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. Not for CMPSCI majors. 3 credits.

CMPSCI 120  Introduction to Problem Solving with the Internet (R2)  
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 a basic introduction to client-side web programming in HTML, CSS and Javascript. In addition to static and dynamic web page and web site design and implementation, we will cover timely strategies for finding information, participating in virtual communities, managing email, and ensuring privacy. We will survey key social and political topics that are relevant to the Internet such as copyright laws, First Amendment issues, and public key cryptography. Prerequisites: some hands-on experience with PCs or MACs or UNIX (programming experience is NOT required). Not for CMPSCI majors. 3 credits.

CMPSCI 121  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 187  Programming with Data Structures (R2)  
David Barrington
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, David A. Smith
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  
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.
A brief introduction to the C++ programming language for students with a good working knowledge of Java. This course is good preparation for CMPSCI 291SP and courses that use C++. Prerequisites: CMPSCI 121 and 187 or permission of instructor. Runs for 8 weeks. This course is for CMPSCI minors and majors only, but it does not count towards either degree. 1 credit.

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.

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

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.

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.
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**  
David Fisher

In this course, students learn and gain practical experience with software engineering principles and techniques. The practical experience centers on a semester-long team project in which a software development project is carried through all the stages of the software life cycle. Topics in this course include requirements analysis, specification, design, abstraction, programming style, testing, maintenance, communication, teamwork, and software project management. Particular emphasis is placed on communication and negotiation skills and on designing and developing maintainable software. Use of computer required. Several written assignments and in-class presentations, two exams, and a major term project. Prerequisite: CMPSCI 220 or CMPSCI 287. 4 credits.

**CMPSCI 325**  
**Usability**  
Samuel Dunn, Zachary Dunn

In this course we examine the important problems in Usability, Human Computer Interaction, User Interfaces, and Human Centered Computing. We will examine elements of HCI history, understanding human capabilities, HCI design, several methods for prototyping user interfaces, and new applications and paradigms in human computer interaction. This is not a course in how to make dialog boxes, but rather a much more general approach to interacting with human beings and evaluating designs. Some elementary programming 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 383**  
**Artificial Intelligence**  
Andrew Barto

The Course explores key concepts of artificial intelligence, including problem solving, state-space representation, heuristic search techniques, game playing, knowledge representation, logical reasoning, automated planning, reasoning under uncertainty, decision theory and machine learning. We will examine how these concepts are applied in the context of several applications. Prerequisites: (CMPSCI 287 or CMPSCI 220 or CMPSCI 291SP) and (CMPSCI 240 or CMPSCI 311). 3 credits.

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

How does the computer actually work? In this course we peel away the layers of abstraction and look at how switches become logic circuits, how logic circuits do math, and how programs really execute. We will wire up some simple examples of logic, move on to programming a minimalist simulation of a computer, and gradually build up to the point where we can appreciate how a C program is translated into machine code and what really happens when it executes. We will also see the impact of hidden acceleration mechanisms like caches, pipelines, and branch predictors. This course is equivalent to CmpSci 535 and is approved as a CS elective for CMPSCI major. Prerequisite: CmpSci 220 or equivalent. 3 credits.
In this course, we study and discuss legal issues related to crimes involving computers and networks. Our main topics will include recent and important case law, statutes, and constitutional clauses concerning authorization, access, vice crimes, search and seizure, wiretaps, the right to privacy, FISA, and jurisdiction. Students are assumed to be familiar with general computing concepts and applications, but the instructor will provide an introduction to legal concepts. Students will be required to complete substantial readings, complete significant written analysis of rulings, and participate in lively class discussion. Prerequisite: ENGLWRIT 112. This course counts as a CMPSCI Elective toward the CMPSCI major. 3 credits.

** CMPSCI 391LI **
Seminar - Computer Crime Law
Brian Levine, Marc Liberatore
In this course, we study and discuss legal issues related to crimes involving computers and networks. Our main topics will include recent and important case law, statutes, and constitutional clauses concerning authorization, access, vice crimes, search and seizure, wiretaps, the right to privacy, FISA, and jurisdiction. Students are assumed to be familiar with general computing concepts and applications, but the instructor will provide an introduction to legal concepts. Students will be required to complete substantial readings, complete significant written analysis of rulings, and participate in lively class discussion. Prerequisite: ENGLWRIT 112. 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
Timothy Richards
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
Honors Colloq
This course provides an introduction to fundamental concepts in computer networks, including their design and implementation. Topics covered include the Web and other applications, transport protocols (providing reliability and congestion control), routing, and link access. Special attention is also paid to wireless networks and security. Homework assignments involve programming and written tasks. Prerequisites: Experience programming; CMPSCI 291SP (or CMPSCI 377) and CMPSCI 240 (or STATS 515). 3 credits.

** CMPSCI 460 **
Introduction to Computer and Network Security
Christopher Misra
This course provides an introduction to the principles and practice of system and network security with a focus on both fundamentals and practical information. The three key topics of this course are cryptography, system and network security, and protecting information technology resources. Subtopics include ciphers, key exchange, security services (integrity, availability, confidentiality, etc.), system, network, and web based security attacks, vulnerabilities, exploits, countermeasures and responses. Students will complete several lab assignments experimenting with security countermeasures. Grades will be determined by class participation, lab work, homework, quizzes and exams. Prerequisites include CMPSCI 377 and a familiarity with Unix. 3 credits.

** CMPSCI 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 491LP **
Seminar - Introduction to Natural Language Processing
** DELETED **
Course Description not on file. 3 credits.

** CMPSCI 491OO **
Seminar - Object Oriented Languages
** DELETED **
The goal of the course is to give students an advanced background for both technical work and research involving object-oriented languages and systems. The course covers topics such as advanced Java/C++ language mechanisms, design patterns critique, aspect-oriented programming, OO frameworks, mixins, OO language implementation (compiler optimizations, virtual machine principles), type systems and genericity. This course counts as a CS Elective toward the CMPSCI major. 3 credits.
CMPSCI 503  Embedded Computing Systems  
Roderic Grupen

Embedded computing applications do not require the same specifications as general purpose computational platforms, but are instead intimately related to a particular physical process. This course introduces tools necessary to design embedded computational applications. We will explore these applications by building autonomous mobile robots from scratch to accomplish a task that the class will design. Each student will design and build a project as part of a team that will be demonstrated at the end of the semester. The course is heavily project-oriented (with a required lab) and discussions will include topics such as: (1) mechanisms, sensors, actuators and feedback systems, (2) analog and digital circuits, power amplifiers, signal processing, operational amplifiers, multiplexing, (3) I/O - A/D, D/A, and latching, serial and parallel interfaces, (4) signal processing/conditioning and (5) an introduction to real-time programming. 3 credits.

CMPSCI 513  Logic In Computer Science  
Neil Immerman

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

CMPSCI 521  Software Engineering: Analysis and Evaluation  
Leon Osterweil

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). No required text. Papers from the open literature will be assigned and made available. 3 credits.

CMPSCI 529  Software Engineering Project Management  
David Fisher

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

CMPSCI 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  
Ramesh Sitaraman

Principles underlying the design and analysis of efficient algorithms. Topics to be covered include: divide-and-conquer algorithms, graph algorithms, matroids and greedy algorithms, randomized algorithms, NP-completeness, approximation algorithms, linear programming. Prerequisites: The mathematical maturity expected of incoming Computer Science graduate students, knowledge of algorithms at the level of CMPSCI 311. 3 credits.
Software 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 683  Artificial Intelligence**
Shlomo Zilberstein
In-depth introduction to Artificial Intelligence focusing on techniques that allow intelligent systems to reason effectively with uncertain information and cope limited computational resources. Topics include: problem-solving using search, heuristic search techniques, constraint satisfaction, local search, abstraction and hierarchical search, resource-bounded search techniques, principles of knowledge representation and reasoning, logical inference, reasoning under uncertainty, belief networks, decision theoretic reasoning, representing and reasoning about preferences, planning under uncertainty using Markov decision processes, multi-agent systems, and computational models of bounded rationality. 3 credits.

**CMPSCI 689  Machine Learning**
Sridhar Mahadevan
Machine learning is the computational study of artificial systems that can adapt to novel situations, discover patterns from data, and improve performance with practice. This course will cover the popular frameworks for learning, including supervised learning, reinforcement learning, and unsupervised learning. The course will provide a state-of-the-art overview of the field, emphasizing the core statistical foundations. Detailed course topics: overview of different learning frameworks such as supervised learning, reinforcement learning, and unsupervised learning; mathematical foundations of statistical estimation; maximum likelihood and maximum a posteriori (MAP) estimation; missing data and expectation maximization (EM); graphical models including mixture models, hidden-Markov models; logistic regression and generalized linear models; maximum entropy and undirected graphical models; nonparametric models including nearest neighbor methods and kernel-based methods; dimensionality reduction methods (PCA and LDA); computational learning theory and VC-dimension; reinforcement learning; state-of-the-art applications including bioinformatics, information retrieval, robotics, sensor networks and vision. Prerequisites: undergraduate level probability and statistics, linear algebra, calculus, AI; computer programming in some high level language. 3 credits.

**CMPSCI 691A  Seminar - Computer Vision**
Erik Learned-Miller
People are able to infer the characteristics of a scene or object from an image of it. In this course, we will study what is involved in building artificial systems which try to infer such characteristics from an image. Topics include: Basics of image formation - the effect of geometry, viewpoint, lighting and albedo on image formation. Basic image operations such as filtering, convolution and correlation. Frequency representations of images. The importance of scale in images. Measurements of image properties such as color, texture, appearance and shape. Inference of motion and structure from moving objects and images. Detecting and recognizing objects in images. Statistical methods in computer vision. Prerequisite: Linear algebra, calculus, basic knowledge of probability, and an ability to program. 3 credits.

**CMPSCI 691NW  Seminar - Introduction to Network Science**
Donald Towsley, Bruno Ribeiro
This course will introduce and develop the mathematical theory of networks, particularly social and technological networks. Applications will be made to search, identifying influential nodes, network resilience, epidemiology, and many other areas. Topics to be covered will include experimental studies of social networks, the world wide web, information and biological networks; methodologies for the analysis of network data; graph theory; probabilistic models of networks, preferential attachment models, and the small-world model; network dynamics. Students should have had courses in calculus, linear algebra, and probability. In particular, students should be comfortable with setting up and numerically solving ordinary differential equations, the calculation of eigenvalues and eigenvectors of matrices, and dealing with finite state Markov chains. Student’s grades will be determined by assignments given every 1-2 weeks, a midterm, and a final exam. The required textbook is *Networks: An Introduction*, M. E. J. Newman, Oxford University Press, Oxford (2010). 3 credits.
This seminar examines recent work in explanatory and tutoring systems. Participants study artificial intelligence in education, a young field that explores theories about learning, and explores how to build software that delivers differential teaching as it adapts its response to student needs and domain knowledge. Such software supports people who work alone or in collaborative inquiry, students who question their own knowledge, and students who rapidly access and integrate global information. This course describes how to build these tutors and how to produce the best possible learning environment, whether for classroom instruction or lifelong learning. The objective of the course is to stimulate awareness of research issues and to promote sound analytic and design skills as they pertain to building knowledge representations and control strategies. Specific topics include collaboration, inquiry, dialogue systems, machine learning, simulators, authoring tools and user models. The course is appropriate for students from many disciplines (computer science, linguistics, education, and psychology), researchers, and practitioners from academia, industry, and government. No programming is required. Students read and critique tools, methods, and ideas, learn how artificial intelligence is applied (e.g., vision, natural language), and study the complexity of human learning through advances in cognitive science. Weekly assignments invite students to critique the literature and a final project requires a detailed specification (not a program) for a tutor about a chosen topic. Students present book chapter material weekly in parallel with readings from the research literature. Several working systems will be available for hands-on critique. 3 credits.

** DELETED**

The goal of the course is to give students an advanced background for both technical work and research involving object-oriented languages and systems. The course covers topics such as advanced Java/C++ language mechanisms, design patterns critique, aspect-oriented programming, OO frameworks, mixins, OO language implementation (compiler optimizations, virtual machine principles), type systems and genericity. 3 credits.

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. 3 credits.

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.

Good decision-making is dependent on comprehensive, accurate knowledge. But the information relevant to many important decisions in areas such as business, government, medicine and scientific research is massive, and growing at an accelerating pace. Relevant raw data is widely available on the web and other data sources, but usually in order to be useful it must be gathered, extracted, organized, and normalized into a queryable, minable knowledge base. Hand-built knowledge bases such as Wikipedia have proven useful, but more than human editing will be necessary to create a wide variety of domain-specific, deeply comprehensive, highly-structured knowledge bases. Various automated methods have begun to reach levels of accuracy and scalability that make them applicable to automatically constructing useful knowledge bases from text and other sources. These capabilities have been enabled by research in areas including natural language processing, information extraction, information integration, databases, search and machine learning. In this seminar we will relevant read papers in all these areas, write responses to them, and discuss them. Students will also have the option to join a team of volunteers who will build a system that constructs a KB of all UMass faculty, postdocs and graduate students, and strives to predict students' year of PhD completion. 3 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.
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.

**Honors Colloquium for CMPSCI 250**

*Stephen Constantine*

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

**Honors Colloquium for CMPSCI 320**

*David Fisher*

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

**Honors Colloquium for CMPSCI 453**

*James Kurose*

Students enrolled in the honors section of CMPSCI 453 will meet for an additional hour per week with the instructor to discuss a project in advanced topics in Computer Networks including wireless networks and sensor networks. The project will involve hands-on exposure in designing, deploying and measuring outdoor wireless sensor networks. 1 credit.