

CICS 191CICS1 CICS First Year Seminar

INSTRUCTOR(S): Purity Mugambi, Brittany Johnson, Justin Obara, Ayse Ozisik, Arta Razavi, Edward Rietman, Michelle Trim, Robin

An exploration of computer science for first-year computer science majors and exploratory track students, focusing on a single topic. 1 credit.

Section 01- What Makes a Good Computer Scientist?

Section 02- Great Women in Computing: Past, Present, and Future

Section 03- Law and Computer Science

Section 04- What Makes a Good Computer Scientist?

Section 05- Computer Science Brain Teasers

Section 06- What Makes a Good Computer Scientist?

Section 07- What Makes a Good Computer Scientist?

Section 08- Computer Science Brain Teasers

Section 09- The Technological Singularity

Section 10- The Technological Singularity

Section 11- What Makes a Good Computer Scientist?

Section 12- Computer Science Brain Teasers

Section 13- What Makes a Good Computer Scientist?

Section 14- Great Women in Computing: Past, Present, and Future

Section 15- Computer Science Brain Teasers

Section 16- What Makes a Good Computer Scientist?

CICS 191CICS2 Computing and Informatics Exploratory RAP First Year Seminar

INSTRUCTOR(S): Arta Razavi

A weekly seminar for students in the Computing and Informatics Exploratory Track RAP residential program. Exploration of strategies for success at UMass and computer science topics through presentations and problem-solving discussions. Open only to first-year students in the Computing and Informatics Exploratory Track RAP. This course does not count toward any requirements for the CS major or minor. 1 credit.

CICS 191CMPS2 CS Major RAP First Year Seminar

INSTRUCTOR(S): Marc Liberatore

A weekly seminar for students in the CS RAP residential program. Exploration of great ideas in computer science through readings, presentations, and problem-solving. Open only to first-year students in the CS RAP. This course does not count toward any requirements for the CS major or minor. 1 credit.

CICS 256 Make: A Hands-on Introduction to Physical Computing

INSTRUCTOR(S): Rui Wang, Tauhidur Rahman

Inspired by the Maker movement, this course provides a hands-on introduction to physical computing: sensing and responding to the physical world using computers. Specific topics include: basic electronics and circuit design, microcontroller programming using Arduinos, sensing and responding to the physical world, rapid prototyping (3D printing and laser cutting etc.), soft circuits and wearable electronics. The course will encourage and empower students to invent, design, and build practical hardware projects that interact with the physical world. This course has a required lab section, and counts as one of the CS Lab Science Requirement courses for the BS-CS. Prerequisite: COMPSCI 187 and Basic Math Skills (R1). 4 credits.

CICS 305 Social Issues in Computing

INSTRUCTOR(S): Michelle Trim, Justin Obara, Catherine Tisdale, Siobhan Mei

Through a careful analysis and discussion of a range of computing issues, topics, and policies, we will explore various impacts of computers on modern society. This class satisfies the Junior Year Writing requirement by providing directed practice and specific instruction in a range of writing genres. Students will produce approximately 20-25 pages of polished written work over the course of the semester. Prerequisite: ENGLWRIT 112 (or equivalent) and COMPSCI 220 and COMPSCI 230 and COMPSCI 240 (or COMPSCI 250). 3 credits.

CICS 397A Special Topics - Predictive Analytics with Python

INSTRUCTOR(S): Matthew Rattigan

Twenty first century technological advances are generating ever-greater volumes of data. Examples of these sources include the ubiquitous internet and the notorious smart phone. There is an astounding number of opportunities to use these data for good (and bad) in the applied sciences, business, social media, politics, cyber security, to name a few areas. Gaining insight from these data requires a firm understanding of the mathematics and computational methods upon which the methods are based and put into use. That said, the elements of data science are indeed accessible to a fairly broad audience, and so our goal is to provide course participants with an understanding of these elements through application. The specific course objectives are to educate participants in some of the most commonly used data analytics including methods for reducing massively large data to informative statistics, data visualization, and cluster analysis. Practical data science demands the ability to program in a scripting language and therefore, students in this course will learn and use the most popular of these languages Python. The first learning goal is to understand these central data analytic methods, and the second learning goal is to know how to use them with Python. Our approach is close to the metal you'll create the Python scripts from the ground up and apply them to real and fascinating data sets. This course requires mathematical background in probability and statistics, calculus, and background in linear algebra is helpful but not required. The general awareness of big data applications of current environment gives better insight of the course. Recommended Text: Steele, Brian, Chandler, John, Reddy, Swarna; Algorithms for Data Science; ISBN: 978-3-319-45795-6. Non-CS Majors only. Does not count as a CS Elective (BA or BS). Prerequisites: Either COMPSCI 190F, or COMPSCI 119 and STATISTCS 240 (or OIM 240, or PSYCH 240, or STATISTCS 515, or RES-ECON 212, or SOCIOL 212). 3 credits.

COMPSCI 105 Computer Literacy

INSTRUCTOR(S): 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. 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

INSTRUCTOR(S): William Verts

This introduction to computer programming emphasizes problem solving using the Python language. Students will explore basic concepts in computer science and computer programming by writing Python 3 programs to analyze text, create files of graphics commands viewable in a Web browser, and many other tasks. No prior programming experience is needed. Not for CS majors. 3 credits.

COMPSCI 120 Introduction to Problem Solving with the Internet

INSTRUCTOR(S): 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-CS majors with timely skills needed to tap the net as well as an introduction to basic networking, client-side web programming in HTML, CSS, and Javascript, and server-side programming in Python. In addition to static and dynamic web page and web site design and implementation, we will cover strategies for finding information, managing e-mail, and ensuring privacy. We will survey current social, technical, and political topics that are relevant to the Internet such as spam and malware, net neutrality, censorship, copyright laws, and public key cryptography. Prerequisites: some hands-on experience with PCs or MACs or UNIX (programming experience is NOT required). Not for CS majors. 3 credits.

COMPSCI 121 Introduction to Problem Solving with Computers

INSTRUCTOR(S): Gordon Anderson, Neena Thota

COMPSCI 121 provides an introduction to problem solving and computer programming using the programming language Java. 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, polymorphism, arrays, graphics, and GUIs. No previous programming experience is required; however, this course is intended for Computer Science majors or those who plan on applying to the major. Non-majors are strongly encouraged to take one of our programming courses designed for non-majors. Use of a laptop computer on which you can install software is required. Prerequisite: R1 (or a score of 20 or higher on the math placement test Part A), or one of the following courses: MATH 101&102 or MATH 104 or MATH 127 or MATH 128 or MATH 131 or MATH 132. 4 credits.

COMPSCI 186 Using Data Structures

INSTRUCTOR(S): Marc Liberatore

This course introduces foundational abstract data types and algorithms. The main focus is on the use of data structures in designing and developing programs to solve problems in a variety of domains. Specific topics include lists, sets, maps, graphs, stacks, queues, searching, and sorting. There will be weekly programming assignments, assignments in discussion sections consisting of programming and written exercises, several announced quizzes, and a final exam. This course is not a substitute for COMPSCI 187. If unsure of whether this course or COMPSCI 187 is more appropriate, contact instructor. Was COMPSCI 190D. Prerequisites: COMPSCI 121 and Basic Math Skills (R1). 4 credits.

COMPSCI 187 Programming with Data Structures

INSTRUCTOR(S): Mark Corner, Meng-Chieh Chiu

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. The course involves weekly programming assignments, in-class quizzes, discussion section exercises, and multiple exams. Prerequisites: COMPSCI 121 (or equivalent Java experience). A grade of B or better in COMPSCI 121 (or a grade of C or better in COMPSCI 186 (or COMPSCI 190D) is required 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 220 Programming Methodology

INSTRUCTOR(S): Arjun Guha

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. 4 credits.

COMPSCI 230 Computer Systems Principles

INSTRUCTOR(S): J Moss, Meng-Chieh Chiu

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. 4 credits.

COMPSCI 240 Reasoning Under Uncertainty

INSTRUCTOR(S): Sunghoon Lee, Jie Xiong

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. Counting and probability -- basic counting problems, probability definitions, mean, variance, binomial distribution, discrete random variables, continuous random variables, Markov and Chebyshev bounds, Laws of large number, and central limit theorem. Probabilistic reasoning -- conditional probability and odds, Bayes' Law, Markov Chains, Bayesian Network, Markov Decision Processes. Prerequisites: COMPSCI 187 and MATH 132 or consent of instructor. 4 credits.

COMPSCI 250 Introduction to Computation

INSTRUCTOR(S): Marius Minea

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 midterm exams, timed final. Prerequisite: MATH 132 and COMPSCI 187. 4 credits.

COMPSCI 311 Introduction to Algorithms

INSTRUCTOR(S): Marius Minea, David Barrington

This course will introduce you to algorithms in a variety of areas of interest, such as sorting, searching, string-processing, and graph algorithms. You will learn to study the performance of various algorithms within a formal, mathematical framework. You will also learn how to design very efficient algorithms for many kinds of problems. There will be one or more programming assignments as well to help you relate the empirical performance of an algorithm to theoretical predictions. Mathematical experience (as provided by COMPSCI 250) is required. You should also be able to program in Java, C, or some other closely related language. Prerequisite: COMPSCI 187 and either COMPSCI 250 or MATH 455. 4 credits.

COMPSCI 320 Introduction to Software Engineering

INSTRUCTOR(S): Gordon Anderson, 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, in-class presentations, and a term project. This course satisfies the IE Requirement. Prerequisite: COMPSCI 220. 4 credits.

COMPSCI 325 Introduction to Human Computer Interaction

INSTRUCTOR(S): Narges Mahyar

Human-Computer Interaction design is "design for human use". Computers are a ubiquitous part of many interactions in our lives, from the mundane everydayness of light switches and "smart" vending machines to entertainment and education to sophisticated instruments and complex energy and defense systems. In this course, we will challenge you to broaden your grasp of what a user interface can and should be, and try your hand at doing better yourself. It is a fast-paced, hands-on, project-based experience that will challenge many of your ideas of what computer science is and can be. It is designed around active lecture sessions supported by readings, working classes, and team projects, where students practice and explore the concepts introduced in lecture, and go well beyond them to learn and apply HCI techniques that build into group projects. More specifically, the course adopts a human-centered design (HCD) approach and teaches a highly iterative process called design thinking. The design thinking process draws heavily on the fundamentals of human-computer interaction (HCI) methods. I also cover design methodologies, evaluation methodologies (both quantitative and qualitative), human information processing, cognition, and perception. 3 credits.

COMPSCI 326 Web Programming

INSTRUCTOR(S): Timothy Richards

The World Wide Web was proposed originally as a collection of static documents inter-connected by hyperlinks. Today, the web has grown into a rich platform, built on a variety of protocols, standards, and programming languages, that aims to replace many of the services traditionally provided by a desktop operating system. Topics will include: producing dynamic content using a server-based language, content serving databases and XML documents, session state management, multi-tier web-based architectures, web security, and core technologies including HTTP, HTML5, CSS, JavaScript, and SQL will be emphasized. This course will also study concepts and technologies including AJAX, social networking, mashups, JavaScript libraries (e.g., jQuery), and web security. This course is hands-on and project-based; students will construct a substantial dynamic web application based on the concepts, technologies, and techniques presented during lecture. This course satisfies the IE Requirement. Prerequisites: COMPSCI 220 or COMPSCI 230. 4 credits.

COMPSCI 335 Inside the Box: How Computers Work

INSTRUCTOR(S): 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, then move on to programming an embedded ARM processor in a mix of assembly language and C, interfacing with various I/O devices and sensors, to experience what happens when machine code executes. We will also see the impact of hidden acceleration mechanisms like caches, pipelines, and branch predictors. Prerequisite: COMPSCI 220 (or COMPSCI 230). 3 credits.

COMPSCI 345 Practice and Applications of Data Management

INSTRUCTOR(S): Alexandra Meliou

Computing has become data-driven, and databases are now at the heart of commercial applications. The purpose of this course is to provide a comprehensive introduction to the use of data management systems within the context of various applications. Some of the covered topics include application-driven database design, schema refinement, implementation of basic transactions, data on the web, and data visualization. The class will follow a flipped classroom model; students will be required to review materials in preparation for each week and they will work collaboratively on practical problems in class. This course counts as a CS Elective toward the COMPSCI major (BA/BS). Students who have completed COMPSCI 445 are not eligible to take this course without instructor permission. Prerequisite: COMPSCI 187. 3 credits.

COMPSCI 377 Operating Systems

INSTRUCTOR(S): 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

INSTRUCTOR(S): Matthew Rattigan, Jaime Dávila

The course explores key concepts underlying intelligent systems, which are increasingly deployed in consumer products and online services. Topics include 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 the use of these concepts in the design of intelligent agents in the context of several applications. Prerequisites: COMPSCI 220 (or COMPSCI 230) and COMPSCI 240 (or STAT 515). 3 credits.

COMPSCI 391L Seminar - Computer Crime Law and the Technologies of Investigation and Privacy

INSTRUCTOR(S): Marvin Cable

A study, analysis, and discussion of the legal issues related to crimes involving computers and networks, including topical actions by dissidents and governments. We will also study the technologies of forensic investigation, intelligence gathering, privacy enhancement, and censorship resistance. Our main legal topics will include recent and important case law, statutes, and constitutional clauses concerning authorization, access, search and seizure, wiretaps, the right to privacy, and FISA. Our technology topics will include methods of investigation and resistance in the context of the Internet and Cellular networks. Students are assumed to have no background in legal concepts. Students will be required to complete substantial legal readings, complete significant written analysis of rulings, learn about technologies in detail, and participate in lively class discussion. Prerequisite: COMPSCI 230 and ENGLWRIT 112. This course counts as a CS Elective toward the CS major. 3 credits.

COMPSCI 445 Information Systems

INSTRUCTOR(S): Gerome Miklau

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. MapReduce and Spark). Additional application topics may include web application development, data integration, processing data streams, database security and privacy. Prerequisites: COMPSCI 220 (or 230) and COMPSCI 311 and COMPSCI 345. 3 credits.

COMPSCI 453 Computer Networks

INSTRUCTOR(S): Parviz Kermani

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 460 Introduction to Computer and Network Security

INSTRUCTOR(S): Amir Houmansadr

This course provides an introduction to the principles and practice of computer and network security with a focus on both fundamentals and practical information. The three key topics of this course are cryptography, privacy, and network security. Subtopics include ciphers, key exchange, security services (integrity, availability, confidentiality, etc.), network and web based security attacks, anonymous communications, vulnerabilities, and countermeasures. Students will complete several lab assignments. Grades will be determined by class participation, lab work, homework, quizzes, and exams. Prerequisites include COMPSCI 377 and a familiarity with Unix. 3 credits.

COMPSCI 503 Embedded Computing Systems

INSTRUCTOR(S): Roderic Grupen

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

COMPSCI 513 Logic in Computer Science

INSTRUCTOR(S): Neil Immerman

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

COMPSCI 514 Algorithms for Data Science

INSTRUCTOR(S): Cameron Musco

With the advent of social networks, ubiquitous sensors, and large-scale computational science, data scientists must deal with data that is massive in size, arrives at blinding speeds, and often must be processed within interactive or quasi-interactive time frames. This course studies the mathematical foundations of big data processing, developing algorithms and learning how to analyze them. We explore methods for sampling, sketching, and distributed processing of large scale databases, graphs, and data streams for purposes of scalable statistical description, querying, pattern mining, and learning. Course was previously COMPSCI 590D. Undergraduate Prerequisites: COMPSCI 240 and COMPSCI 311. 3 credits

COMPSCI 520 Theory and Practice of Software Engineering

INSTRUCTOR(S): Yuriy Brun

Introduces students to the principal activities and state-of-the-art techniques involved in developing high-quality software systems. Topics include: requirements analysis, formal specification methods, software design, software testing and debugging, program analysis, and automated software engineering. Prerequisites: COMPSCI 320 with a grade of C or better. 3 credits.

COMPSCI 529 Software Engineering Project Management

INSTRUCTOR(S): Gordon Anderson, 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 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 532 Systems for Data Science

INSTRUCTOR(S): Marco Serafini

In this course, students will learn the fundamentals behind large-scale systems in the context of data science. We will cover the issues involved in scaling up (to many processors) and out (to many nodes) parallelism in order to perform fast analyses on large datasets. These include locality and data representation, concurrency, distributed databases and systems, performance analysis and understanding. We will explore the details of existing and emerging data science platforms, including MapReduce-Hadoop, Spark, and more. Undergraduate Prerequisites: COMPSCI 311, COMPSCI 345, and COMPSCI 377. 3 credits.

COMPSCI 546 Applied Information Retrieval

INSTRUCTOR(S): David Fisher

This course will provide a "flipped classroom" experience, with both online lectures, programming exercises, and other activities as well as a weekly lecture/exploration/discussion section. This is a graduate level course intended to cover information retrieval and other information processing activities, from an applied perspective. There will be numerous programming projects, as well as short answer homeworks. It provides a richer technical follow on to COMPSCI 446 (Search Engines), for undergraduates interested in a deeper understanding of the technologies. It also provides a strong basis for continuing on with COMPSCI 646 (Information Retrieval), for those graduate students who are interested in a more complete theoretical coverage of the area. Topics will include: search engine construction (document acquisition, processing, indexing, and querying); learning to rank; information retrieval system performance evaluation; classification and clustering; other machine learning information processing tasks; and many more. Undergraduate prerequisites: COMPSCI 320 and either COMPSCI 383, COMPSCI 446, or COMPSCI 585. 3 credits.

COMPSCI 585 Introduction to Natural Language Processing

INSTRUCTOR(S): Mohit Iyyer

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

COMPSCI 589 Machine Learning

INSTRUCTOR(S): Madalina Fiterau Brostean, Robert Hall

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. Students will complete programming assignments and exams. Python is the required programming language for the course. Undergraduate Prerequisites: COMPSCI 383 and MATH 235. 3 credits.

COMPSCI 590CC Cloud Computing

INSTRUCTOR(S): Arun Venkataramani

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 or BS). Prerequisite: COMPSCI 377 or an equivalent first course in operating systems. 3 credits.

COMPSCI 590E Ethical Considerations in Computing

INSTRUCTOR(S): Michelle Trim

This course considers an array of ethical issues in computing. Readings, class discussions, and guest speakers will cover topics related to avenues of development in artificial intelligence, privacy, identity, inclusiveness, environmental responsibility, internet censorship, network policy, plagiarism, intellectual property and others. All examples will be drawn from current and recent events with readings from a range of sources both journalistic and academic. Course assignments will have real world applications and offer students opportunities for developing their speaking and writing skills. Class discussions will be a vibrant component of the course. Open to Graduate students only. 3 credits.

COMPSCI 590N Introduction to Numerical Computing with Python

INSTRUCTOR(S): David Wemhoener

This course is an introduction to computer programming for numerical computing. The course is based on the computer programming language Python and is suitable for students with no programming or numerical computing background who are interested in taking courses in machine learning, natural language processing, or data science. The course will cover fundamental programming, numerical computing, and numerical linear algebra topics, along with the Python libraries that implement the corresponding data structures and algorithms. The course will include hands-on programming assignments and quizzes. No prior programming experience is required. Familiarity with undergraduate-level probability, statistics and linear algebra is assumed. Open to Graduate students only. 1 credit.

COMPSCI 590OP Applied Numerical Optimization

INSTRUCTOR(S): Shiting Lan

This course provides an overview of the important topic of numerical optimization. In this introductory-level course, we will cover the basic concepts of optimization, the key algorithms, and their applications in image/signal processing, machine learning, and statistical estimation. Topics covered include, but are not limited to: i) the basic concepts in optimization, e.g., linear algebra overview, convex sets, norms, optimality conditions, duality, ii) common optimization algorithms, e.g., gradient descent methods, Newton and quasi-Newton methods, conjugate gradient, proximal methods, linear programming, interior point methods, stochastic methods, distributed methods, and iii) applications, e.g., fitting generalized linear models, neural networks, sparsity, recommender systems, image processing, network utility maximization. Required background: basic knowledge of Python programming, basic knowledge of probability and statistics, linear algebra, multivariate calculus. This course counts as a CS Elective (BS or BA). 3 credits.

COMPSCI 591CF Seminar - Cybersecurity Lecture Series

INSTRUCTOR(S): Daniel Holcomb, Wayne Burtleson, Brian Levine, Amir Houmansadr, Eric Sommers

Each week of this one-credit seminar will feature a speaker from industry or government or faculty from UMass. Presentations will focus on security research topics, and for outside speakers will also include discussion of their institution and professional environment. The intended audience is graduate and advanced undergraduate students with an interest in security topics. Meets with ECE/MATH 591CF. May be taken for credit up to 2 times. This course does not count toward any requirements for the CS major or minor. 1 credit.

COMPSCI 597N Special Topics - Introduction to Computer and Network Security

INSTRUCTOR(S): Parviz Kermani

This course provides an introduction to the principles and practice of computer and network security with a focus on both fundamentals and practical information. The three key topics of this course are cryptography, privacy, and network security. Subtopics include ciphers, key exchange, security services (integrity, availability, confidentiality, etc.), network and web based security attacks, anonymous communications, vulnerabilities, and countermeasures. Students will complete several lab assignments. Grades will be determined by class participation, lab work, homework, quizzes, and exams. Open to Graduate students only. 3 credits.

COMPSCI 611 Advanced Algorithms

INSTRUCTOR(S): Andrew McGregor

Principles underlying the design and analysis of efficient algorithms. Topics to be covered include: divide-and-conquer algorithms, graph algorithms, matroids and greedy algorithms, randomized algorithms, NP-completeness, approximation algorithms, linear programming. Prerequisites: The mathematical maturity expected of incoming Computer Science graduate students, knowledge of algorithms at the level of COMPSCI 311. 3 credits.

COMPSCI 613 Advanced Logic in Computer Science

INSTRUCTOR(S): 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, automatic theorem proving, model checking. We will learn about and use the Coq theorem prover, Datalog, a Model Checker, and SAT and SMT solvers. Prerequisites: Students taking this course should have undergraduate preparation in discrete math and algorithms. Requirements will include readings, class participation, weekly problem sets, a midterm and a final project. 3 credits.

COMPSCI 630 Systems

INSTRUCTOR(S): Emery Berger

This class is an in-depth introduction to systems, focusing on principles of system design that cross-cut numerous systems artifacts, including operating systems, databases, runtime systems, and architecture. We will cover all levels of the "system stack", from chips to distributed systems. This class may be used to satisfy systems core requirements. 3 credits.

COMPSCI 635 Modern Computer Architecture

INSTRUCTOR(S): Charles Weems

This course examines the structure of modern computer systems. We explore recent research results that are influencing modern 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 670 Computer Vision

INSTRUCTOR(S): Subhransu Maji

This course will explore current techniques for the analysis of visual data (primarily color images). In the first part of the course we will examine the physics and geometry of image formation, including the design of cameras and the study of color sensing in the human eye. In each case we will look at the underlying mathematical models for these phenomena. In the second part of the course we will focus on algorithms to extract useful information from images. This includes detection of reliable interest points for applications such as image alignment, stereo and instance recognition; robust representations of images for recognition; and principles for grouping and segmentation. Time permitting we will look at some additional topics at the end of the course. Course assignments will highlight several computer vision tasks and methods. For each task you will construct a basic system, then improve it through a cycle of error analysis and model redesign. There will also be a final project, which will investigate a single topic or application in greater depth. This course assumes a good background in basic probability, linear algebra, and ability to program in MATLAB. Prior experience in signal/image processing is useful but not required. 3 credits.

COMPSCI 682 Neural Networks: A Modern Introduction

INSTRUCTOR(S): Erik Learned-Miller

This course will focus on modern, practical methods for deep learning. The course will begin with a description of simple classifiers such as perceptrons and logistic regression classifiers, and move on to standard neural networks, convolutional neural networks, and some elements of recurrent neural networks, such as long short-term memory networks (LSTMs). The emphasis will be on understanding the basics and on practical application more than on theory. Most applications will be in computer vision, but we will make an effort to cover some natural language processing (NLP) applications as well, contingent upon TA support. The current plan is to use Python and associated packages such as Numpy and TensorFlow. Prerequisites include Linear Algebra, Probability and Statistics, and Multivariate Calculus. All assignments will be in the Python programming language. 3 credits.

COMPSCI 687 Reinforcement Learning

INSTRUCTOR(S): Philip Thomas

This course will provide an introduction to, and comprehensive overview of, reinforcement learning. In general, reinforcement learning algorithms repeatedly answer the question "What should be done next?", and they can learn via trial and error to answer these questions even when there is no supervisor telling the algorithm what the correct answer would have been. Applications of reinforcement learning span across medicine (How much insulin should be injected next? What drug should be given next?), marketing (What ad should be shown next?), robotics (How much power should be given to the motor?), game playing (What move should be made next?), environmental applications (Which countermeasure for an invasive species should be deployed next?), and dialogue systems (What type of sentence should be spoken next?), among many others. Broad topics covered in this course will include: Markov decision processes, reinforcement learning algorithms (model-based / model-free, batch / online, value function based, actor-critics, policy gradient methods, etc.), hierarchical reinforcement learning, representations for reinforcement learning, and connections to animal learning. Special topics may include ensuring the safety of reinforcement learning algorithms, theoretical reinforcement learning, and multi-agent reinforcement learning. This course will emphasize hands-on experience, and assignments will require the implementation and application of many of the algorithms discussed in class. PREREQUISITES: COMPSCI 589, or COMPSCI 689, or COMPSCI 683, with a grade of C or better. Familiarity with an object oriented programming language is required. 3 credits.

COMPSCI 689 Machine Learning

INSTRUCTOR(S): Benjamin Marlin

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.

COMPSCI 690C Foundations of Applied Cryptography

INSTRUCTOR(S): Adam O'Neill

This is a graduate-level introduction to cryptography, emphasizing formal definitions and proofs of security. Though the course is theoretical in nature, its viewpoint will be "theory applied to practice." We will discuss cryptographic algorithms that are used in practice and how to reason about their security. More fundamentally, we will try to understand what security "is" in a rigorous way that allows us to follow sound cryptographic principles and uncover design weaknesses. Tentatively, we will cover: blockciphers, pseudorandom functions and permutations, symmetric encryption schemes and their security, hash functions, message authentication codes and their security, authenticated encryption schemes and protocols such as SSL/TLS, public-key encryption schemes and their security, digital signature schemes and their security, and public-key infrastructures. 3 credits.

COMPSCI 690Q Quantum Information Systems

INSTRUCTOR(S): Donald Towsley

Fundamentals of quantum information systems, including quantum computation, quantum cryptography, and quantum information theory. Topics include: quantum circuit model, qubits, unitary operators, measurement, entanglement, quantum algorithms for factoring and search, quantum key distribution, error-correction and fault-tolerance, information capacity of quantum channels, complexity of quantum computation. 3 credits.

COMPSCI 691DD Seminar - Research Methods in Empirical Computer Science

INSTRUCTOR(S): David Jensen

This course introduces graduate and undergraduate students to concepts, practices, and tools for conducting effective research. 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 simulation of computational phenomena, and synthesizing broader theories. The course is 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 is required. Students are expected to participate actively in class discussions and to provide meaningful comments on the work of other students. For PhD students, this course helps accelerate their current and future research. For MS students, this course provides a grounding in research methods that will aid entry into research-oriented industrial positions and provide information about continuing to PhD studies. For undergraduates considering graduate studies, this course will help inform and accelerate that direction. This course can be used to satisfy the 499Y requirement for Departmental and Multidisciplinary Honors students whose theses or projects have a substantial empirical component. Approval of the Computer Science Honors Program Director is required prior to registering. 3 credits.

COMPSCI 691O Seminar - Tools for Explanatory and Tutoring Systems

INSTRUCTOR(S): Beverly Woolf

This seminar examines recent work in explanatory and tutoring systems. Participants study artificial intelligence in education, a young field that explores theories about learning, and 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 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.

COMPSCI 691WM Seminar - Wearable and Mobile Sensors in Clinical Sciences

INSTRUCTOR(S): Sunghoon Lee

The health care systems of many industrialized nations, including the US, are facing daunting challenges such as a clear trend towards an aging of the population. This large elderly population having complex health conditions is beginning to severely stress the Medicare system. Researchers, health service providers and government leaders are seeking technological solutions to this problem in order to expand the capabilities of the healthcare system. In this seminar, we will discuss the current state-of-the-art and emerging on- and off-body sensors and mobile technologies, and their applications in clinical sciences. Various topics will be covered, including 1) different types of body sensors and mobile technologies that are motivated by practical medical needs, 2) construction of appropriate clinical trials using technological outcome measures, 3) analysis of the obtained data to quantify patients' conditions, and 4) validating the system's clinical efficacy. This course will primarily involve reading and discussing papers (1 credit), and a final project (3 credit). 1-3 credits.

COMPSCI 692A Seminar - Automated Knowledge Base Construction

INSTRUCTOR(S): Andrew McCallum

Knowledge gathering, representation, and reasoning are among the fundamental challenges of artificial intelligence. Large-scale repositories of knowledge about entities, relations, and their abstractions are known as "knowledge bases". Most major technology companies now have substantial efforts in knowledge base construction, and related scholarly work spans many research areas, including machine learning, deep learning, natural language processing, information integration, databases, search, data mining, knowledge representation, human computation, human-computer interfaces, and fairness. Recently there has been a surge of interest combined neural and symbolic representations that support queries and reasoning. In this course, we will study topics related to knowledge base construction and use, including entity recognition, entity resolution, relation extraction, embedded representations for knowledge, data structures for knowledge indexing, reasoning, semantic parsing, and question answering. There will be special focus on recent research in all of these areas. The course will consist of lectures, paper presentations, semester-long research projects. 3 credits.

COMPSCI 692E Seminar - Equitable Algorithms and Systems

INSTRUCTOR(S): Gerome Miklau, Alexandra Meliou, Przemyslaw Adam Grabowicz

This seminar will focus on recent research into equitable algorithms and systems, broadly construed. This includes research that supports properties of fairness, accountability, and transparency, as well as respect for the privacy, safety, and equitable treatment of contributing individuals. Specific research topics will span systems, artificial intelligence, and theory. 1 credit.

COMPSCI 692N Seminar - Novel Computational Hardware and Machine Learning

INSTRUCTOR(S): Edward Rietman

The seminar will consist of readings and discussions on novel computational systems and machine learning. Will include oscillatory computing, DNA computing, quantum computing, molecular computing, non-Turing architecture, among others. We will also look at novel machine learning algorithms. 3 credits.

COMPSCI 698W Practicum - CS Research Writing Practicum

INSTRUCTOR(S): Justin Obara

This CS research writing class uses a workshop format to focus on structure and phrasing while engaging students in a process-based approach to writing. Instruction will emphasize genre and discourse analysis and engage students in activities to strengthen audience awareness. As such, students will analyze representative examples of computer science research writing for stylistic and argumentative conventions and then integrate the awareness of these conventions and "moves" into their own writing. Students will produce or substantially revise a complete piece of writing. 6 weeks. 1 credit.

COMPSCI 701 Advanced Topics in Computer Science

INSTRUCTOR(S):

Advanced Topics in Computer Science Master's Project: Advanced research project in Computer Science.

The 3 credit option is for the second semester of a two semester sequence, 701Y followed by 701.

The 6 credit option is for a project that will be begun and completed within the same semester. 6 credits.

COMPSCI 701Y Advanced Topics in Computer Science (1st Semester)

INSTRUCTOR(S):

Advanced Topics in Computer Science Master's Project: Advanced research project in Computer Science.

Indicates the first semester of a two-semester sequence, 701Y (3 credits) followed by 701 (3 credits), with grade for both assigned at the end.. 3 credits.

COMPSCI 741 Complexity Theory

INSTRUCTOR(S): David Barrington

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

COMPSCI 890T Teaching Assistants as Tomorrow's Faculty

INSTRUCTOR(S): Neena Thota

Teaching Assistants as Tomorrow's Faculty prepares Teaching Assistants (TAs) at the College of Information and Computer Sciences to fulfill their duties in an effective and pedagogically sound manner. The one credit (not repeatable) course is semester long and taken by all TAs prior to assuming assistantship. 1 credit.

COMPSCI 891M Seminar - Theory of Computation

INSTRUCTOR(S): David Barrington

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.

COMPSCI H250 Honors Colloquium for Introduction to Computation

INSTRUCTOR(S): Marius Minea

We will have weekly readings from Godel, Esher, Bach: An Eternal Golden Braid by Douglas Hofstadter. This book contains mathematical problems related to the main course material, and presents some of the same topics as well as others. Students will report on their reading in the seminar, and we will discuss connections between the book and the CS 250 material. Each student will make an oral presentation on a topic of their choice at the end of the term. 1 credit.

COMPSCI H320 Honors Colloquium for Software Engineering

INSTRUCTOR(S): Leon Osterweil

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 COMPSCI 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.

COMPSCI H335 Honors Colloquium for Inside the Box: How Computers Work

INSTRUCTOR(S): Charles Weems

Honors section students are expected to meet weekly with the instructor. These meetings can be a combination of lecture, student research presentations, project progress reports, discussions, demonstrations of work, and problem solving. Students define their own program of enrichment, which will typically be either a research project or a project to develop an application of embedded systems. Students may work individually or in teams. Grading is based upon participation in the weekly meetings and the quality of the finished project. Prerequisite: COMPSCI 220 (or COMPSCI 230). 1 credit.

COMPSCI H345 Honors Colloquium for Practice and Applications of Data Management

INSTRUCTOR(S): Alexandra Meliou

Students will study in depth data management topics covered in class. Activities include discussions with the professor and TA, readings, and a project. The project is a semester-long group activity that combines several advanced topics into a practical application. 1 credit.

COMPSCI H589 Honors Colloquium for Machine Learning

INSTRUCTOR(S): Madalina Fiterau Brostean

This colloquium will enrich the primary course by focusing on reading, presenting, and discussing foundational and recent research papers from the machine learning literature. Students will write weekly reading responses, and lead one to two group discussions over the course of the semester. 1 credit.

INFO 101 Introduction to Informatics

INSTRUCTOR(S): Michelle Trim

An introduction to the main concepts of Informatics. There are several 'Big Ideas' in computing, including but not limited to abstraction, data and information, algorithms, programming, the internet, and the global impacts of computing. This class provides an introduction to those ideas and considers some of the ways that those computing principles might be used to solve real world problems. Computer-based assignments are an integral part of this course but no programming knowledge or prior programming experience is expected or required. Not for CS majors. 3 credits.

INFO 150 A Mathematical Foundation for Informatics

INSTRUCTOR(S): Peter Haas, Jaime Dávila

Mathematical techniques useful in the study of computing and information processing. The mathematical method of definition and proof. Sets, functions, and relations. Combinatorics, probability and probabilistic reasoning. Graphs and trees as models of data and of computational processes. Prerequisite: R1 math skills recommended. Not intended for Computer Science majors – students interested in a majors-level treatment of this material should see COMPSCI 240 and 250, or MATH 455. 3 credits.