NLP pipeline, such as tagging, entity recognition, parsing, and coreference; the techniques necessary to implement state-of-the-art systems for these tasks, such as classifiers and conditional random fields; and modern tasks and tools to attack them, such as deep learning, embeddings, semi-supervised learning, and dual decomposition for joint inference. There will be weekly lectures, and paper readings, and the grades will be determined based on written summaries of recent research papers and a substantial research project in NLP to be concluded by the end of the course. PhD students only and permission of instructor. 3 credits.

Seminar - Advanced Natural Language Processing
Andrew McCallum

The purpose of this course is to provide students with supplementary material and insights about the software development enterprise. Students meet once a week for a one-hour discussion of software engineering topics whose exploration is intended to provide depth and perspective on the regular material of CS 320. Topics may be suggested by current events or by problems that may arise in the course of the 320 semester. Students will be required to write a term paper as part of the requirements for this course. 1 credit.

Honors Colloquium for CMPSCI 390DB
Alexandra Meliou

Students will study in depth data management topics covered in class. Activities include discussions with the professor and TA, readings, and a project. The project is a semester-long group activity that combines several advanced topics into a practical application. 1 credit.

Honors Colloquium for CMPSCI 391IB
Charles Weems

CMPSCI H391IB provides an opportunity for University Honors students enrolled in CMPSCI 391IB to take a deeper look at some aspect of computer hardware technology and low-level programming. The specific choice of topics and activities is agreed upon by the instructor and student on an individual or team basis at the beginning of the semester. This may involve additional reading and discussion, a special programming project, conducting experiments, writing a research paper, additional lecture material or whatever mode of learning that we collaboratively agree will best meet the interests of each student. The colloquium students meet with the instructor one hour per week for discussion. Depending on how we decide to structure the class, this time may also be used for instructor presentations, review of project progress and help with problems, findings from experiments, preliminary reports on research, etc. Recommended for Sophomores, Juniors, Seniors; Majors. 1 credit.

Honors Colloquium for CMPSCI 453
James Kurose

Students will meet with instructor in small group setting with the class instructor on a weekly basis to discuss related topics of interest, including but not limited to: Internet privacy, network neutrality, network source code implementation. 1 credit.