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
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)
Wendy Lehnerw
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 191P  Introduction to Programming
Wendy Lehnerw
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.

CMPSCI 197C  Special Topics - Programming in C++
Marc Cartright
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.
CMPSCI 197U Special Topics - A Hands-on Introduction to UNIX
Jacqueline Feild
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 CMPSCI minors and majors only, but it does not count towards either degree. 1 credit.

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

CMPSCI 240 Reasoning About Uncertainty
Hava Siegelmann
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

CMPSCI 291SP Seminar - Computer Systems Principles
Mark Corner
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 successful 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.

CMPSCI 291SR Seminar - Computer Systems Programming
Mark Corner
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
Lori Clarke
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 370  Introduction to Computer Vision
Erik Learned-Miller
This introductory computer vision class will address fundamental questions about getting computers to "see" like humans. We
investigate questions such as -What is the role of vision in intelligence? -How are images represented in a computer? -How can we
write algorithms to recognize an object? -How can humans and computers "learn to see better" from experience? We will write a
number of basic computer programs to do things like recognize handwritten characters, track objects in video, and understand the
structure of images. Prerequisite: CMPSCI 240 or 383.  3 credits.

CMPSCI 377  Operating Systems
Deepak Ganesan
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 (CMPSCI 291SP or ECE 232).
Strongly recommend CMPSCI 220 with a 'C' or better prior to taking this course.  4 credits.

CMPSCI 383  Artificial Intelligence
Shlomo Zilberstein
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 391F  Seminar - Fundamentals of Graphic Communication
Beverly Woolf, Tony Sindelar
The purpose of this course is to provide students with an understanding of computer-based design and multimedia production. The
course will cover basic concepts in the areas of graphic design, animation, and designing for interactivity. These concepts will help
prepare students for more advanced studies in 2D and 3D animation and in multimedia production and programming. Course work
will be based around several large ongoing projects built incrementally as individual homework assignments. These projects will be
presented via the World Wide Web and should demonstrate an understanding of techniques and principles learned in earlier projects.
Student interaction and collaboration will be encouraged through in-class activities as well as online discussions through SPARK. This
course does not count toward the CMPSCI major or minor.  3 credits.
Introduction to formal language theory. Topics include finite state languages, context-free languages, the relationship between language classes and formal machine models, the Turing Machine model of computation, theories of computability, resource-bounded models, and NP-completeness. Prerequisites: CMPSCI 311 or equivalent. It is recommended that students have a 'B-' or better in 311 in order to attempt 401. 3 credits.

This course covers basic methods and concepts in order to explain how robots work. We will study how they sense things in the world, how you make a robot move, and how robots can make their own decisions. We will study mechanisms (kinematics and dynamics), actuators, sensors, (with a focus on machine vision), signal processing, feedback control theory, machine learning, and path planning. Students will build software systems to control a simulated mobile manipulator and do experiments with system identification and control, image processing, and path planning to reinforce the material presented in class. Prerequisites: Math 235 (or Math 236) and CMPSCI 220 (or CMPSCI 287 or CMPSCI 291SP). 3 credits.

This course provides an overview of the important issues in information retrieval, and how those issues affect the design and implementation of search engines. The course emphasizes the technology used in Web search engines, and the information retrieval theories and concepts that underlie all search applications. Mathematical experience (as provided by CMPSCI 240) is required. You should also be able to program in Java (or some other closely related language). This course may be used in place of CMPSCI 445 as a semi-required course under the OLD rules for the BS. Prerequisite: CMPSCI 240 or CMPSCI 383, or equivalent. 3 credits.

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 (or CMPSCI 291SP), CMPSCI 311, and CMPSCI 377. 3 credits.

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.

This course introduces to commonwealth honors completing department capstones basic ideas about conducting a personal research program. Students will learn basic methods for activities such as reading technical papers, selecting research topics, devising research questions, planning research, analyzing experimental results, modeling and simulating computational phenomena, and synthesizing broader theories. The course will be structured around three activities: lectures on basic concepts of research strategy and techniques, discussions of technical papers, and preparation and review of written assignments, all centered on the student's specific capstone project. Significant reading, reviewing, and writing will be required, and students will be expected to participate actively in class discussions. 3 credits.

In this course, students will learn how to put "principles into practice," in a hands-on-networking lab course. The course will cover router, switches and end-system labs in the areas of Single Segment IP Networks, Multiple Segment IP Networks and Static Routing, Dynamic Routing Protocols (RIP, OSPF and BGP), LAN switching, Transport Layer Protocols: UDP and TCP, NAT, DHCP, DNS, and SNMP. Students will also get engaged in evaluating power consumption of network components as an aid in the design of energy efficient (green) networks. The labs are due at a rate of roughly one lab per week. A short pre-lab Q&A, as well as lab writeups, are required for each lab. These labs will be done in a networked lab setting consisting of 4 Cisco2600-family routers, 4 hubs, and 4 Linux hosts. The textbook is Mastering Networks: An Internet Lab Manual by Jorg Lieberherr; Magda El Zarki. ISBN: 0-201-78134-4. Publisher: Addison-Wesley. This course counts as a CS Elective toward the CMPSCI major. Prerequisite: CMPSCI 453. 3 credits.
The iPad, iPhone, and iPod Touch present examples of a mobile processor with an interesting set of peripheral devices and limitations. They are programmed using Objective C and a large set of APIs. This seminar will be an introduction to the Objective C language and iOS development environment, and will include student programming assignments and development projects targeting the Apple iPhone/iPad simulator. We will spend much of the class time actually working through examples, and discovering how some of the features really work. Students will need to have access to an Intel machine running OSX, because the development environment and simulator work only on that platform. Students also must join the Apple Developer Program, and accept their license agreements in order to obtain and use the necessary software. This course counts as a CS Elective toward the CMPSCI major. 3 credits.

CMPSCI 520 Software Engineering: Synthesis and Development
Leon Osterweil
Introduces students to the principal activities involved in developing high-quality software systems in a variety of application domains. Topics include: requirements analysis, formal and informal specification methods, process definition, software design, testing, and risk management. The course will pay particular attention to differences in software development approaches in different contexts. Prerequisites: CMPSCI 320 with a grade of ‘C’ or better, or E&C-ENG 373 with a grade of ‘C’ or better. 3 credits.

CMPSCI 529 Software Engineering Project Management
Lori Clarke
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 601 Computation Theory
Neil Immerman
An in-depth introduction to the main models and concepts of the theory of computation, including: Computability: what problems can be solved in principle; Complexity: what problems can be solved in a given amount of time, space, parallel time; Logic: how do formal specification and proof mirror other forms of computation? Students will learn to go from a concrete problem to a mathematical model; and, after proving things about the mathematical model, to correctly interpret what they have learned about the concrete problem. Prerequisites: Undergraduate-level courses in discrete mathematics (CMPSCI 250) and algorithms (CMPSCI 311). Additional mathematical maturity (e.g., CMPSCI 611, A’s in 250 or 311, or other mathematical background) quite desirable as is background in automat theory and formal languages (CMPSCI 401). Course requirements: biweekly problem sets, midterm and final. Also open to qualified undergraduates. 3 credits.

CMPSCI 603 Robotics
Roderic Grupen
This course is designed to be a advanced course in robotics that covers mechanisms (kinematics and dynamics), actuators, sensors, signal processing, feedback control, and signal processing. The target is to provide an understanding of robot systems that interact with, interpret feedback from, and manipulate the world about them. We will relate the subject matter to biological systems whenever possible, including discussion about the relationships between learning and development in human beings and what it has to say about programming robots. Students will experiment with system identification and control, image processing, path planning, grasping, and machine learning to reinforce the material covered in class. 3 credits.
This course examines the varied approaches to the development of computer software. We examine various ideas about how software products should be structured and function. We then examine how software processes serve as vehicles for manufacturing such products. This approach facilitates the direct study of different software development approaches, and a more direct study of their effects on the products they produce. This approach will be used by students, who will examine representative current software development product and process approaches in in-class presentations and written project papers as part of their coursework. 3 credits.

**CMPSCI 635**  Modern Computer Architecture

*Charles Weems*

This course examines the structure of modern computer systems. We explore hardware and technology trends that have led to current machine organizations, then consider specific features and their impact on software and performance. These may include superscalar issue, caches, pipelines, branch prediction, and parallelism. Midterm and final exams, individual projects, homework, in-class exercises. Prerequisites: CMPSCI 535 or equivalent. 3 credits.

**CMPSCI 645**  Database Design and Implementation

*Yanlei Diao*

This course covers the design and implementation of traditional relational database systems and advanced data management systems. The course will treat fundamental principles of databases: the relational model, conceptual design, query languages, and selected theoretical topics. We also cover core database implementation issues including storage and indexing, query processing and optimization, as well as transaction management, concurrency, and recovery. Additional topics will address the challenges of modern Internet-based data management. These include XML data management, information integration, incomplete and probabilistic databases, and database security. 3 credits.

**CMPSCI 660**  Advanced Information Assurance

*Brian Levine, Benjamin Ransford*

This course provides an in-depth examination of the fundamental principles of information assurance. While the companion course for undergraduates is focused on practical issues, the syllabus of this course is influenced strictly by the latest research. We will cover a range of topics, including authentication, integrity, confidentiality of distributed systems, network security, malware, privacy, intrusion detection, intellectual property protection, and more. Prerequisites: CMPSCI 460 or 466, or equivalent. 3 credits.

**CMPSCI 677**  Operating Systems

*Prashant Shenoy*

This course provides an in-depth examination of the principles of distributed systems in general, and distributed operating systems in particular. Covered topics include processes and threads, concurrent programming, distributed interprocess communication, distributed process scheduling, virtualization, distributed file systems, security in distributed systems, distributed middleware and applications such as the web and peer-to-peer systems. Some coverage of operating system principles for multiprocessors will also be included. A brief overview of advanced topics such as multimedia operating systems and mobile computing will be provided, time permitting. Prerequisites: Students should be able to easily program in a high-level language such as C, have had a course on data structures, be familiar with elements of computer architecture and have had previous exposure to the operating system concepts of processes, virtual memory, and scheduling. A previous course on uniprocessor operating systems (e.g., CMPSCI 377) will be helpful but not required. 3 credits.

**CMPSCI 687**  Reinforcement Learning

*Sridhar Mahadevan*

This course will provide a comprehensive overview of the use of modern machine learning, optimization, and statistical methodologies to model and control stochastic dynamical systems. Applications include activity modeling, game playing, industrial optimization, and robotics. A broad range of modeling paradigms will be explored, including completely and partially observable stochastic models, factored representations, and extensions to hierarchical decision-making models. A variety of algorithmic approaches will be investigated, including classical dynamic programming, approximate linear programming, Monte-Carlo sampling, and policy gradient methods. A particular focus of the course will be on constructing novel representations of fully and partially observable Markov decision processes. Insights from other fields such as biology on individual and group decision-making will be discussed. Prerequisites: 689 (Machine Learning); 683 (Artificial Intelligence) or 383 (undergraduate AI); knowledge of probability and statistics (such as covered in STATISTICS 607 or 608 or 515). 3 credits.
This course introduces graduate students to basic ideas about conducting a personal research program. Students will learn basic methods for activities such as reading technical papers, selecting research topics, devising research questions, planning research, analyzing experimental results, modeling and simulating computational phenomena, and synthesizing broader theories. The course will be structured around three activities: lectures on basic concepts of research strategy and techniques, discussions of technical papers, and preparation and review of written assignments. Significant reading, reviewing, and writing will be required, and students will be expected to participate actively in class discussions. 3 credits.

**Seminar - Research Methods in Empirical Computer Science**

David Jensen

This course introduces graduate students to basic ideas about conducting a personal research program. Students will learn basic methods for activities such as reading technical papers, selecting research topics, devising research questions, planning research, analyzing experimental results, modeling and simulating computational phenomena, and synthesizing broader theories. The course will be structured around three activities: lectures on basic concepts of research strategy and techniques, discussions of technical papers, and preparation and review of written assignments. Significant reading, reviewing, and writing will be required, and students will be expected to participate actively in class discussions. 3 credits.

**Seminar - Principles of Data Privacy**

Gerome Miklau

This graduate seminar will investigate the question: how can we learn aggregate facts about a population of people without violating the privacy of any one individual? The course will briefly review societal interpretations of privacy and then focus on recent definitions of privacy proposed in the data mining and database communities. We will explore syntactic notions of privacy, in which aggregation and suppression of data is used to protect it, and we will review the limitations of these techniques. We will also cover semantic notions of privacy, such as differential privacy, in which the answers to questions about the data are perturbed to protect individual privacy. Throughout the course we will seek to understand the limits of how accurately data mining and analysis can be performed. Course assignments will involve statistical reasoning to prove properties of privacy mechanisms as well as implementation of privacy mechanisms to evaluate their effectiveness on real data sets. The course is open to graduate students in Computer Science and related fields. Other students may be enrolled by permission of the instructor. Lect A=3 credits; Lect B=1 credit.

**Seminar - Graphical Models**

Andrew McCallum

Graphical models--the marriage of probability theory and graph theory--have become the lingua franca for describing solutions to a wide array of problems, ranging from computer vision to sensor networks, natural language processing to computational biology. The essence of their power is that they enable the compact representation and manipulation of the exponentially large probability distributions that are required to represent the uncertainty and partial observability that occur in so many real-world problems. This course will cover (a) representation, including Bayesian and Markov networks, dynamic Bayesian networks and relational models, (b) inference, both exact and approximate, and (c) estimation of both parameters and structure of graphical models. Rather than theory and proofs, we will focus on gathering the practical understanding necessary to create and use these models. Thus, there will be a strong emphasis on implementing the methods we study. Although the course is listed as a seminar, it will be taught as a regular lecture course with regular programming assignments and exams. Students entering the class should have a good programming skills and pre-existing working knowledge of probability, statistics, and algorithms. 3 credits.

**Seminar - Introduction to Modern Computational Neuroscience**

Hava Siegelmann

This course provides an introduction to neuroscience from a computational perspective. The emphasis is on mathematical and information processing models of the brain at a range of levels (synapses to behavior) and timescales (milliseconds to days). Topics include biophysics of single neurons, synapses, dendrites and axons; neural coding at the single cell and population level; dynamics of large networks, including computing with population codes; and learning at the systems and behavioral levels. We will also introduce systems in neuroscience based on their functionality: memory, emotions, and addiction. We will also study the new findings of neurogenesis and epigenetics. The course is run primarily for graduate students but undergrads with background in AI and mathematics or psychology and programming are invited to join. Minimal programming requirement: proficiency in any programming language to write basic programs and run small simulations. Minimal math requirement: algebra (matrix and vector operations). Extra knowledge in control theory or functional analysis or differential equations or graphic and programming design or neuroscience, though not required, will enable guided research projects. Most of the course material will be drawn from the textbook "Theoretical Neuroscience" by Peter Dayan & Larry Abbott (MIT Press, ISBN 0-262-04199-5) except for some newer topics. 3 credits.

Spring 2011 --- page 7 --- 2/14/2011
Computational social science is an emerging field at the intersection of computer science, statistics, and the social sciences, in which quantitative methods and computational tools are used to identify and answer social science questions. The field is driven by new sources of data from the Internet, government databases, university libraries, crowdsourcing systems, and more, as well as by recent advances in computational modeling, machine learning, statistics, and social network analysis. This course will provide an overview of computational social science, with particular emphasis on problems involving text analysis. The course will primarily consist of reading, presenting, and discussing recent papers. In addition, students taking the class for 3 credits will be expected to define and undertake a semester-long research project that relates to computational social science. Students entering the course should have a good knowledge of probability, statistics, and machine learning, as well as a strong interest in political science, public policy, and sociology. To facilitate productive discussion, students with diverse research backgrounds and interests are especially encouraged to participate. May be repeated for credit up to 2 times. Lect A=3 credits; Lect B=1 credit.

CMPSCI 791SS Seminar - Computational Social Science
Hanna Wallach

Computational social science is an emerging field at the intersection of computer science, statistics, and the social sciences, in which quantitative methods and computational tools are used to identify and answer social science questions. The field is driven by new sources of data from the Internet, government databases, university libraries, crowdsourcing systems, and more, as well as by recent advances in computational modeling, machine learning, statistics, and social network analysis. This course will provide an overview of computational social science, with particular emphasis on problems involving text analysis. The course will primarily consist of reading, presenting, and discussing recent papers. In addition, students taking the class for 3 credits will be expected to define and undertake a semester-long research project that relates to computational social science. Students entering the course should have a good knowledge of probability, statistics, and machine learning, as well as a strong interest in political science, public policy, and sociology. To facilitate productive discussion, students with diverse research backgrounds and interests are especially encouraged to participate. May be repeated for credit up to 2 times. Lect A=3 credits; Lect B=1 credit.

Spring 2011 --- page 8 --- 2/14/2011
Students will study additional topics in computability and complexity, through discussions (1 hour/week with professor), readings, and exercises. Specific topics will depend on student interest but may include algebraic automata theory, alternate models of computability, and interactive proofs. 1 credit.