

# UMASS-COMPUTER SCIENCE - Course Descriptions

## Fall 2015

- COMPSCI 105**                      **Computer Literacy (R2)**                      **CREDITS 3**  
*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 119 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.
- COMPSCI 119**                      **Introduction to Programming**                      **CREDITS 3**  
*William Verts*  
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.
- COMPSCI 120**                      **Introduction to Problem Solving with the Internet (R2)**                      **CREDITS 3**  
*William Verts*  
The Internet is a goldmine of information and software resources for those who know how to plug in and navigate it. Originally designed by computer scientists for computer scientists, the net is now a driving force behind life in the information age and a new global economy. This course will provide non-CMPSCI majors with timely skills needed to tap the net as well as an introduction to basic networking, client-side web programming in HTML, CSS, and Javascript, and server-side programming in Python. In addition to static and dynamic web page and web site design and implementation, we will cover strategies for finding information, managing e-mail, and ensuring privacy. We will survey current social, technical, and political topics that are relevant to the Internet such as spam and malware, net neutrality, censorship, copyright laws, and public key cryptography. Prerequisites: some hands-on experience with PCs or MACs or UNIX (programming experience is NOT required). Not for CMPSCI majors. 3 credits.
- COMPSCI 121**                      **Introduction to Problem Solving with Computers (R2)**                      **CREDITS 4**  
*Gordon Anderson*  
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 119 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.
- COMPSCI 187**                      **Programming with Data Structures (R2)**                      **CREDITS 4**  
*Gerome Miklau*  
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.
- COMPSCI 190DM**                      **A Mathematical Foundation for Informatics**                      **CREDITS 3**  
*David Barrington*  
Mathematical techniques useful in the study of computing and information processing. The mathematical method of definition and proof. Sets, functions, and relations. Combinatorics, probability and probabilistic reasoning. Graphs and trees as models of data and of computational processes. Prerequisite: R1 math skills recommended. Not intended for Computer Science majors – students interested in a majors-level treatment of this material should see CMPSCI 240 and 250, or MATH 455. 3 credits.

<b>COMPSCI 190IN</b>	<b>Introduction to Informatics</b>	<b>CREDITS 3</b>
<i>Michelle Trim</i>		
An introduction to the main concepts of Informatics. There are several "Big Ideas" in computing, including but not limited to abstraction, data and information, algorithms, programming, and analysis of both computational problems and computational artifacts. This class provides an introduction to those ideas and considers some of the ways that those computing principles might be used to solve real world problems. Computer-based assignments are an integral part of this course but no programming knowledge or prior programming experience is expected or required. Not for CMPSCI majors. 3 credits.		
<b>COMPSCI 191CICS1</b>	<b>Computing and Informatics Exploratory First Year Seminar</b>	<b>CREDITS 1</b>
<i>Gregory Boisseau</i>		
A weekly seminar for students in the Computing and Informatics Exploratory Track. Exploration of strategies for success at UMass and computer science topics through presentations and problem-solving games. Open only to first-year students in the Computing and Informatics Exploratory Track. This course does not count toward any requirements for the CMPSCI major or minor. 1 credit.		
<b>COMPSCI 191CICS2</b>	<b>Computing and Informatics Exploratory RAP First Year Seminar</b>	<b>CREDITS 1</b>
<i>Gregory Boisseau</i>		
A weekly seminar for students in the Computing and Informatics Exploratory Track RAP residential program. Exploration of strategies for success at UMass and computer science topics through presentations and problem-solving games. Open only to first-year students in the Computing and Informatics Exploratory Track RAP. This course does not count toward any requirements for the CMPSCI major or minor. 1 credit.		
<b>COMPSCI 191CMPS1</b>	<b>CS Major First Year Seminar</b>	<b>CREDITS 1</b>
<i>Richard Freedman</i>		
An exploration of computer science for first-year CMPSCI majors, focusing on a single topic. Sections 1 and 3 demonstrate computing principles through the history of video games. Sections 2 and 4 investigate the history of computing and its impact on the modern economy. Section 5 explores affective computing, which addresses systems and methods for emotion recognition, emotion expression in virtual agents and robots, and emotion modeling and affective agent architectures. Each section also looks at the ethical responsibilities of computing professionals and at civility and diversity in the world of computing. 1 credit		
<b>COMPSCI 191CMPS2</b>	<b>Computer Science Majors RAP Seminar</b>	<b>CREDITS 1</b>
<i>David Barrington</i>		
A weekly seminar for students in the CMPSCI RAP residential program. Exploration of computer science topics through presentations and problem-solving games. Mandatory pass/fail. Open only to first-year students in the CMPSCI RAP. This course does not count toward any requirements for the CMPSCI major or minor. 1 credit.		
<b>COMPSCI 197B</b>	<b>Special Topics - Advanced Assignments for COMPSCI 121</b>	<b>CREDITS 1</b>
<i>Gordon Anderson</i>		
This add on to COMPSCI 121 consists entirely of five additional more advanced programs that students write to extend the coding experience available in the base 121 class. The course is intended primarily for students in 121 who have some previous programming experience. Must be enrolled in COMPSCI 121 concurrently. 1 credit.		
<b>COMPSCI 197C</b>	<b>Special Topics - Programming in C</b>	<b>CREDITS 1</b>
<i>Nicolas Scarrci</i>		
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 COMPSCI 230 and courses that use C and C++. Prerequisites: COMPSCI 121 and 187. Runs for 6 weeks. This course is for CMPSCI minors and majors only, but it does not count towards either degree. 1 credit.		
<b>COMPSCI 197U</b>	<b>Special Topics - A Hands-on Introduction to UNIX</b>	<b>CREDITS 1</b>
<i>Tian Guo</i>		
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.		
<b>COMPSCI 197WP</b>	<b>Special Topics - Introduction to HTML/CSS</b>	<b>CREDITS 1</b>
<i>Nicolas Scarrci</i>		
An academic approach to building websites with html and css. This course is aimed at beginning programmers and self-taught web developers who would like to cement their skills. Topics include structural html5, css, and introductory javascript/jquery (if time permits). Prerequisite: COMPSCI 121. Mandatory P/F. Students who are enrolled in or have taken COMPSCI 326 are not eligible to take this course. 1 credit.		

<b>COMPSCI 220</b>	<b>Programming Methodology</b>	<b>CREDITS 4</b>
<i>Timothy Richards</i>		
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: COMPSCI 187 or ECE 242. 4 credits.		
<b>COMPSCI 230</b>	<b>Computer Systems Principles</b>	<b>CREDITS 4</b>
<i>J. Eliot Moss</i>		
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. Prerequisites: COMPSCI 187 or ECE 242. 4 credits.		
<b>COMPSCI 240</b>	<b>Reasoning Under Uncertainty</b>	<b>CREDITS 4</b>
<i>Benjamin Marlin</i>		
Development of mathematical reasoning skills for problems that involve uncertainty. Each concept will be illustrated by real-world examples and demonstrated through in-class and homework exercises, some of which will involve Java programming. Counting and probability -- basic counting problems, probability definitions, mean, variance, binomial distribution, Markov and Chebyshev bounds. Probabilistic reasoning -- conditional probability and odds, Bayes' Law, Naive Bayes classifiers, Monte Carlo simulation. Markov chains, Markov decision processes, classical game theory, introduction to information theory. This is a core course for the new COMPSCI curriculum and may be used as a math elective for the old curriculum. Prerequisites: COMPSCI 187 (or ECE 242) and MATH 132 or consent of instructor. 4 credits.		
<b>COMPSCI 250</b>	<b>Introduction to Computation</b>	<b>CREDITS 4</b>
<i>David Barrington</i>		
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. Prerequisite: MATH 132 and COMPSCI 187 (or ECE 242). MATH 132 may be used as a co-requisite with permission of instructor. 4 credits.		
<b>COMPSCI 305</b>	<b>Social Issues in Computing</b>	<b>CREDITS 3</b>
<i>Michelle Trim</i>		
Using a range of different disciplinary perspectives we will explore various impacts of computers on modern society. This exploration will focus primarily on the social impacts of computers, with an emphasis on ethical concerns. Students will gain practice in several technical communication genres, public writing, and academic writing. Students will produce approximately 20-25 pages of polished writing over the course of the semester, along with two presentations. Writing opportunities will include career documents, instructions, software/hardware reviews, analytical essays, technical arguments and public/advocacy writing. Prerequisite: ENGLWRIT 112. 3 credits.		
<b>COMPSCI 311</b>	<b>Introduction to Algorithms</b>	<b>CREDITS 4</b>
<i>Ramesh Sitaraman</i>		
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 COMPSCI 250) is required. You should also be able to program in Java, C, or some other closely related language. Prerequisite: COMPSCI 250 or MATH 455. 4 credits.		
<b>COMPSCI 320</b>	<b>Introduction to Software Engineering (IE)</b>	<b>CREDITS 4</b>
<i>Gordon Anderson</i>		
In this course, students learn and gain practical experience with software engineering principles and techniques. The practical experience centers on a semester-long team project in which a software development project is carried through all the stages of the software life cycle. Topics in this course include requirements analysis, specification, design, abstraction, programming style, testing, maintenance, communication, teamwork, and software project management. Particular emphasis is placed on communication and negotiation skills and on designing and developing maintainable software. Use of computer required. Several written assignments, in-class presentations, exams, and a term project. This course satisfies the IE Requirement. Prerequisite: COMPSCI 220. 4 credits.		

**COMPSCI 325**                      **Usability**    **CREDITS**    **3**

*Eva Hudlicka*

In this course we examine the important problems in Usability, Human Computer Interaction, User Interfaces and Human Centered Computing. We will examine elements of HCI history, human information processing capabilities, HCI design, user interface prototyping methods and new applications and directions in HCI, including affective HCI. This is not a course on how to make better dialog boxes, but rather a much more thorough exploration of how humans interact with computers and how to evaluate the effectiveness of HCI designs. Some elementary programming, or the use of UI prototyping tools, will be required, but students without prior programming experience should feel right at home in this class. The course is available to any undergraduate student and is not limited to computer scientists. IT-minor students are especially encouraged to participate. The course emphasizes group projects and students will be required to work in teams. There is one midterm exam, numerous class presentations of on-going project work, and a final demo and presentation of the major class project. No prerequisites. 3 credits.

**COMPSCI 326**                      **Web Programming (IE)**    **CREDITS**    **3**

*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. This course satisfies the IE Requirement. Prerequisites: COMPSCI 220 or COMPSCI 230. 3 credits.

**COMPSCI 345**                      **Practice and Applications of Data Management**    **CREDITS**    **3**

*Alexandra Meliou*

Computing has become data-driven, and databases are now at the heart of commercial applications. The purpose of this course is to provide a comprehensive introduction to the use of data management systems within the context of various applications. Some of the covered topics include application-driven database design, schema refinement, implementation of basic transactions, data on the web, and data visualization. The class will alternate between lecture and practice, and the students will experience the covered topics through a semester-long collaborative mini-project. This course counts as a CS Elective toward the COMPSCI major (BA/BS). Students who have taken COMPSCI 445 are not eligible to take this course. Prerequisite: COMPSCI 187 (or ECE 242). 3 credits.

**COMPSCI 377**                      **Operating Systems**    **CREDITS**    **4**

*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: COMPSCI 230 with a grade of 'C' or better. 4 credits.

**COMPSCI 383**                      **Artificial Intelligence**    **CREDITS**    **3**

*Hava Siegelmann*

The Course explores key concepts of artificial intelligence, including state-space and heuristic search techniques, game playing, knowledge representation, automated planning, reasoning under uncertainty, decision theory and machine learning. We will examine how these concepts are applied in the context of several applications. Prerequisites: COMPSCI 220 (or COMPSCI 230) and COMPSCI 240. 3 credits.

<b>COMPSCI 390MB</b>	<b>Mobile Health Sensing and Monitoring</b>	<b>CREDITS 3</b>
<i>Deepak Ganesan</i>		
<p>In recent years, the ability to continuously monitor activities, health, and lifestyles of individuals using sensor technologies has reached unprecedented levels. The typical smartphone comes equipped with a plethora of sensors for monitoring activity, speech patterns, social interactions, and location. In addition, mobile accessories such as wearable wristbands and chestbands now enable routine and continuous monitoring of a host of physiological signals (e.g., heart rate, respiratory rate, skin Conductance, and others.). In conjunction, these sensors can enable higher-order inferences about more complex human activities/behavioral states (e.g., activity patterns, stress, sleep, etc.). Such ubiquitous sensing in daily life, referred to as mobile health sensing and monitoring, promises to revolutionize our understanding of human activities and health conditions. This course is an introduction to personal health sensing and monitoring through mobile phones and on-body sensors and addresses several aspects including mobile devices and applications for health, sensor data quality and reliability challenges, inference of key health assessments from sensor data including such as activity patterns, sleep patterns, or stress, sensor data visualization and feedback, and practical considerations such as battery lifetime. This course counts as a CS Elective toward the CMPSCI major (BA/BS). Prerequisite: CMPSCI 187 (or equivalent). 3 credits.</p>		
<b>COMPSCI 391H</b>	<b>Software Engineering with Honors</b>	<b>CREDITS 1</b>
<i>Leon Osterweil</i>		
<p>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.</p>		
<b>COMPSCI 391L</b>	<b>Seminar - Computer Crime Law and the Technologies of Investigation</b>	<b>CREDITS 3</b>
<i>Brian Levine</i>		
<p>A study, analysis, and discussion of the legal issues related to crimes involving computers and networks, including topical actions by dissidents and governments. We will also study the technologies of forensic investigation, intelligence gathering, privacy enhancement, and censorship resistance. Our main legal topics will include recent and important case law, statutes, and constitutional clauses concerning authorization, access, search and seizure, wiretaps, the right to privacy, and FISA. Our technology topics will include methods of investigation and resistance in the context of the Internet and Cellular networks. Students are assumed to have no background in legal concepts. Students will be required to complete substantial legal readings, complete significant written analysis of rulings, learn about technologies in detail, and participate in lively class discussion. Prerequisite: CMPSCI 230 and ENGLWRIT 112. This course counts as a CS Elective toward the CMPSCI major. 3 credits.</p>		
<b>COMPSCI 397J</b>	<b>Special Topics - Introductory Javascript Game Development</b>	<b>CREDITS 1</b>
<i>Colin Tincknell</i>		
<p>Students will be taught the basics of game development using a Javascript/HTML5 game engine. This course will be project based, with students implementing features onto simple existing games to familiarize themselves with the engine. After becoming comfortable with the engine, students will propose a game project and then develop it from scratch. Prerequisite: CMPSCI 326. Not for CS Major/Minor Requirements. 1 credit.</p>		
<b>COMPSCI 453</b>	<b>Computer Networks</b>	<b>CREDITS 3</b>
<i>John Ridgway</i>		
<p>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). 3 credits.</p>		
<b>COMPSCI 460</b>	<b>Introduction to Computer and Network Security</b>	<b>CREDITS 3</b>
<i>Amir Houmansadr</i>		
<p>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.</p>		
<b>COMPSCI 474</b>	<b>Image Synthesis</b>	<b>CREDITS 3</b>
<i>Rui Wang</i>		
<p>This course provides a broad overview of the theory and practice of image synthesis. Topics include photorealistic rendering, global illumination techniques, Monte Carlo ray tracing, texture synthesis, image editing and compositing. Previous experience with introductory computer graphics (such as CMPSCI 473) is preferred. Prerequisites are: CMPSCI 187 or equivalent, MATH 235. 3 credits.</p>		

**COMPSCI 496S**                      **Independent Study - Scalable Web Systems**                      **CREDITS 1**

*Timothy Richards*

Managing agile development of complex web systems is a difficult task. These applications require a broad range of conceptual understanding ranging from user interface design and client/server-side concepts and programming to caching, monitoring, scalability, and performance issues. These techniques need to work collectively to bring the end user a single working system. Unlike traditional waterfall methods of software development that demand a focus on design before development, agile development emphasizes techniques that require an iterative design, development, and deploy process. Test driven development is the focus and a working system is always available for review. This 1 credit course will require students to be managers that lead a software development team (students enrolled in 497S) to the successful completion and deployment of a real web application. Students enrolled in this course will be required to hold team meetings, manage work and design, and evaluate the individual work and contributions of students taking 497S and meet with the instructor to read and discuss agile software development and techniques from the literature. Prerequisites: COMPSCI 230 or 220 and 326. Co-requisites: COMPSCI 497S. 1 credit.

**COMPSCI 497S**                      **Special Topics - Scalable Web Systems**                      **CREDITS 3**

*Timothy Richards*

The web has become a large and complex area for application development. Access to an abundance of open source languages, libraries, and frameworks has led to the quick and easy construction of a variety of applications with several moving parts working in coordination to present to the user the illusion of a single program. In reality, web applications and services are extremely difficult to get right. They involve complicated user interfaces, multiple databases, security holes and performance issues, and a multitude of ever changing remote services, spread across several physical and virtual machines. These complications are further stressed by the large number of concurrent users that access these applications every second. This complexity leads to failed applications, performance bottlenecks, and security breaches. This course focuses on all areas of building scalable web systems. Topics include the study of cyber security, big data, system design, best practices, redundancy, scalability, performance and monitoring, and key principles in the agile development of complicated distributed web applications. Students will work as a team to design, develop, monitor, and analyze a large and scalable web application. Each student will focus on an area of interest and use what they have learned to contribute to the deployment of a real web application. This course counts as a CS Elective toward the CS major (BA/BS). Prerequisites: COMPSCI 230 or 220 and 326. 3 credits.

**COMPSCI 503**                      **Embedded Computing Systems**                      **CREDITS 3**

*Roderic Grupen*

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

**COMPSCI 513**                      **Logic In Computer Science**                      **CREDITS 3**

*Neil Immerman*

Rigorous introduction to mathematical logic from an algorithmic perspective. Topics include: Propositional logic: Horn clause satisfiability and SAT solvers; First Order Logic: soundness and completeness of resolution, compactness theorem. We will use the Coq theorem prover and Datalog. Prerequisites: COMPSCI 250 and COMPSCI 311. 3 credits.

**COMPSCI 521**                      **Software Engineering: Analysis and Evaluation**                      **CREDITS 3**

*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: COMPSCI 320, Introduction to Software Engineering (or equivalent course). 3 credits.

**COMPSCI 529**                      **Software Engineering Project Management**                      **CREDITS 3**  
*Gordon Anderson*

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 COMPSCI 320. As project managers, the students in COMPSCI 529 will be responsible for: supervising and managing the work of teams of COMPSCI 320 students; interfacing with the other COMPSCI 529 students managing other teams in the course; interfacing with the course instructor, course TA, and course customer. COMPSCI 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, COMPSCI 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. COMPSCI 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 COMPSCI 320. Additional meetings with team members and other students in COMPSCI 529 are also expected to be arranged by mutual agreement. An additional one hour weekly meeting of all of the students in COMPSCI 529 is required. Enrollment in this course is only by permission of the instructor, and is restricted to students who have previously taken COMPSCI 320, and received a grade of A or A-. 3 credits.

**COMPSCI 585**                      **Introduction to Natural Language Processing**                      **CREDITS 3**  
*Brendan O'Connor*

Natural Language Processing (NLP) is the engineering art and science of how to teach computers to understand human language. NLP is a type of artificial intelligence technology, and it's now ubiquitous -- NLP lets us talk to our phones, use the web to answer questions, map out discussions in books and social media, and even translate between human languages. Since language is rich, subtle, ambiguous, and very difficult for computers to understand, these systems can sometimes seem like magic -- but these are engineering problems we can tackle with data, math, machine learning, and insights from linguistics. This course will introduce NLP methods and applications including probabilistic language models, machine translation, and parsing algorithms for syntax and the deeper meaning of text. During the course, students will (1) learn and derive mathematical models and algorithms for NLP; (2) become familiar with basic facts about human language that motivate them, and help practitioners know what problems are possible to solve; and (3) complete a series of hands-on projects to implement, experiment with, and improve NLP models, gaining practical skills for natural language systems engineering. Prerequisites: COMPSCI 220 (or COMPSCI 230) and COMPSCI 240. An alternate prerequisite of LINGUIST 492B is acceptable for Linguistics majors. 3 credits.

**COMPSCI 590GC**                      **Three Dimensional Modeling and Simulation**                      **CREDITS 3**  
*Evangelos Kalogerakis*

The course will teach advanced algorithms and techniques for 3D geometric modeling, animation and physics-based simulation, which are all fundamental components of content creation software. The course will start by covering the most commonly used digital representations of shapes, such as polygon meshes, point clouds, NURBS surfaces, and subdivision surfaces. Then it will cover algorithms for surface scanning, reconstruction, registration, differential geometry, re-meshing, smoothing, texturing, parameterization, and Laplacian surface editing. The course will proceed with algorithms for sketch-based modeling and procedural modeling. Finally, the course will cover methods for character rigging and animation, physics-based simulation of rigid bodies, deformable solids, fluids and cloth. Students will be asked to complete programming assignments related to shape modeling, reconstruction and physics-based animation. For 690GC students, there will also be a final project that involves the implementation of a research idea related to 3D modeling or animation. There are no prerequisites for COMPSCI graduate students, although familiarity with computer graphics, linear algebra and ordinary differential equations is essential. For undergrads, the course has the following prerequisites: Introduction to Linear Algebra (Math 235, grade A- or better) and Introduction to Computer graphics (COMPSCI 390CG or 473, grade A- or better). This course counts as a CS Elective toward the COMPSCI major (BA/BS) and it also belongs to the computer graphics, vision, robotics track. 3 credits.

**COMPSCI 591CF**                      **Seminar - Cyber Security Faculty Lecture Series [ECE 591CF]**                      **CREDITS 1**  
*Brian Levine, Amir Houmansadr*

This course is a one-credit seminar on security research across departments at UMass. Each presentation will cover an active research topic at UMass in a way that assumes only a basic background in security. External speakers may also be invited. Note that this course is not intended to be an introduction to cybersecurity, and will not teach the fundamentals of security in a way that would be useful as a foundation for future security coursework. The intended audience is graduate and advanced undergraduate students, as well as faculty. Meets with ECE 591CF. May be taken for credit up to 2 times. This course does not count toward any requirements for the CS major or minor.

**COMPSCI 597N**                      **Special Topics - Introduction to Computer and Network Security**                      **CREDITS 3**  
*Amir Houmansadr*

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, and class projects and presentations. 3 credits.

- COMPSCI 611**                      **Advanced Algorithms**                      **CREDITS 3**  
*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 COMPSCI 311. 3 credits.
- COMPSCI 621**                      **Advanced Software Engineering: Analysis and Evaluation**                      **CREDITS 3**  
*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: COMPSCI 320, Introduction to Software Engineering (or equivalent course). 3 credits.
- COMPSCI 630**                      **Systems**                      **CREDITS 3**  
*Emery Berger*  
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.
- COMPSCI 653**                      **Computer Networking**                      **CREDITS 3**  
*Donald Towsley*  
This course covers advanced fundamental principles of computer networks, studying foundational material in the field. Topics include advanced network architecture, network algorithmics, network control, network measurement, and wireless networks. The goal of this course is to teach networking fundamentals/techniques that will be useful for years to come. Prerequisites: Introductory (undergraduate level) courses in computer networks (e.g., COMPSCI 453), and algorithms (e.g., COMPSCI 311). Some familiarity with probability and with optimization theory will be helpful. 3 credits.
- COMPSCI 683**                      **Artificial Intelligence**                      **CREDITS 3**  
\*\* DELETED  
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.
- COMPSCI 689**                      **Machine Learning**                      **CREDITS 3**  
*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.



<b>COMPSCI 690GC</b>	<b>Three Dimensional Modeling and Simulation</b>	<b>CREDITS 3</b>
<i>Evangelos Kalogerakis</i>		
<p>The course will teach advanced algorithms and techniques for 3D geometric modeling, animation and physics-based simulation, which are all fundamental components of content creation software. The course will start by covering the most commonly used digital representations of shapes, such as polygon meshes, point clouds, NURBS surfaces, and subdivision surfaces. Then it will cover algorithms for surface scanning, reconstruction, registration, differential geometry, re-meshing, smoothing, texturing, parameterization, and Laplacian surface editing. The course will proceed with algorithms for sketch-based modeling and procedural modeling. Finally, the course will cover methods for character rigging and animation, physics-based simulation of rigid bodies, deformable solids, fluids and cloth. Students will be asked to complete programming assignments related to shape modeling, reconstruction and physics-based animation. For 690GC students, there will also be a final project that involves the implementation of a research idea related to 3D modeling or animation. There are no prerequisites for COMPSCI graduate students, although familiarity with computer graphics, linear algebra and ordinary differential equations is essential. For undergrads, the course has the following prerequisites: Introduction to Linear Algebra (Math 235, grade A- or better) and Introduction to Computer graphics (COMPSCI 390CG or 473, grade A- or better). 3 credits.</p>		
<b>COMPSCI 690LG</b>	<b>Advanced Logic in Computer Science</b>	<b>CREDITS 3</b>
<i>Neil Immerman</i>		
<p>Rigorous introduction to mathematical logic from an algorithmic perspective. Topics include: Propositional logic: Horn clause satisfiability and SAT solvers; First Order Logic: soundness and completeness of resolution, compactness theorem, automatic theorem proving, model checking. We will learn about and use the Coq theorem prover, Datalog, a Model Checker, and SAT and SMT solvers. Prerequisites: Students taking this course should have undergraduate preparation in discrete math and algorithms. Requirements will include readings, class participation, weekly problem sets, a midterm and a final project. 3 credits.</p>		
<b>COMPSCI 691NN</b>	<b>Seminar - Hot Topics in Programming Languages and Systems</b>	<b>CREDITS 1-3</b>
<i>Emery Berger</i>		
<p>This graduate seminar course will cover recent developments in programming languages and systems, examining the latest research papers from top programming languages and systems conferences. Topics of interest include bug detection and correction, domain-specific languages, and emerging topics, including probabilistic and approximate computing. 3 credits with project, 1 credit without.</p>		
<b>COMPSCI 691O</b>	<b>Seminar - Tools for Explanatory and Tutoring Systems</b>	<b>CREDITS 3</b>
<i>Beverly Woolf</i>		
<p>This seminar examines recent work in explanatory and tutoring systems. Participants study <i>artificial intelligence in education</i>, 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.</p>		
<b>COMPSCI 697L</b>	<b>Seminar - Deep Learning</b>	<b>CREDITS 3</b>
<i>Sridhar Mahadevan</i>		
<p>Deep learning has been termed one of the leading scientific breakthroughs in recent years (MIT Technology Review). It has attracted significant industrial investment with large groups at Google, Facebook, Baidu, IBM, Microsoft etc. working on a plethora of commercial applications of deep learning. It has resulted in state of the art performance in a variety of areas, including computer vision, natural language processing, reinforcement learning, and speech recognition. This course is intended to provide students with an in-depth introduction to both theory and practice of deep learning. Topics: historical overview of neural networks and relevant ideas from brain modeling in neuroscience; classic architectures, including perceptron and backpropagation; deep learning models, including restricted Boltzmann machines, autoencoders, and convolutional neural networks. Stochastic gradient methods for training deep learning models, including Adagrad, Nesterov's method etc.; Overview of software packages for building deep learning models, including Caffe, Pylearn, Torch/Lua, and Theano. Applications of deep learning to computer vision, natural language processing, reinforcement learning, robotics, and speech recognition; Recent insights from statistical physics and high-dimensional non-convex optimization to explain deep learning. Workload: weekly reading of articles from the literature on deep learning. Final group project exploring some application of deep learning. Class presentation and participation. 3 credits.</p>		
<b>COMPSCI 701</b>	<b>Advanced Topics in Computer Science</b>	<b>CREDITS 6</b>
<i>STAFF</i>		
<p>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.</p>		

**COMPSCI 745****Advanced Topics in Database Systems****CREDITS 3****\*\* DELETED**

This course covers advanced data management systems, including high-performance database systems and recent big data systems. The course first covers the design and implementation of high-performance database systems including data warehouses, decision support systems, column stores, and parallel databases. It also covers advanced databases to support new data models, including temporal databases, sequence databases, and data stream systems. Then a special focus of the course will be on recent big data systems, including MapReduce-based parallel processing, bigtable-style storage systems, and many new processing algorithms and techniques developed in industry and academia in this setting. The prerequisite is a graduate course on the principles and implementations of traditional database systems, an equivalent of COMPSCI 645. Students with other backgrounds are asked to contact the instructor for approval for enrollment. 3 credits.

**COMPSCI 891M****Theory of Computation****CREDITS 1***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.

**COMPSCI FFYS 197CIC1 FFYS - Artificial Intelligence and Society****CREDITS 1***Shlomo Zilberstein*

How is progress in artificial intelligence affecting society? Artificial intelligence (AI) is fueling the development of autonomous systems that have already changed the way space exploration is conducted and is beginning to transform everyday life with a range of household products. Such systems offer transformational impact on society as they help reduce human labor and risks, while improving productivity and efficiency. Meanwhile, some scientists such as Stephen Hawking warn that thinking machines pose a threat to our very existence. This class covers the major accomplishments of AI, what AI systems can do today and how they do it. Students will explore the potential and limitations of AI, the impact of AI technology on society, and the ethical questions that are raised by these technological developments. Mandatory Pass/Fail. 1 credit.