

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 COMPSCI 119 or COMPSCI 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 CS majors. 3 credits.

COMPSCI 119 **Introduction to Programming** **CREDITS 3**

William Verts

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 CS majors. 3 credits.

COMPSCI 121 **Introduction to Problem Solving with Computers (R2)** **CREDITS 4**

Gordon Anderson

COMPSCI 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, COMPSCI 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 145 **Representing, Storing, and Retrieving Information** **CREDITS 3**

William Verts

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

COMPSCI 187 **Programming with Data Structures (R2)** **CREDITS 4**

Mark Corner

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: COMPSCI 121 (or equivalent Java experience) (A B or better in COMPSCI 121 is highly recommended for students enrolling in COMPSCI 187) and Basic Math Skills (R1). Basic Java language concepts are introduced quickly; if unsure of background, contact instructor. 4 credits.

COMPSCI 197C **Special Topics - Programming in C** **CREDITS 1**

Nicolas Scarrci

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 CS minors and majors only, but it does not count towards either degree. 1 credit.

| | | |
|--|---|------------------|
| COMPSCI 197U | Special Topics - A Hands-on Introduction to UNIX | CREDITS 1 |
| <i>John Foley</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. | | |
| COMPSCI 197WP | Special Topics - Introduction to HTML/CSS | CREDITS 1 |
| <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. | | |
| COMPSCI 220 | Programming Methodology | CREDITS 4 |
| <i>Arjun Guha</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. | | |
| COMPSCI 230 | Computer Systems Principles | CREDITS 4 |
| <i>John Ridgway, Kaituo Li</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 C language, and moves up the stack from there to the features of modern architectures, assembly languages, and operating system services such as I/O and synchronization. Prerequisites: COMPSCI 187 or ECE 242. 4 credits. | | |
| COMPSCI 240 | Reasoning Under Uncertainty | CREDITS 4 |
| <i>Barna Saha, Andrew McGregor</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. Prerequisites: COMPSCI 187 (or ECE 242) and MATH 132 or consent of instructor. 4 credits. | | |
| COMPSCI 250 | Introduction to Computation | CREDITS 4 |
| <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. | | |
| COMPSCI 290NW | A Networked World | CREDITS 3 |
| <i>Marc Liberatore</i> | | |
| The course will cover the technical foundations of today's communication networks, particularly the Internet. It will also address key social, policy, economic and legal aspects of these networks, their use (and abuse), and their regulation. This course covers computer science topics, but all material will be presented in a way that is accessible to an educated audience with or without a strong technical background. Not intended for Computer Science majors - students interested in a majors-level treatment of this material should see COMPSCI 453. 3 credits. | | |
| COMPSCI 305 | Social Issues in Computing | CREDITS 3 |
| <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 10 informal writing assignments and 4-6 larger written projects. Writing experiences will also include writing for electronic environments, collaborative writing, and public writing; there will be one individual and one team presentation assignment. Prerequisite: ENGLWRIT 112. 3 credits. | | |

- COMPSCI 348** **Introduction to Knowledge Discovery** **CREDITS 3**
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: COMPSCI 187 (or ECE 242), COMPSCI 240, and COMPSCI 250. 3 credits.
- COMPSCI 370** **Introduction to Computer Vision** **CREDITS 3**
Subhransu Maji
 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: COMPSCI 240 or 383. 3 credits.
- COMPSCI 370HH** **Introduction to Computer Vision with Honors Colloquium** **CREDITS 4**
Subhransu Maji *Honors Colloq*
 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. The colloquium will focus on advanced topics and recent research topics related to computer vision. Students will participate in group discussions and carry out a group or individual project which will be an extension to the project work in COMPSCI 370. Students will be graded based on their active participation during meetings, written summaries of assigned readings, and project work. Prerequisite: COMPSCI 240 or 383. 4 credits.
- COMPSCI 373** **Introduction to Computer Graphics** **CREDITS 3**
Evangelos Kalogerakis
 This course introduces the fundamental concepts of 2D and 3D computer graphics. It covers the basic methods for modeling, rendering, and imaging. Topics include: image processing, digital photography, 2D/3D modeling, 3D graphics pipeline, OpenGL, shading, texture mapping, ray tracing, 3D printing. Throughout the class, we will teach students to learn modern graphics techniques, to model the visual world algorithmically, and to implement algorithms using Java. Students who have taken COMPSCI 473 are not eligible to take this course. Prerequisites: COMPSCI 187 (or ECE 242) and COMPSCI 190DM (or MATH 235 or COMPSCI 240 or equivalent courses from other departments). 3 credits.
- COMPSCI 373HH** **Introduction to Computer Graphics with Honors Colloquium** **CREDITS 4**
Evangelos Kalogerakis *Honors Colloq*
 This course introduces the fundamental concepts of 2D and 3D computer graphics. It covers the basic methods for modeling, rendering, and imaging. Topics include: image processing, digital photography, 2D/3D modeling, 3D graphics pipeline, OpenGL, shading, texture mapping, ray tracing, 3D printing. Throughout the class, we will teach students to learn modern graphics techniques, to model the visual world algorithmically, and to implement algorithms using Java. In addition to the regular requirements of COMPSCI 373, honors section students will learn about advanced computer graphics techniques for automatic analysis, manipulation and synthesis of 3D content (shapes, scenes, textures, animations). Activities include discussions with the instructor and TA, paper readings, and a scall-scale research project involving the implementation of a 3D virtual/augmented reality application, game engine, a user interface for 3D modeling, or a machine learning algorithm for automatic analysis and synthesis of 3D models and animations. Students who have taken COMPSCI 473 are not eligible to take this course. Prerequisites: COMPSCI 187 (or ECE 242) and COMPSCI 190DM (or MATH 235 or COMPSCI 240 or equivalent courses from other departments). 4 credits.
- COMPSCI 377** **Operating Systems** **CREDITS 4**
Timothy Richards
 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. 4 credits.

| | | |
|---|--|----------------------|
| COMPSCI 383 | Artificial Intelligence | CREDITS 3 |
| <i>Patrick Taylor</i> | | |
| The course explores key concepts of artificial intelligence, including state-space and heuristic search techniques, problem-solving, game playing, knowledge representation, automated planning, reasoning under uncertainty, decision theory, and machine learning. AI is largely defined as the study of agents that receive percepts from the environment and perform actions. Each such agent implements a function that maps percept sequences to actions, and the course covers different ways to represent these functions, such as production systems, reactive agents, real-time conditional planners, neural networks, and decision-theoretic systems. Prerequisites: COMPSCI 220 (or COMPSCI 230) and COMPSCI 240. 3 credits. | | |
| COMPSCI 397F | Special Topics - Introduction to Data Science | CREDITS 3 |
| <i>Gordon Anderson</i> | | |
| The terms "data science" and "big data" appear in the news media and in everyday conversations. Moreover, we are told that we live in the "age of information", where almost every business venture and scientific research initiative collect a massive amount of data which may contain valuable information. This course is an introduction to the concepts and skills involved with the collection, management, analysis, and presentation of large data sets and the data products that result from the work of data scientists. Privacy and ethical issues are discussed. Students will work with data from the financial, epidemiological, educational, and other domains. The course provides many case studies and examples of real-world data that students work with using the R programming language as well as the structured query language (SQL). This course consists of two meetings per week. Each meeting includes a lecture, where conceptual material will be presented, followed by lab time where students receive instruction on the use of software tools and apply the concepts by working on data sets. Readings will be assigned as preparation for each class meeting. Several projects will be assigned during the course. The projects provide students with an opportunity to explore the topics in more depth in a specialized domain. Two midterm exams and one final exam will be given. Grades are determined by a combination of class participation including the in-class lab activities, projects, and exam scores. Software: The R software for statistical analysis (www.r-project.org). This course does not satisfy requirements for the CS major. Prerequisites: COMPSCI 190IN and (COMPSCI 119 or COMPSCI 121). 3 credits. | | |
| COMPSCI 397G | Special Topics - Creative Game Design and Development | CREDITS 3 |
| <i>Joshua Newman</i> | | |
| In this class, we will explore, through a series of projects, the fundamental questions of game design. What are the common features of hopscotch, Skyrim, boxing, Farmville, poker, and Tic-Tac-Toe? How do you create an engrossing, challenging, vivid, or surprising environment of play? How do you determine the value of skill, chance, cooperation, and competition in game play? What effect does the social, sexual, gender, political, and economic environment of the game's creation have on the play of the game? This course will introduce students to theoretical structures and demonstrate their implementation through the development of several small physical and digital games, ending with a group digital game project. Students will be evaluated based on class participation, process papers, and the creation of their own games. Not for CS Major/Minor Requirements. 3 credits. | | |
| COMPSCI 445 | Information Systems | CREDITS 3 |
| <i>Gerome Miklau</i> | | |
| This course is an introduction to the efficient management of large-scale data. The course includes principles for representing information as structured data, query languages for analyzing and manipulating structured data, and core systems principles that enable efficient computation on large data sets. Classical relational database topics will be covered (data modeling, SQL, query optimization, concurrency control), as well as semi-structured data (XML, JSON), and distributed data processing paradigms (e.g. map-reduce). Additional application topics may include web application development, data integration, processing data streams, database security and privacy. Prerequisite: COMPSCI 220 (or 230) and COMPSCI 311. 3 credits. | | |
| COMPSCI 445HH | Information Systems with Honors Colloquium | CREDITS 3 |
| <i>Gerome Miklau</i> | | <i>Honors Colloq</i> |
| This course is an introduction to the efficient management of large-scale data. The course includes principles for representing information as structured data, query languages for analyzing and manipulating structured data, and core systems principles that enable efficient computation on large data sets. Classical relational database topics will be covered (data modeling, SQL, query optimization, concurrency control), as well as semi-structured data (XML, JSON), and distributed data processing paradigms (e.g. map-reduce). Additional application topics may include web application development, data integration, processing data streams, database security and privacy. The colloquium will focus on advanced topics and recent research topics related to information management and data science. Students will participate in group discussions and carry out a group or individual project which will extend the project work in COMPSCI 445. Students will be graded based on their active participation during meetings, written summaries of assigned readings, and project work. Prerequisite: COMPSCI 220 (or 230) and COMPSCI 311. 4 credits. | | |
| COMPSCI 446 | Search Engines | CREDITS 3 |
| <i>David Fisher</i> | | |
| 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 COMPSCI 240) is required. You should also be able to program in Java (or some other closely related language). Prerequisite: COMPSCI 240 or COMPSCI 383, or equivalent. 3 credits. | | |

| | | |
|--|--|------------------|
| COMPSCI 453 | Computer Networks | CREDITS 3 |
| <i>John Ridgway</i> | | |
| Introduction to computer communication networks and protocols. Fundamental concepts in the design and analysis of computer networks. Topics include: layered network architectures, applications, network, programming interfaces, transport, congestion, routing, data link protocols, local area networks, emerging high-speed networks, network security, and wireless networks. Examples drawn from the Internet (e.g., TCP, UDP, and IP) protocol suite. Homework assignments involve programming and written tasks. Prerequisites: Experience programming; COMPSCI 230 (or COMPSCI 377). 3 credits. | | |
| COMPSCI 491G | Seminar - Computer Networking Lab | CREDITS 3 |
| <i>Arunkumar Venkataramani</i> | | |
| 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. This course counts as a CS Elective toward the CS major (BA/BS). Prerequisite: COMPSCI 453. 3 credits. | | |
| COMPSCI 496S | Independent Study - Scalable Web Systems | CREDITS 1 |
| <i>Timothy Richards</i> | | |
| 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 |
| <i>Timothy Richards</i> | | |
| 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 501 | Formal Language Theory | CREDITS 3 |
| <i>David Barrington</i> | | |
| 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: COMPSCI 311 or equivalent. It is recommended that students have a 'B-' or better in 311 in order to attempt 501. 3 credits. | | |
| COMPSCI 520 | Software Engineering: Synthesis and Development | CREDITS 3 |
| <i>Rene Just</i> | | |
| 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: COMPSCI 320 with a grade of 'C' or better. 3 credits. | | |

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

John Ridgway

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 535 **Computer Architecture** **CREDITS 3**

Charles Weems

The structure of digital computers is studied at several levels, from the basic logic level, to the component level, to the system level. Topics include: the design of basic components such as arithmetic units and registers from logic gates; the organization of basic subsystems such as the memory and I/O subsystems; the interplay between hardware and software in a computer system; the von Neumann architecture and its performance enhancements such as cache memory, instruction and data pipelines, coprocessors, and parallelism. Weekly assignments, semester project, 2 hours exams, final. Prerequisites: COMPSCI 391IB/335. 3 credits.

COMPSCI 589 **Machine Learning** **CREDITS 3**

Benjamin Marlin

This course will introduce core machine learning models and algorithms for classification, regression, clustering, and dimensionality reduction. On the theory side, the course will focus on understanding models and the relationships between them. On the applied side, the course will focus on effectively using machine learning methods to solve real-world problems with an emphasis on model selection, regularization, design of experiments, and presentation and interpretation of results. The course will also explore the use of machine learning methods across different computing contexts including desktop, cluster, and cloud computing. The course will include programming assignments, a midterm exam, and a final project. Python is the required programming language for the course. Prerequisites: COMPSCI 383 and MATH 235. 3 credits.

COMPSCI 589 (CPE) **Machine Learning** **CREDITS 3**

Rukmini Vijaykumar

This course will introduce core machine learning models and algorithms for classification, regression, clustering, and dimensionality reduction. On the theory side, the course will focus on understanding models and the relationships between them. On the applied side, the course will focus on effectively using machine learning methods to solve real-world problems with an emphasis on model selection, regularization, design of experiments, and presentation and interpretation of results. The course will also explore the use of machine learning methods across different computing contexts including desktop, cluster, and cloud computing. The course will include programming assignments, a midterm exam, and a final project. Python is the required programming language for the course. Students are expected to have an undergraduate background in Computer Science. Location: OFF CAMPUS-Boston, MA. 3 credits.

COMPSCI 589HH **Machine Learning with Honors Colloquium** **CREDITS 4**

Benjamin Marlin

Honors Colloq

This course will introduce core machine learning models and algorithms for classification, regression, clustering, and dimensionality reduction. On the theory side, the course will focus on understanding models and the relationships between them. On the applied side, the course will focus on effectively using machine learning methods to solve real-world problems with an emphasis on model selection, regularization, design of experiments, and presentation and interpretation of results. The course will also explore the use of machine learning methods across different computing contexts including desktop, cluster, and cloud computing. The course will include programming assignments, a midterm exam, and a final project. Python is the required programming language for the course. This course is an honors colloquium for COMPSCI 589: Machine Learning. It will include an exploration of the mathematical foundations of the machine learning algorithms presented in class. It will also include the presentation of more advanced models and algorithms. Students will participate in and lead group discussions on these topics, as well as on their final course projects. Prerequisites: COMPSCI 383 and MATH 235. 4 credits.

| | | |
|--|---|------------------|
| COMPSCI 590C | Human Computer Interaction | CREDITS 3 |
| <i>Eva Hudlicka</i> | | |
| <p>This course will provide an in-depth introduction to Human Computer Interaction (HCI). The topics will include: HCI history; discussion of human capabilities and limitations (perceptual, cognitive, physical), human error, and human emotions, and their relevance for HCI; generic principles of design and the principles guiding user interface and interaction design; design of WIMP (windows, icons, menus, pointing) and direct-manipulation user interfaces; virtual and augmented reality and novel modes of interaction; approaches to prototyping (storyboarding, paper prototyping, and low- and high-fidelity user interface prototyping); cognitive modeling; usability evaluation; affective HCI and current trends in HCI; as well as ethical considerations. The course will include several brief projects reinforcing selected topics (UI design, cognitive modeling, usability evaluation) and a semester-long major project that will involve the design and development of a mobile device app. This course counts as a CS Elective toward the CS major (BA/BS). Undergraduate prerequisites: COMPSCI 187 (or E&C-ENG 242) and COMPSCI 220 (or COMPSCI 230). 3 credits.</p> | | |
| COMPSCI 590CC | Cloud Computing | CREDITS 3 |
| <i>Arun Venkataramani</i> | | |
| <p>This course will teach the principles and practice of distributed systems as applied in today's cloud computing environments. The course will cover concepts in distributed computing including distributed clocks, consistency, fault tolerance, and consensus. The course will also cover popular cloud computing service models, MapReduce and related programming models, datacenter architectures, and security and privacy issues in public clouds. The course will expose students to popular cloud platforms such as Amazon EC2, Google AppEngine, and Microsoft Azure. The assignments in the course will include a nontrivial programming component that assumes familiarity with multithreaded programming and network programming. This course counts as a CS Elective toward the CS major (BA/BS). Prerequisite: COMPSCI 377 or an equivalent first course in operating systems. 3 credits.</p> | | |
| COMPSCI 590D | Algorithms for Data Science | CREDITS 3 |
| <i>Barna Saha</i> | | |
| <p>Big Data brings us to interesting times and promises to revolutionize our society from business to government, from healthcare to academia. As we walk through this digitized age of exploded data, there is an increasing demand to develop unified toolkits for data processing and analysis. In this course our main goal is to rigorously study the mathematical foundation of big data processing, develop algorithms and learn how to analyze them. Specific Topics to be covered include: 1) Clustering 2) Estimating Statistical Properties of Data 3) Near Neighbor Search 4) Algorithms over Massive Graphs and Social Networks 5) Learning Algorithms 6) Randomized Algorithms. This course counts as a CS Elective toward the CS major. 3 credits.</p> | | |
| COMPSCI 591NR | Seminar - Neural Networks-An Introduction | CREDITS 3 |
| <i>Hava Siegelmann</i> | | |
| <p>We will introduce neural networks and modeling of brain functions: the inspiration, engineering applications, theoretical analysis, as well as psychological and biological modeling. We will review different neural networks within the two groups: Feed forward networks, applications, and approximation theorem; Recurrent neural networks and computational power; We will also focus on supervised and unsupervised learning with applications in clustering; representability; and Models of memory, diseases, and healthy perception. This course counts as a CS Elective toward the CS major. Prerequisite: COMPSCI 311. 3 credits.</p> | | |
| COMPSCI 591SP | Seminar - Multidisciplinary Study of Security and Privacy | CREDITS 3 |
| <i>Brian Levine</i> | | |
| <p>Evaluating and addressing the security and privacy needs of real systems is increasingly a multidisciplinary task. Even if systems could be "fully secured" against all technical vulnerabilities they would still play a role in violating higher-layer policies. For example, most systems can be used to aid and abet crimes that harm persons, steal resources, or play a role in creating risk. Interdisciplinary work has a greater chance at addressing problems beyond technical flaws. In this course, we will use blockchain solutions as a platform for studying these issues. We will examine the cryptography that supports Bitcoin, security and privacy attacks on its Internet-based architecture, and the financial, legal, and social issues the technology perturbs. Assignments will include advanced programming projects and reading research papers. This course counts as a CS Elective toward the CS major (BA/BS), as well an Any 2 menu choice for the Security & Privacy track. Undergraduate Prerequisite: COMPSCI 460 or COMPSCI 453. 3 credits.</p> | | |
| COMPSCI 597CR | Special Topics - Crypto Engineering [Cross-List w/ECE 597CR] | CREDITS 3 |
| <i>Christof Paar, Wayne Burleson</i> | | |
| <p>Cryptographic algorithms are the "engine" that drives virtually every security solution, from email encryption to biometric passports. This course investigates the engineering challenges that arise in the context of employing cryptography. The course has two main parts: (1) Crypto Algorithms with practical relevance and the efficient implementation, and (2) Contemporary topics in security and crypto engineering. With respect to the first part, we will treat AES, RSA and digital signatures. These are the most widely used crypto schemes in practice. They are used in a large number of security applications, e.g., in SSL for secure web browsing. For each of the crypto schemes, the details of how they work and the underlying mathematical concepts will be introduced. The second part of the course will address topics in modern crypto engineering. These include hardware and software implementations of block and stream ciphers, public-key schemes, random number generation, and physical unclonable functions. Applications in payment systems and medical devices will be highlighted. Implementation attacks such as power, timing and electromagnetic side-channels will also be introduced. The course grade will be based on homework assignments, exams and projects. This course counts as a substitute for COMPSCI 466 toward the CS major (BA/BS). 3 credits.</p> | | |

| | | |
|---|---|------------------|
| COMPSCI 601 | Computation Theory | CREDITS 3 |
| <i>Neil Immerman</i> | | |
| 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: an undergraduate course in automata theory and formal languages such as COMPSCI 501 or permission of instructor. Course requirements: biweekly problem sets, midterm and final. Also open to qualified undergraduates. 3 credits. | | |
| COMPSCI 603 | Robotics | CREDITS 3 |
| <i>Joydeep Biswas</i> | | |
| This course is intended to serve as an advanced overview of robotics spanning the complete autonomy loop: robot hardware, perception, planning, and control. We will study algorithms and data structures related to these topics, covering widely adopted, and state of the art techniques. Students will gain hands-on experience in implementing, and extending such algorithms using real robot data, as well as simulations. 3 credits. | | |
| COMPSCI 620 | Advanced Software Engineering: Synthesis and Development | CREDITS 3 |
| <i>Rene Just</i> | | |
| 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. | | |
| COMPSCI 631 | Programming Languages | CREDITS 3 |
| <i>Arjun Guha</i> | | |
| This course surveys the principles of modern programming languages by implementation. Topics include building interpreters for several kinds of programming languages, type-inference algorithms, program analyses, and program verification. We will conclude with topics that bridge to compilers and runtime systems, including program transformations and garbage collection. 3 credits. | | |
| COMPSCI 635 | Modern Computer Architecture | CREDITS 3 |
| <i>Charles Weems</i> | | |
| 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: COMPSCI 535 or equivalent. 3 credits. | | |
| COMPSCI 650 | Applied Information Theory | CREDITS 3 |
| <i>Arya Mazumdar</i> | | |
| This course will introduce the basic concepts of Information Theory: entropy, relative entropy, mutual information, channel capacity, and rate distortion (data compression). Applications of these concepts will be emphasized. The notions of sparse-graph codes, message passing and belief propagation will also be introduced and studied in detail. Applications in inference algorithms and machine learning will be highlighted; in particular, topics of neural networks, statistical estimation, community detection and clustering will be covered. 3 credits. | | |
| COMPSCI 677 | Operating Systems | CREDITS 3 |
| <i>Prashant Shenoy</i> | | |
| 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. <i>Prerequisites:</i> 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., COMPSCI 377) will be helpful but not required. Lect 2 is on-line. 3 credits. | | |

| | | |
|--|---|--------------------|
| COMPSCI 683 | Artificial Intelligence | CREDITS 3 |
| <i>Shlomo Zilberstein</i> | | |
| 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 688 | Probabilistic Graphical Models | CREDITS 3 |
| <i>Brendan O'Connor</i> | | |
| Probabilistic graphical models are an intuitive visual language for describing the structure of joint probability distributions using graphs. They enable the compact representation and manipulation of exponentially large probability distributions, which allows them to efficiently manage the uncertainty and partial observability that commonly occur in real-world problems. As a result, graphical models have become invaluable tools in a wide range of areas from computer vision and sensor networks to natural language processing and computational biology. The aim of this course is to develop the knowledge and skills necessary to effectively design, implement and apply these models to solve real problems. The course will cover (a) Bayesian and Markov networks and their dynamic and relational extensions; (b) exact and approximate inference methods; (c) estimation of both the parameters and structure of graphical models. Although the course is listed as a seminar, it will be taught as a regular lecture course with programming assignments and exams. Students entering the class should have good programming skills and knowledge of algorithms. Undergraduate-level knowledge of probability and statistics is recommended. 3 credits. | | |
| COMPSCI 690OP | Optimization in Computer Science | CREDITS 3 |
| <i>Sridhar Mahadevan</i> | | |
| Much recent work in computer science in a variety of areas, from game theory to machine learning and sensor networks, exploits sophisticated methods of optimization. This course is intended to give students an in-depth background in both the foundations as well as some recent trends in the theory and practice of optimization for computer science. There is currently no course in the department that covers these topics, and yet it is critical to a large number of research projects done within the department. 3 credits. | | |
| COMPSCI 691DD | Seminar - Research Methods in Empirical Computer Science | CREDITS 3 |
| <i>Emery Berger</i> | | |
| 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. | | |
| COMPSCI 691NR | Seminar - Neural Networks-An Introduction | CREDITS 3 |
| <i>Hava Siegelmann</i> | | |
| We will introduce neural networks and modeling of brain functions: the inspiration, engineering applications, theoretical analysis, as well as psychological and biological modeling. We will review different neural networks within the two groups: Feed forward networks, applications, and approximation theorem; Recurrent neural networks and computational power; We will also focus on supervised and unsupervised learning with applications in clustering; representability; and Models of memory, diseases, and healthy perception. 3 credits. | | |
| COMPSCI 697D | Seminar - Special Topics in Distributed Systems and Networks | CREDITS 1-3 |
| <i>Donald Towsley, Ramesh Sitaraman</i> | | |
| Course description not on file. Lect 01=3 credits; Lect 02=1 credit. | | |
| COMPSCI 701 | Advanced Topics in Computer Science | CREDITS 6 |
| <i>STAFF</i> | | |
| 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. | | |

**** DELETED**

Many cognitive abilities once solely the province of biological systems are now routinely achievable with machines. Once no more than a dream, abilities such as complex 3D perception, natural language, machine translation, speech recognition, learning, and reasoning are now routinely achievable even on low cost hardware, such as cellphones, tablets, or similar devices, which can access larger scale servers for offline computation. This seminar explores the next frontiers for AI, presuming that many human cognitive abilities will be largely achieved in the next decade (vision, language, learning, reasoning, speech recognition), with the massive computing and data resources likely to be available to individuals and to corporations like Google, Baidu, Facebook, and IBM. To put it another way, core cognitive abilities are no longer "mysteries". but matters of mundane albeit challenging engineering. What's left for AI? Will "artificial intelligence" be achieved when machines achieve human level performance in these core cognitive abilities, or, as one philosopher put it in the early heady days of AI, have we merely climbed a tall tree on the way to getting to the moon? Our thesis is that the really interesting aspects of AI are only now beginning to be possible, given that the "operating system" level functionalities listed above are achievable. This seminar will discuss how AI research in the next decade or two can begin to tackle each of these problems. 1. Emotion: current AI systems can learn effectively from reinforcement, but feel no emotion. Is emotion necessary to build truly intelligent cyborgs? We explore the current theories of emotion, and its connections to rational decision making. 2. Consciousness: AI systems can carry out inference and reason about choices, but are not conscious of their own self, in the sense that most mammals are. What's missing? We review the current theories of consciousness. 3. Curiosity: AI systems can be programmed to carry out tasks using high level rewards and goals, but seem incapable of setting themselves their own tasks and goals. How can we build AI systems that are curious in the sense that children are. What are the essential elements of curiosity? 4. Mortality: AI systems have no fear of "death" and do not concern themselves with their "mortality", in the sense that humans (and presumably other mammals) do. However, as AI systems learn from experience and compile massive knowledge bases, they may face similar challenges in being "unplugged", faced with the loss of everything they know and have learned. (HAL in 2001: A Space Odyssey certainly feared its own death). Why is mortality important to incorporate in future AI systems? 3 credits.

Barna Saha

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. 1 credit.