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.

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 a basic introduction to client-side web programming in HTML, CSS and Javascript. In addition to static and dynamic web page and web site design and implementation, we will cover timely strategies for finding information, participating in virtual communities, managing email, and ensuring privacy. We will survey key social and political topics that are relevant to the Internet such as copyright laws, First Amendment issues, 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.

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.

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.

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.

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.
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.

Special Topics - Programming in C
Nicolas Scarcci

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.

Special Topics - A Hands-on Introduction to UNIX
Robert Walls, Yung-Chih Chen

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.

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.

Reasoning Under Uncertainty
Benjamin Marlin

Development of mathematical reasoning skills for problems that involve uncertainty. Each concept will be illustrated by real-world examples and demonstrated though 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.

Introduction to Computation
Neil Immerman


Social Issues in Computing
Stephen Constantine

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
Phil 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  Honors Colloq
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 and in-class presentations, two exams, and a major term project. Prerequisite: CMPSCI 220 or CMPSCI 287. 4 credits.

CMPSCI 325  Usability
Samuel Dunn, Zachary Dunn
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.

CMPSCI 326  Web Programming
Timothy Richards
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. This course is for CMPSCI minors and majors and counts as a CS Elective toward the major. 3 credits.

CMPSCI 370  Introduction to Computer Vision
Erik Learned-Miller
This introductory computer vision class will address fundamental questions about getting computers to "see" like humans. We investigate questions such as -What is the role of vision in intelligence? -How are images represented in a computer? -How can we write algorithms to recognize an object? -How can humans and computers "learn to see better" from experience? We will write a number of basic computer programs to do things like recognize handwritten characters, track objects in video, and understand the structure of images. Prerequisite: CMPSCI 240 or 383. 3 credits.

CMPSCI 377  Operating Systems
Prashant Shenoy
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.
CMPSCI 383  Artificial Intelligence
David Jensen
The Course explores key concepts underlying intelligent systems. Such systems are increasingly deployed in consumer products and online services, and this course takes you "under the hood" to explain basic technologies used for representation, learning, and reasoning, and to explain how these technologies combine to produce intelligent behavior. While several other CMPSCI courses focus on specific technologies and applications (reasoning under uncertainty, machine learning, information retrieval, vision, and robotics), this course surveys a broad array of such technologies and explains how they can be combined to produce the intelligent systems of today and tomorrow. Prerequisites: (CMPSCI 287 or CMPSCI 220 or CMPSCI 230) and (CMPSCI 240). 3 credits.

CMPSCI 391IB  Seminar - Inside the Box: How Computers Really Work
Charles Weems  Honors Colloq
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 403  Introduction to Robotics: Mechanics, Dynamics, and Control
Roderic Grupen
This course covers basic methods and concepts in order to explain how robots work. We will study how they sense things in the world, how you make a robot move, and how robots can make their own decisions. We will study mechanisms (kinematics and dynamics), actuators, sensors, (with a focus on machine vision), signal processing, feedback control theory, machine learning, and path planning. Students will build software systems to control a simulated mobile manipulator and do experiments with system identification and control, image processing, and path planning to reinforce the material presented in class. Prerequisites: Math 235 (or Math 236) and CMPSCI 220 (or CMPSCI 287 or CMPSCI 230). 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 an optional (but recommended) 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 445  Information Systems
Yanlei Diao
This course is an introduction to the efficient management of large-scale data. The course includes principles for representing information as structured data, query languages for analyzing and manipulating structured data, and core systems principles that enable efficient computation on large data sets. Classical relational database topics will be covered (data modeling, SQL, query optimization, concurrency control), as well as semi-structured data (XML, JSON), and distributed data processing paradigms (e.g. map-reduce). Additional application topics may include web application development, data integration, processing data streams, database security and privacy. Prerequisite: CMPSCI 220 (or 287, or 230) and CMPSCI 311. 3 credits.

CMPSCI 453  Computer Networks
Parviz Kermani
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 520 Software Engineering: Synthesis and Development**

Introduces students to the principal activities involved in developing high-quality software systems in a variety of application domains. Topics include: requirements analysis, formal and informal specification methods, process definition, software design, testing, and risk management. The course will pay particular attention to differences in software development approaches in different contexts. Prerequisites: CMPSCI 320 with a grade of ‘C’ or better, or E&C-ENG 373 with a grade of ‘C’ or better. 3 credits.

**CMPSCI 521 Software Engineering: Analysis and Evaluation**

Yuriy Brun

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 529 Software Engineering Project Management**

Leon Osterweil

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.

**CMPSCI 585 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.

**CMPSCI 591NR Seminar - Neural Networks-An Introduction**

Hava Siegelmann

We will introduce neural networks and modeling of brain functions: the inspiration, engineering applications, theoretical analysis, as well as psychological and biological modeling. We will review different neural networks within the two groups: Feed forward networks, applications, and approximation theorem; Recurrent neural networks and computational power; We will also focus on supervised and unsupervised learning with applications in clustering; representability; and Models of memory, diseases, and healthy perception. This course counts as a CS Elective toward the CMPSCI major. Prerequisite: CMPSCI 311. 3 credits.
**CMPSCI 610  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 an optional (but recommended) 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.

**CMPSCI 611  Advanced Algorithms**  
Andrew McGregor  
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.

**CMPSCI 620  Advanced Software Engineering: Synthesis and Development**  
**DELETED**  
This course examines the varied approaches to the development of computer software. We examine various ideas about how software products should be structured and function. We then examine how software processes serve as vehicles for manufacturing such products. This approach facilitates the direct study of different software development approaches, and a more direct study of their effects on the products they produce. This approach will be used by students, who will examine representative current software development product and process approaches in in-class presentations and written project papers as part of their coursework. 3 credits.

**CMPSCI 621  Advanced Software Engineering: Analysis and Evaluation**  
Yuriy Brun  
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 645  Database Design and Implementation**  
Alexandra Meliou  
This course covers the design and implementation of traditional relational database systems and advanced data management systems. The course will treat fundamental principles of databases: the relational model, conceptual design, query languages, and selected theoretical topics. We also cover core database implementation issues including storage and indexing, query processing and optimization, as well as transaction management, concurrency, and recovery. Additional topics will address the challenges of modern Internet-based data management. These include data mining, provenance, information integration, incomplete and probabilistic databases, and database security. 3 credits.

**CMPSCI 646  Information Retrieval**  
James Allan  
The course will cover basic and advanced techniques for text-based information systems. Topics covered include retrieval models, indexing and text representation, browsing and query reformulation, data-intensive computing approaches, evaluation, and issues surrounding implementation. The course will include a substantial project such as the implementation of major elements of search engines and applications. Prerequisite: CMPSCI 446, CMPSCI 445, or equivalent. 3 credits.

**CMPSCI 660  Advanced Information Assurance**  
Kevin Fu  
This course provides an in-depth examination of the fundamental principles of information assurance. While the companion course for undergraduates is focused on practical issues, the syllabus of this course is influenced strictly by the latest research. We will cover a range of topics, including authentication, integrity, confidentiality of distributed systems, network security, malware, privacy, intrusion detection, intellectual property protection, and more. Prerequisites: CMPSCI 460 or 466, or equivalent. 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.

**CMPSCI 691BD**  **Seminar - Readings in Big Data Systems**  
*Arunkumar Venkataramani*

This course will cover recent advances in big data systems, an increasingly important and challenging area that deals with the processing and analysis of extremely large amounts of data. We will present and discuss recent research papers in big data systems coupled with a project involving one of several real-life large-scale datasets crawled from a variety of online sources. The projects will enable students to familiarize themselves with tools and platforms such as Hadoop, EC2, etc. as well as adaptation of data mining algorithms so as to meet the challenges of data analysis at scale. The class may optionally be taken for 1 credit without the project option. Lect A=3 cr; Lect B=1 cr.

**CMPSCI 691BM**  **Seminar - Bayesian Methods for Text**  
*Hanna Wallach*

This course will provide an overview of the key ideas that underlie the Bayesian approach to modeling data, with a particular focus on text. The course will primarily consist of discussing, deriving, and implementing a number of Bayesian models of text (and their associated inference algorithms) in order to understand their fundamental strengths and weaknesses, as well as explore the relationships between them. The aim of the course is to develop the knowledge and skills needed to design, implement, and apply such models to real-world data. Students entering the course should have good programming skills, knowledge of algorithms, knowledge of probability, statistics, or machine learning, and a strong interest in text analysis. To facilitate productive discussion, students with diverse research backgrounds and interests are especially encouraged to participate. 3 credits.
In recent years, the ability to continuously monitor activities, health, and lifestyles of individuals using sensor technologies has reached unprecedented levels. Wearable "on-body" sensors now enable routine and continuous monitoring of a host of physiological signals (e.g., heart rate, blood pressure, respiratory rate, blood glucose, etc.), physical activity (e.g. calorie expenditure), and sleep patterns (e.g. REM vs deep sleep). In addition, the typical smartphone comes routinely equipped with a plethora of sensors for monitoring both activity and location, enabling (in combination with other sensors) higher-order inferences about more complex human activities/behavioral states (e.g., stress, addiction, etc.). Such ubiquitous sensing in daily life, referred to as mobile health, promises to revolutionize our understanding of the varied social, environmental, and behavioral context (and potentially determinants) of a wide range of human activities and health conditions. This course is an exploration of challenges in mobile health including: a) practical considerations including energy-efficiency, interruptions, wearability, privacy, etc. b) inference of key health assessments from sensor data including stress, mood, eating behavior, sleep patterns, calorie intake and expenditure, mental health, etc. c) personalized health assessment by combining continuous mobile sensor data using a variety of on-body sensors (chestband, wristband, smartphone) d) novel interventions that can take advantage of these models to elicit changes in health behavior. Students from diverse research backgrounds/interests are encouraged to attend for more productive discussion. The course will consist of reading, presenting, and discussing papers, and working with on-body sensors and/or datasets as appropriate. We will also have guest lectures that you are expected to attend. Students taking the class for 3 credits will be expected to define and execute a semester-long research project. Those interested in doing hands-on projects will be given access to Google Nexus Prime smartphones, Zephyr Bioharness chestbands with EKG, respiration, skin conductance, and accelerometer sensors, and wristbands that can sense activity patterns. Students interested in data analytics will have access to several datasets. Lect A=3 cr; Lect B=1 cr.

We will introduce neural networks and modeling of brain functions: the inspiration, engineering applications, theoretical analysis, as well as psychological and biological modeling. We will review different neural networks within the two groups: Feed forward networks, applications, and approximation theorem; Recurrent neural networks and computational power; We will also focus on supervised and unsupervised learning with applications in clustering; representability; and Models of memory, diseases, and healthy perception. 3 credits.

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 adapts 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.

This is a research course in statistics of networks and graphs. Topics will be wide-ranging, including models for cross-sectional and dynamic networks, partially-observed networks, and networks with latent structures. Also included are network sampling, non-parametric statistical methods for networks, and data stream methods for graphs. Course discussion will highlight comparisons across frameworks for statistical analysis of networks, as well as across application areas. Classes will require reading research articles before class, listening to presentations of the articles, and discussing them. Students will be responsible for in-class presentations of at least one article over the semester. This course is offered for either one or three credits. In addition to class participation as described above, students enrolled for 3 credits will be responsible for completing a course project. Meets with STATISTICS 691SG. Prerequisites: STAT 516, 608, or permission of instructor. Lect A=3 cr; Lect B=1 cr.
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 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.

**CMPSCI 791GC Seminar - Graph Canonization**

*Neil Immerman*

What is this course about? Martin Grohe has proved a breakthrough result showing that all classes of graphs that exclude some minor can be canonized in polynomial time via first-order logic with fixed-point and counting operators. This is a major advance in graph isomorphism research and in trying to capture the language of (order-independent) polynomial-time properties of graphs or other logical structures. Grohe has produced a draft monograph of this work called, Descriptive Complexity, Canonisation, and Definable Graph Structure Theory (http://www2.informatik.hu-berlin.de/~grohe/pub/cap/index.html), available online. We will read and understand the monograph together, with students taking equal share of presentations. 3 credits.

**CMPSCI 791SU Seminar - Machine Learning for Shape Understanding**

*Evangelos Kalogerakis*

The emergence of modern geometry acquisition devices, such as the Kinect, and the appearance of large-scale shape repositories, such as the Google Warehouse, are revolutionizing computer graphics, making three-dimensional content ubiquitous. The need for algorithms that understand and intelligently process 3D shapes is thus greater than ever. This seminar will provide a state-of-the-art overview of the field with particular emphasis on machine learning applications to shape understanding. The seminar will cover topics such as extraction of 3D geometric descriptors and symmetries from shapes, probabilistic models for shape segmentation, retrieval, and reconstruction, 3D template fitting, discovery of structural and contextual relationships of shapes in scenes, generative models of shapes, and applications to 3D modeling. Students read, present, and critique state-of-the-art research papers on the above topics and learn how machine learning can be used for 3D shape analysis and synthesis. Familiarity with probability, statistics, and linear algebra is required. Experience with vision, graphics, or machine learning useful but not required. 3 credits.

**CMPSCI 891M Theory of Computation**

*Neil Immerman*

The theory seminar is a weekly meeting in which topics of interest in the theory of computation - broadly construed - are presented. This is sometimes new research by visitors or local people. It is sometimes work in progress, and it is sometimes recent material of others that some of us present in order to learn and share. This is a one-credit seminar which may be taken repeatedly for credit up to six times.

**CMPSCI H320 Honors Colloquium for CMPSCI 320**

*Leon Osterweil*

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.

**CMPSCI H377 Honors Colloquium for CMPSCI 377**

*Prashant Shenoy*

The purpose of the OS honors colloquium is to supplement the regular class materials with more in-depth insights into OS kernel design as well as the design of specialized OS kernels for embedded devices, smartphones, and tablets. Students meet once a week for a one-hour discussion of OS topics that will provide additional depth and perspective on the regular 377 material. There will be an opportunity to study ongoing research topics and additional topics may be suggested based on current developments in the field. Students will be required to write a term paper as part of the requirements for this course. 1 credit.
CMPSCI H391IB   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.