CMPSCI 105  Computer Literacy (R2)
William Verts
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. Pre-registration is not available to majors and pre-majors. 3 credits.

CMPSCI 120  Introduction to Problem Solving with the Internet (R2)
W. Richards Adrion  Honors Colloq
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). CMPSCI majors and pre-majors may not pre-register. 3 credits.

CMPSCI 121  Introduction to Problem Solving with Computers (R2)
Robert Moll
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.

CMPSCI 145  Representing, Storing, and Retrieving Information
William Verts
An introductory course in the use of data in computer systems, a core course for the Information Technology certificate. Formats for representing text, numbers, sound, images, etc., as strings of bits. Equations of lines and curves, modeling of synthetic scenes (i.e., ray tracing), exploring the frequency domain and holography. Basic information theory, use and limitations of file compression and encryption. Structured databases and how to use them. Information retrieval in heterogenous environments such as the Web. XML as a language for defining new formats for representing data. Review of historical, pre-computer methods of information representation. Prerequisites: "Basic computer literacy", i.e., user-level familiarity with a modern operating system and some experience with application programs. Tier I math skills. Recommended for First Year and Sophomore Non-Majors. Prerequisite: R1. 3 credits.

CMPSCI 187  Programming with Data Structures (R2)
Wendy Lehnert
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. There will be weekly assignments, consisting of programming and written exercises, a midterm, and a final exam. 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.
CMPSCI 191A  Computer Science Majors RAP Seminar  
David Barrington, Wendy Lehnert  
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 191P  Introduction to Programming  
Wendy Lehnert  
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. This course does not count toward the CMPSCI major without UPD Approval. 3 credits.

CMPSCI 197U  Special Topics - A Hands-on Introduction to UNIX  
Brian Levine, Timothy Wood  
This course offers a 4-week introduction to working with Unix, and it is intended to help students work with tools commonly used in CS courses. The fall semester offering of this class moves more quickly and covers more topics than the spring semester. 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  
Development of individual skills necessary for designing, implementing, testing and modifying larger programs, including: advanced uses of integrated design environments, design strategies and patterns, testing, working with large code bases and libraries, code refactoring, use of debuggers and tools for version control, and system build tools. There will be a significant programming project, done by pairs in phases, and a mid-term and final examination. Prerequisite: CMPSCI 187 or ECE 242. 4 credits.

CMPSCI 240  Reasoning About Uncertainty  
Hanna Wallach  
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.

CMPSCI 250  Introduction to Computation  
David Barrington  
Honors Colloq  
Lecture, discussion. Basic concepts of discrete mathematics useful to computer science: set theory, strings and formal languages, propositional and predicate calculus, relations and functions, basic number theory. Induction and recursion: interplay of inductive definition, inductive proof, and recursive algorithms. Graphs, trees, and search. Finite-state machines, regular languages, nondeterministic finite automata, Kleene's Theorem. Problem sets, 2-3 midterm exams, timed final. Corequisite: MATH 132/136 or equivalent. Prerequisite: MATH 131 (or MATH 132 or MATH 235) and CMPSCI 187 (or ECE 242). 4 credits.

CMPSCI 291SP  Seminar - Computer Systems Principles  
Emery Berger  
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 to fulfill the requirement of CMPSCI 201 for students in the CMPSCI major. This course can be used as a "core" course for students in the CMPSCI minor. Prerequisites: CMPSCI 187 or ECE 242. 3 credits.
Additional required credit for CMPSCI 291SP. 1 credit.

CMPSCI 305  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  
Ramesh Sitaraman  
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  
Mark Corner  
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 in Flash (or another user interface prototyping tool) 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 348  Introduction to Knowledge Discovery  
David Jensen  
Knowledge discovery is the process of discovering useful regularities in large and complex data sets. The field encompasses techniques from artificial intelligence (representation and search), statistics (inference), and databases (data storage and access). When integrated in to useful systems, these techniques can help human analysts make sense of vast stores of digital information. This course presents the fundamental principles of the field, familiarizes students with the technical details of representative algorithms, and connects these concepts to applications in industry, science, and government, including fraud detection, marketing, scientific discovery, and web mining. The course assumes that students are familiar with basic concepts and algorithms from probability and statistics. Prerequisites: CMPSCI 187 (or ECE 242), CMPSCI 240, and CMPSCI 250. Students that take the Fall 2010 offering of CMPSCI 348 can count it towards any component of the BS degree requiring CMPSCI 383. 3 credits.
CMPSCI 365  Digital Forensics
Brian Levine
The goal of forensics is to gather artifacts for refinement into evidence that supports or refutes a hypothesis about an alleged crime or policy violation. Done correctly, forensics represents the application of science to law. This course is a broad introduction to forensic investigation of digital information and devices. We will cover the acquisition, analysis, and courtroom presentation of information from file systems, operating systems, networks, database systems, common applications, media files, and cell phones. Students do not need experience with these systems. We will review the use of some professional tools that automate data harvesting, however, the primary goal of the class is to understand why and from where evidence is recoverable in these systems. Several assignments involve coding forensic tools from scratch. A separate lab discussion ensures that students gain hands-on instruction and experience with these systems each week. For a portion of the class, we will cover some relevant issues from the law, privacy, and current events. Thus, the class serves the well-rounded student who is eager to participate in class discussion on a variety of technical and social issues. Prerequisites: CMPSCI 220 or CMPSCI 287. 4 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 187 (or ECE 242) and CMPSCI 201 (or ECE 232). Strongly recommend CMPSCI 220 with a 'C' or better prior to taking this course. 4 credits.

CMPSCI 391LI  Seminar - Legal Issues in Computing
Brian Levine
In this seminar, we study and discuss the broad legal issues that are shaping how we use, deploy, and research computing. We will also examine how, conversely, advances in computing and changing social interactions are challenging existing legal concepts. Topics include recent and important statutes and case law concerning search and seizure, privacy, digital evidence, and possession of contraband. Students are assumed to be familiar with general computing concepts and applications, but instructors will provide an introduction to legal concepts. Students will be required to complete and comment on assigned readings, make 1--2 presentations, and complete a semester-long project. The undergraduate and graduate versions of this course meet concurrently. More substantive comments, presentations, and projects are expected of graduate students. This course counts as a CMPSCI Elective toward the CMPSCI major. 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 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
Gerome Miklau
Introduction to database management. Relational database topics include data modeling, query languages, database design, and query optimization. Alternative data management approaches will be covered, including semi-structured data, XML, and text retrieval. Application topics will include web data management, integration of data sources, security and privacy. Prerequisite: CMPSCI 220 (or 287) and CMPSCI 311. 3 credits.

CMPSCI 453  Computer Networks
James Kurose
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 201, CMPSCI 311, and CMPSCI 377. 3 credits.
CMPSCI 460  Introduction to Computer and Network Security
Christopher Misra, Brian Levine
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 466  Applied Cryptography
Kevin Fu
Applied cryptography spans many disciplines including computer systems, computation, mathematics, and law. This experimental course aims to teach students both the theoretical foundations of cryptography and the humility of building practical cryptographic systems. Topics will include computational indistinguishability, cryptographic attacks, applications of cryptography, and advanced topics in block ciphers, hash functions, and public key cryptography. Students will be evaluated based on final group projects, class participation, problem sets with hands-on labs, a midterm, and a final exam. Prerequisites: CMPSCI 311. Prior experience with number theory is recommended. 3 credits.

CMPSCI 473  Introduction to Computer Graphics
Rui Wang
This course teaches the fundamentals of 2D and 3D computer graphics. Students will learn OpenGL and standard graphics algorithms. Specific topics covered include image processing, modeling, rendering, illumination, shadows, textures, and programmable shaders. The completion of this course will prepare students ready for any advanced computer graphics course. Students are expected to be familiar with basic linear algebra (vector and matrix arithmetic) and Java programming. Prerequisites: CMPSCI 187 (or ECE 242) and MATH 235. 3 credits.

CMPSCI 513  Logic In Computer Science
Neil Immerman
Rigorous introduction to mathematical logic from an algorithmic perspective. Topics include: Propositional logic: syntax, semantics, soundness and completeness of natural deduction and resolution; Horn clause satisfiability and SAT solvers; First Order Logic: syntax, semantics and definability; Soundness and completeness of natural deduction for First-Order Logic; Compactness Theorem and Resolution Theorem Proving for First-Order Logic; Logic Programming; Decidable theories, SAT Modulo Theories. Prerequisites: CMPSCI 250 and CMPSCI 311. 3 credits.

CMPSCI 521  Software Engineering: Analysis and Evaluation
Lori Clarke
Software systems have become an integral part of our societal infrastructure. Software 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 testing and analyzing software systems that require high assurance. We will initially look at techniques developed for sequential systems but then examine the complexity that arises from distributed systems. In addition to homework assignments and exams, students will be required to carry out an individual or group project that applies or extends some of the techniques described in class. 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. 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 535  Computer Architecture**

*Charles Weems*  
*Honors Colloq*

The structure of digital computers is studied at several levels, from the basic logic level, to the component level, to the system level. Topics include: the design of basic components such as arithmetic units and registers from logic gates; the organization of basic subsystems such as the memory and I/O subsystems; the interplay between hardware and software in a computer system; the von Neumann architecture and its performance enhancements such as cache memory, instruction and data pipelines, coprocessors, and parallelism. Weekly assignments, semester project, 2 hours exams, final. Prerequisites: CMPSCI 201 and CMPSCI 220 (or CMPSCI 287). 3 credits.

**CMPSCI 585  Introduction to Natural Language Processing**

**DELETED**

The field of natural language processing is concerned with practical and theoretical issues that arise in getting computers to perform various tasks with human languages. In this introductory course you will learn about automated techniques for parsing sentences, tagging words according to their part-of-speech, encoding spelling rules, modeling language semantics, learning to filter junk email, clustering news articles by topic, and translating from one language into another. Our toolbox will contain algorithms, machine learning models, and linguistic concepts aimed at programming useful systems on real data. Whether you are interested in applications in information retrieval and data mining, in computational humanities, or in modeling the human language faculty, this course will help you on your way. Prerequisites: CMPSCI 220 (or CMPSCI 287, CMPSCI 291SP or equivalent). LINGUIST 409 or STATISTIC 501 or 515 are helpful, but not required. 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 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 617  Computational Geometry**

*Ileana Streinu*

Geometric algorithms lie at the heart of many applications, ranging from computer graphics in games and virtual reality engines to motion planning in robotics or even protein modeling in biology. This graduate course is an introduction to the main techniques from Computational Geometry, such as convex hulls, triangulations, Voronoi diagrams, visibility, art gallery problems, and motion planning. The class will cover theoretical as well as practical aspects of the field. The goal of the class is to enable students to exploit a broad range of algorithmic tools from computational geometry to solve problems in a variety of application areas. Prerequisite: Mathematical maturity; CMPSCI 611 or CMPSCI 601. Eligibility: Graduate students only. Others with permission of instructor. 3 credits.

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

*Lori Clarke*

Software systems have become an integral part of our societal infrastructure. Software 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 testing and analyzing software systems that require high assurance. We will initially look at techniques developed for sequential systems but then examine the complexity that arises from distributed systems. In addition to homework assignments and exams, students will be required to carry out an individual or group project that applies or extends some of the techniques described in class. 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 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 653**  
**Advanced Computer Networking**  
_Arunkumar Venkataramani_  
This course covers advanced fundamental principles of computer networks, studying foundational material in the field. Topics include protocol mechanisms and implementation principles, protocol specification/verification techniques, network algorithmics, advanced network architecture, network simulation, performance analysis, and measurement. Prerequisites: introductory (undergraduate level) courses in computer networks (e.g., CMPSCI 453/591,), operating systems (e.g., CMPSCI 377), and algorithms (e.g., CMPSCI 311). Some familiarity with probability will also be needed. 3 credits.

**CMPSCI 683**  
**Artificial Intelligence**  
_Victor Lesser_  
In-depth introduction to Artificial Intelligence focusing on techniques that allow intelligent systems to operate in real-time and cope with missing information, uncertainty, and limited computational resources. Topics include: advanced search and problem-solving techniques, resource-bounded search, principles of knowledge representation and reasoning, meta-reasoning, reasoning under uncertainty, Bayesian networks and influence diagrams, decision theory and the value of information, planning and scheduling, intelligent agents architectures, and learning. Prerequisites: Undergraduate background in Computer Science, and an undergraduate Artificial Intelligence course (CMPSCI 383 or equivalent). 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 691CN**  
**Seminar - Complex Networks**  
_Donald Towsley_  
According to Wikipedia, a complex network is "a network with non-trivial topological features that do not occur in simple networks such as lattices or random graphs". The structure of links in Wikipedia itself is a complex network! Social networks, delay tolerant networks, peer-to-peer networks and the Web are all examples of complex networks. Given the pervasity of such networks in our everyday lives, it is not surprising that they have gained the attention of many researchers. Finding connections between such networks from topological, technological and economic perspectives is a novel and fascinating field. The goal of this course is to provide an introduction to and overview of complex networks. We will cover topics such as power laws, a ubiquitous feature of complex networks, and techniques for characterizing complex networks, social networks, search and navigation in complex networks, epidemic spread, delay tolerant networks, P2P networks, and network economics. The readings will balance practice, theory and mathematical background. Lect A=3 credits; Lect B=1 credit.
Crowdsourcing, or the act of outsourcing a task to the crowd, has the potential to revolutionize information collection and processing systems by enabling in-depth, large-scale, and cost-effective information gathering, more accurate techniques for information extraction from data. Crowdsourced data processing is effective when humans are better than existing automated computer algorithms, for example labeling images, transcribing speech, annotating text, transcribing scanned documents, and so on. Crowdsourcing also provides a powerful mechanism for creating data about the physical world, particularly through the use of mobile phones and their rich set of onboard sensors (GPS, audio, video, accelerometer, etc). These sensors can be utilized to provide continuous and unprecedented visibility into the state of the world across many scales. This course is an exploration of the opportunities and challenges of crowdsourcing, and will discuss a variety of perspectives on the topic including applications, incentives, data quality assurance, privacy, general design principles, incorporation of ML/AI techniques, modeling, integration with social networks and cloud computing, etc. Students from diverse research backgrounds/interests are encouraged to attend for more productive discussion. Lect A=3 credits; Lect B=1 credit.

**CMPSCI 691CS**  Seminar - Crowdsourcing: Opportunities and Challenges  
Deepak Ganesan  
Crowdsourcing, or the act of outsourcing a task to the crowd, has the potential to revolutionize information collection and processing systems by enabling in-depth, large-scale, and cost-effective information gathering, more accurate techniques for information extraction from data. Crowdsourced data processing is effective when humans are better than existing automated computer algorithms, for example labeling images, transcribing speech, annotating text, transcribing scanned documents, and so on. Crowdsourcing also provides a powerful mechanism for creating data about the physical world, particularly through the use of mobile phones and their rich set of onboard sensors (GPS, audio, video, accelerometer, etc). These sensors can be utilized to provide continuous and unprecedented visibility into the state of the world across many scales. This course is an exploration of the opportunities and challenges of crowdsourcing, and will discuss a variety of perspectives on the topic including applications, incentives, data quality assurance, privacy, general design principles, incorporation of ML/AI techniques, modeling, integration with social networks and cloud computing, etc. Students from diverse research backgrounds/interests are encouraged to attend for more productive discussion. Lect A=3 credits; Lect B=1 credit.

**CMPSCI 691GG**  Seminar - Applied Information Theory  
Erik Learned-Miller  
This course will introduce the basic concepts of Information Theory: entropy, relative entropy, mutual information, channel capacity, and rate distortion. Applications, rather than proofs, will be emphasized. In addition, the statistical problem of computing information theoretic quantities from data will be emphasized. Non-parametric and semi-parametric statistical models will also be covered. Although this course is listed as a seminar, it will be taught as a regular lecture course with homework assignments and exams. 3 credits.

**CMPSCI 691LI**  Seminar - Legal Issues in Computing  
Brian Levine  
In this seminar, we study and discuss the broad legal issues that are shaping how we use, deploy, and research computing. We will also examine how, conversely, advances in computing and changing social interactions are challenging existing legal concepts. Topics include recent and important statutes and case law concerning search and seizure, privacy, digital evidence, and possession of contraband. Students are assumed to be familiar with general computing concepts and applications, but instructors will provide an introduction to legal concepts. Students will be required to complete and comment on assigned readings, make 1--2 presentations, and complete a semester-long project. The undergraduate and graduate versions of this course meet concurrently. More substantive comments, presentations, and projects are expected of graduate students. 3 credits.

**CMPSCI 691O**  Seminar - Tools for Explanatory and Tutoring Systems  
Beverly Woolf  
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  
James Allan  
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.
The study of the resources required to solve different problems in various abstract models of computation. Sequential computation: Turing machines, non-determinism, alternation, algebraic automaton theory. Parallel computation: Boolean circuits, branching programs, uniformity. Lower bounds for circuit models. Descriptive Complexity. Possible optional topics depending on student interest: Approximation of NP-complete problems, interactive proofs, non-uniform finite automata, dynamic complexity. Prerequisite: CMPSCI 601 or permission of instructor. Intended for interested general graduate students as well as those concentrating in theory. Course format: Probably 2-3 problem sets, class presentation (may be survey or actual research, might lead to a synthesis paper). Textbook: Xeroxed notes and research papers. 3 credits.

This graduate course covers advanced data and information management systems. It covers the design and implementation of advanced database systems including data warehouses, decision support systems, and data mining. It examines several generations of parallel and distributed databases, including the most recent MapReduce-based parallel systems that are widely used in information industry. This course further includes temporal databases, sequence databases, and data stream systems. Finally, it presents advanced topics in probabilistic data management. 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.

Planning is the hallmark of intelligence and one of the most fundamental problems in AI research. One of the hardest challenges in the field is to find program-like "generalized plans", which can solve many different problem instances. Such plans inherently require branches and loops; finding and analyzing them therefore requires a synthesis of methods from AI and software model checking. This seminar covers different aspects of this problem, addressing questions such as: How must plan representations be extended to express such plans? How can we reason about large state spaces with unknown properties and quantities of objects? How can we efficiently find generalized plans and characterize the range of problems they solve? The first part of the seminar will include a review of the main streams of work in planning: planning as heuristic search, using planning graph analysis, planning as satisfiability or constraint satisfaction, planning as model checking, and planning under uncertainty using nondeterministic and probabilistic methods. We will then examine ways to synthesize generalized plans or learn them from examples, derive applicability conditions, verify their correctness, and establish new metrics to evaluate them. We will discuss applications and connections between these problems and program synthesis, program verification, automatic service composition, and robot task planning. Lect A=3 credits; Lect B=1 credit.

This graduate course covers advanced data and information management systems. It covers the design and implementation of advanced database systems including data warehouses, decision support systems, and data mining. It examines several generations of parallel and distributed databases, including the most recent MapReduce-based parallel systems that are widely used in information industry. This course further includes temporal databases, sequence databases, and data stream systems. Finally, it presents advanced topics in probabilistic data management. 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. May be repeated one time for credit. Lect A=3 credits; Lect B=1 credit.

** DELETED **

The class will meet once a week for two hours. The topic of biologically inspired artificial intelligence will be introduced in terms of its high applicability to modeling the brain and body systems, conditions and diseases; the class will also investigate improved machine learning, improved robotic interfaces, improved dialog systems, and more. During the semester students will engage in research: learning new methods, developing them further and applying them to the topics they find most interesting. Projects will be done in small groups with weekly supervision. Research papers are likely to result from the class. Applicants should ask me about pre-requisites if they have questions about qualifications. 2 credits.

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.
This 1-credit Honors course is designed to combine community service learning with a mechanism for reinforcing learning through tutoring. The Honors students will tutor students at Holyoke Community College (HCC) enrolled in a one-hour-per-week, seven-week Web design class, “IT is all about me.” The HCC curriculum is similar to the web programming component of CMPSCI 120, but with much less depth and a focus on developmental students. The Honors students will work with minority and/or non-native-English speakers in a course for students with weak college preparation, but with the intent to introduce students to the Web and to encourage students to move on to IT majors. Tutoring will occur at 2-3 in-person visits to HCC and via phone, email, text messaging, etc. It is expected that tutoring HCC students will reinforce learning of CMPSCI 120 content. Each student will attend a two-day training session at the Learning Resource Center and take a refresher course every month. Students will prepare a self-reflexive paper that discusses the tutoring experience, comments, outcomes and suggestions about the experience and improvements. As a prerequisite, students must be currently enrolled in or have previously taken CMPSCI 120. Students fluent in Spanish or Russian particularly are encouraged to enroll.

**CMPSCI H120**  
Honors Colloquium for CMPSCI 120  
*W. Richards Adrion*

The Honors students for CMPSCI 250 will read most of Hofstadter's book _Godel, Escher, Bach: An Eternal Golden Braid_. We will consider connections between the book and the content of CMPSCI 250, particularly formal logic and number theory. Students will solve problems using Hofstadter’s formalism and write short essays on his ideas. Students may be asked to prepare short oral presentations on a topic related to the book. There will be one hour-long class meeting per week, at a time to be arranged. The seminar will be one credit, with a grade independent of the CMPSCI 250 grade. Concurrent registration in 250 is required, but if space permits we may admit students who have _previously_ taken 250. 1 credit.

**CMPSCI H250**  
Honors Colloquium for CMPSCI 250  
*David Barrington*

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. Honors students will be expected to assume responsibility for some of the more complex aspects of the project as well as to study more advanced concepts. 1 credit.

**CMPSCI H320**  
Honors Colloquium for CMPSCI 320  
*Leon Osterweil*

The honors section of CMPSCI 535 provides an opportunity for University Honors students enrolled in the class to take a deeper look at some aspect of computer architecture or its underlying technology. The specific choice of topics is agreed upon by the instructor and student on an individual basis. Students may choose to explore the history of some aspect of architecture or technology, look at market influences on the science and engineering of computer hardware, experiment with a novel computer design through simulation, conduct a series of in-depth readings leading to a semester thesis, or other suitable work done under regular consultation with the instructor. Recommended for Juniors, Seniors; Majors. 1 credit.