### CICS 191CICS1 CICS First Year Seminar

INSTRUCTOR(S): Timothy Contois, Mark McCartin-Lim, Amirmohammad Rooshenas 1 Credits

An exploration of computer science for first-year computer science majors and exploratory track students, focusing on a single topic. Specific topics for specific sections are yet to be determined. 1 credit.

### CICS 191CICS2 Computing and Informatics Exploratory RAP First Year Seminar

INSTRUCTOR(S): Gregory Boisseau 1 Credits

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

### CICS 191CMPS2 CS Major RAP First Year Seminar

INSTRUCTOR(S): David Barrington, Jack Wileden 1 Credits

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

### CICS 298E Practicum - Educational Outreach

INSTRUCTOR(S): David Wemhoener 1 Credits

In this course, students will meet once a week to plan set of events during CS Education Week, December 4-10. Class overlaps with CICS 298L to allow coordination with the larger group of student volunteers. The class focus will be on planning events for CS Education Week and honing skills for that event. The events will take place at local primary and secondary schools, with the goal of raising student awareness of computer science as a field of study and profession and increasing student confidence in their ability to participate in computer science. This course is restricted to students in the College of Information and Computer Sciences. This course does not count toward CS Major requirements. 1 credit.

### CICS 298L Practicum - Educational Outreach: Leadership Principles and Practice

INSTRUCTOR(S): David Wemhoener 3 Credits

In this course, students will meet twice a week to plan set of educational events during CS Education Week, December 4-10. The Thursday meetings will overlap with 298E to allow coordination with the larger group of student volunteers. The focus of the course will be on planning events and learning about issues related to representation in computer science education. Students will take coordinating and/or supervisory roles at the sites where they run events, so they will also be expected to work on team-leadership skills. The events will largely take place at local primary and secondary schools, with the goal of raising student awareness of computer science as a field of study and profession and increasing student confidence in their ability to participate in computer science. This course is restricted to students in the College of Information and Computer Sciences. This course does not count toward CS Major requirements. 3 credits.

### COMPSCI 105 Computer Literacy

INSTRUCTOR(S): William Verts 3 Credits

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

INSTRUCTOR(S): William Verts 3 Credits

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 120 Introduction to Problem Solving with the Internet

INSTRUCTOR(S): William Verts 3 Credits

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, Ali Montazeralghaem, Zhichao Yang, Mohit Yadav, Shahrad 4 Credits

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, 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 187 Programming with Data Structures

INSTRUCTOR(S): Rui Wang, Nishant Yadav, Aritra Ghosh, Meng-Chieh Chiu, Venkatesh Narasimha Murthy 4 Credits

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 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 190D Using Data Structures

INSTRUCTOR(S): Marc Liberatore, Joshua Pikovsky, Colin Samplawski 4 Credits

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. Prerequisites: COMPSCI 121 and Basic Math Skills (R1). 4 credits.

### COMPSCI 197C Special Topics - Programming in C

INSTRUCTOR(S): John Foley 1 Credits

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 198J Practicum - High School Teaching Practicum

INSTRUCTOR(S): Emery Berger 1 Credits

Assisting high school students in learning computer programming in the context of a high school course. The language of instruction will be Java or Python. Students taking this independent study will be expected to work under the direct supervision of the high school computer science teacher at a local high school. Students are expected to attend classes weekly for approximately one hour. Course evaluation will be performed by the high school teacher in cooperation with the supervising faculty member, and a final written report will be required. 1 credit.

### COMPSCI 220 Programming Methodology

INSTRUCTOR(S): Roderic Grupen, Joydeep Biswas, Jarrett Holtz, Samer Nashed, Khoshrav Piran Doctor 4 Credits

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

INSTRUCTOR(S): J Moss 4 Credits

Large-scale software systems like Google - deployed over a world-wide network of hundreds of thousands of computers - have become a part of our lives. These are systems success stories - they are reliable, available ("up" nearly all the time), handle an unbelievable amount of load from users around the world, yet provide virtually instantaneous results. On the other hand, many computer systems don't perform nearly as well as Google - hence the now-cliché "the system is down." In this class, we study the scientific principles behind the construction of high-performance, scalable systems. The course begins with a discussion of 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

INSTRUCTOR(S): Arya Mazumdar, Sunghoon Lee, David Tench, Maham Haroon, James Kostas 4 Credits

Development of mathematical reasoning skills for problems that involve uncertainty. Each concept will be illustrated by real-world examples and demonstrated though in-class and homework exercises, some of which will involve Java programming. Counting and probability -- basic counting problems, probability definitions, mean, variance, binomial distribution, Markov and Chebyshev bounds. Probabilistic reasoning -- conditional probability and odds, Bayes' Law, Naive Bayes classifiers, Monte Carlo simulation. Markov chains, Markov decision processes, classical game theory, introduction to information theory. Prerequisites: COMPSCI 187 (or ECE 242) and MATH 132 or consent of instructor. 4 credits.

### COMPSCI 250 Introduction to Computation

INSTRUCTOR(S): David Barrington, Matteo Brucato, Hia Ghosh, Yuzhi Xiao 4 Credits

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). 4 credits.

### COMPSCI 305 Social Issues in Computing

INSTRUCTOR(S): Anna Napoleone, Michelle Trim 3 Credits

Through a careful analysis and discussion of a range of computing issues, topics, and polices, 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 (or COMPSCI 230) and COMPSCI 240 (or COMPSCI 250). 3 credits.

### COMPSCI 311 Introduction to Algorithms

INSTRUCTOR(S): Ramesh Sitaraman 4 Credits

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

### COMPSCI 320 Introduction to Software Engineering

INSTRUCTOR(S): Gordon Anderson, Neena Thota, Rui Huang, Jacob Downs, Ariel Reches 4 Credits

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): Eva Hudlicka, Stephen Giguere, Christopher Nota 3 Credits

In this course we examine the important problems in Usability, Human Computer Interaction, User Interfaces and Human Centered Computing. We will examine elements of HCI history, human information processing capabilities, HCI design, user interface prototyping methods and new applications and directions in HCI, including affective HCI. This is not a course on how to make better dialog boxes, but rather a much more thorough exploration of how humans interact with computers and how to evaluate the effectiveness of HCI designs. Some elementary programming, or the use of UI prototyping tools, will be required, but students without prior programming experience should feel right at home in this class. The course is available to any undergraduate student and is not limited to computer scientists. IT-minor students are especially encouraged to participate. The course emphasizes group projects and students will be required to work in teams. There is one midterm exam, numerous class presentations of on-going project work, and a final demo and presentation of the major class project. The course will be taught in a mixed format, with both on-line and regular classroom components. Lectures wlll be on-line, and weekly discussion groups and studios will be conducted in regular classroom settings. Attendance of some of the studio classes will be required, and there will be several project presentations throughout the semester. Students will be encouraged to use the available class time to work on their team projects. No prerequisites but Junior/Senior standing required. 3 credits.

### COMPSCI 326 Web Programming

INSTRUCTOR(S): Timothy Richards 3 Credits

The World Wide Web was proposed originally as a collection of static documents inter-connected by hyperlinks. Today, the web has grown into a rich platform, built on a variety of protocols, standards, and programming languages, that aims to replace many of the services traditionally provided by a desktop operating system. Topics will include: producing dynamic content using a server-based language, content serving databases and XML documents, session state management, multi-tier web-based architectures, web security, and core technologies including HTTP, HTML5, CSS, JavaScript, and SQL will be emphasized. This course will also study concepts and technologies including AJAX, social networking, mashups, JavaScript libraries (e.g., jQuery), and web security. This course is hands-on and project-based; students will construct a substantial dynamic web application based on the concepts, technologies, and techniques presented during lecture. This course satisfies the IE Requirement. Prerequisites: COMPSCI 220 or COMPSCI 230. 3 credits.

### COMPSCI 328 Mobile Health Sensing and Analytics

INSTRUCTOR(S): Deepak Ganesan 3 Credits

In recent years, the ability to continuously monitor activities, health, and lifestyles of individuals using sensor technologies has reached unprecedented levels. The typical smartphone comes equipped with a plethora of sensors for monitoring activity, speech patterns, social interactions, and location. In addition, mobile accessories such as wearable wristbands and chestbands now enable routine and continuous monitoring of a host of physiological signals (e.g., heart rate, respiratory rate, skin Conductance, and others.). In conjunction, these sensors can enable higher-order inferences about more complex human activities/behavioral states (e.g., activity patterns, stress, sleep, etc.). Such ubiquitous sensing in daily life, referred to as mobile health sensing and monitoring, promises to revolutionize our understanding of human activities and health conditions. This course is an introduction to personal health sensing and monitoring through mobile phones and on-body sensors and addresses several aspects including mobile devices and applications for health, sensor data quality and reliability challenges, inference of key health assessments from sensor data including such as activity patterns, sleep patterns, or stress, sensor data visualization and feedback, and practical considerations such as battery lifetime. This course counts as a CS Elective toward the CS major (BA/BS). Prerequisite: COMPSCI 187 (or equivalent). 3 credits.

### COMPSCI 335 Inside the Box: How Computers Work

INSTRUCTOR(S): Charles Weems 3 Credits

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, David Wemhoener 3 Credits

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 (or ECE 242). 3 credits.

### COMPSCI 377 Operating Systems

INSTRUCTOR(S): Timothy Richards 4 Credits

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): Ian Gemp 3 Credits

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. 3 credits.

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

INSTRUCTOR(S): Marc Liberatore 3 Credits

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 453 Computer Networks

INSTRUCTOR(S): Parviz Kermani 3 Credits

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 3 Credits

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 497P Special Topics - Programming Languages

INSTRUCTOR(S): Arjun Guha 3 Credits

This course surveys the principles of modern programming languages and provides a glimpse of contemporary programming languages research. This is an implementation-heavy course, where you will build typical language-based tools such as interpreters, type-checkers, and type inference algorithms. You will also implement emerging technologies, such as program verifiers and "fancy" type systems. This course satisfies a CS Elective for the CS Major (BS/BA). 3 credits.

### COMPSCI 513 Logic in Computer Science

INSTRUCTOR(S): Neil Immerman 3 Credits

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 520 Theory and Practice of Software Engineering

INSTRUCTOR(S): Yuriy Brun, Rene Just 3 Credits

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, Neena Thota 3 Credits

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

### COMPSCI 585 Introduction to Natural Language Processing

INSTRUCTOR(S): Brendan O'Connor 3 Credits

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): Justin Domke 3 Credits

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. Prerequisites: COMPSCI 383 and MATH 235. 3 credits.

### COMPSCI 590CC Cloud Computing

INSTRUCTOR(S): Arun Venkataramani 3 Credits

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.

### COMPSCI 590D Algorithms for Data Science

INSTRUCTOR(S): Barna Saha, David Wemhoener 3 Credits

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 (BS/BA). Undergraduate Prerequisites: COMPSCI 240 and COMPSCI 311. 3 credits.

### COMPSCI 590E Ethical Considerations in Computing

INSTRUCTOR(S): Michelle Trim 3 Credits

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. Guest speakers will include CS industry professionals, and course assignments will have real world applications. Class discussions will be a vibrant component of the course. 3 credits.

### COMPSCI 590N Introduction to Numerical Computing with Python

INSTRUCTOR(S): Roy Adams 1 Credits

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. 1 credit.

### COMPSCI 590R Applied Information Retrieval

INSTRUCTOR(S): David Fisher 3 Credits

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. This course counts as a CS Elective toward the CS major (BS/BA). 3 credits.

### COMPSCI 590S Systems for Data Science

INSTRUCTOR(S): Emery Berger 3 Credits

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. This course counts as a CS Elective toward the CS major (BS/BA). Undergraduate Prerequisites: COMPSCI 311, COMPSCI 345, and COMPSCI 377. 3 credits.

### COMPSCI 611 Advanced Algorithms

INSTRUCTOR(S): Andrew McGregor 3 Credits

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 631 Programming Languages

INSTRUCTOR(S): Arjun Guha 3 Credits

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

INSTRUCTOR(S): Charles Weems 3 Credits

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 653 Computer Networking

INSTRUCTOR(S): Phillipa Gill 3 Credits

This course covers advanced fundamental principles of computer networks, studying foundational material in the field. Topics include advanced network architecture, network algorithmics, network control, network measurement, and wireless networks. The goal of this course is to teach networking fundamentals/techniques that will be useful for years to come. Prerequisites: Introductory (undergraduate level) courses in computer networks (e.g., COMPSCI 453), and algorithms (e.g., COMPSCI 311). Some familiarity with probability and with optimization theory will be helpful. 3 credits.

### COMPSCI 670 Computer Vision

INSTRUCTOR(S): Subhransu Maji 3 Credits

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 3 Credits

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. Some assignments will be in Python and some in C++. 3 credits.

### COMPSCI 687 Reinforcement Learning

INSTRUCTOR(S): Philip Thomas 3 Credits

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 (including deep 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 (assignments will use C++, but familiarity with C++ specifically will not be assumed). 3 credits.

### COMPSCI 689 Machine Learning

INSTRUCTOR(S): Benjamin Marlin 3 Credits

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 690LG Advanced Logic in Computer Science

INSTRUCTOR(S): Neil Immerman 3 Credits

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 690M Machine Learning Theory

INSTRUCTOR(S): Akshay Krishnamurthy 3 Credits

When, how, and why do machine learning algorithms work? This course answers these questions by studying the theoretical aspects of machine learning, with a focus on statistically and computationally efficient learning. Broad topics will include: PAC-learning, uniform convergence, and model selection; supervised learning algorithms including SVM, boosting, kernel methods; online learning algorithms and analysis; unsupervised learning with guarantees. Special topics may include: Bandits, active learning, semi-supervised learning and others. 3 credits.

### COMPSCI 690V Visual Analytics

INSTRUCTOR(S): Georges Grinstein 3 Credits

In this course, students will work on solving complex problems in data science using exploratory data visualization and analysis in combination. Students will learn to deal with the Five V's: Volume, Variety, Velocity, Veracity, and Variability, that is with large data, complex heterogeneous data, streaming data, uncertainty in data, and variations in data flow, density and complexity. Students will be able to select the appropriate tools and visualizations in support of problem solving in different application areas. The course is a practical continuation of COMPSCI 590V - Data Visualization and Exploration and focuses on complex problems and applications, however 590V is not a prerequisite and both 590V and 690V may be taken independently of each other. The data sets and problems will be selected mainly from the IEEE VAST Challenges, but also from the KDD CUP, Amazon, Netflix, GroupLens, MovieLens, Wiki releases, Biology competitions and others. We will solve crime, cyber security, health, social, communication, marketing and similar large-scale problems. Data sources will be quite broad and include text, social media, audio, image, video, sensor, and communication collections representing very real problems. Hands-on projects will be based on Python or R, and various visualization libraries, both open source and commercial. 3 credits.

### COMPSCI 691DD Seminar - Research Methods in Empirical Computer Science

INSTRUCTOR(S): David Jensen 3 Credits

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 691NN Seminar - Hot Topics in Programming Languages and Systems

INSTRUCTOR(S): René Just, Arjun Guha 1 -3 Credits

This graduate seminar course will cover recent developments in programming languages and systems, examining the latest research papers from top programming languages and systems conferences. Topics of interest include bug detection and correction, domain-specific languages, and emerging topics, including probabilistic and approximate computing. 3 credits with project, 1 credit without.

### COMPSCI 691O Seminar - Tools for Explanatory and Tutoring Systems

INSTRUCTOR(S): Beverly Woolf 3 Credits

This seminar examines recent work in explanatory and tutoring systems. Participants study artificial intelligence in education, a young field that explores theories about learning, and explores how to build software that delivers differential teaching as it adapt its response to student needs and domain knowledge. Such software supports people who work alone or in collaborative inquiry, students who question their own knowledge, and students who rapidly access and integrate global information. This course describes how to build these tutors and how to produce the best possible learning environment, whether for classroom instruction or lifelong learning. The objective of the course is to stimulate awareness of research issues and to promote sound analytic and design skills as they pertain to building knowledge representations and control strategies. Specific topics include collaboration, inquiry, dialogue systems, machine learning, simulators, authoring tools and user models. The course is appropriate for students from many disciplines (computer science, linguistics, education, and psychology), researchers, and practitioners from academia, industry, and government. No programming is required. Students read and critique tools, methods, and ideas, learn how artificial intelligence is applied (e.g., vision, natural language), and study the complexity of human learning through advances in cognitive science. Weekly assignments invite students to critique the literature and a final project requires a detailed specification (not a program) for a tutor about a chosen topic. Students present book chapter material weekly in parallel with readings from the research literature. Several working systems will be available for hands-on critique. 3 credits.

### COMPSCI 691Q Seminar - Quantum Computing

INSTRUCTOR(S): Donald Towsley 1 -3 Credits

This seminar will cover topics in quantum information science, including both classical ones such as quantum key distribution and Grover's algorithm, and the emerging area of quantum machine learning. 1-3 credits.

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

INSTRUCTOR(S): Sunghoon Lee 1 -3 Credits

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 701 Advanced Topics in Computer Science

INSTRUCTOR(S): 6 Credits

This is a 6 credit reading course corresponding to the master's project. The official instructor is the GPD although the student does the work with and is evaluated by the readers of his or her master's project.

### COMPSCI 890T Teaching Assistants as Tomorrow's Faculty

INSTRUCTOR(S): Neena Thota 1 Credits

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.

### COMPSCI 891M Seminar - Theory of Computation

INSTRUCTOR(S): Neil Immerman 1 Credits

The theory seminar is a weekly meeting in which topics of interest in the theory of computation - broadly construed - are presented. This is sometimes new research by visitors or local people. It is sometimes work in progress, and it is sometimes recent material of others that some of us present in order to learn and share. This is a one-credit seminar which may be taken repeatedly for credit up to six times. 1 credit.

### COMPSCI H320 Honors Colloquium for Software Engineering

INSTRUCTOR(S): Leon Osterweil 1 Credits

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

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

INSTRUCTOR(S): Charles Weems 1 Credits

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. In addition to the regular requirements of COMPSCI 335, 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). 4 credits.

### COMPSCI H345 Honors Colloquium for Practice and Applications of Data Management

INSTRUCTOR(S): Alexandra Meliou 1 Credits

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 H377 Honors Colloquium for Operating Systems

INSTRUCTOR(S): Timothy Richards 1 Credits

Students taking the honors section will be required to do additional readings, a semester-long project, and write a conference-style paper to be submitted at the end of the semester. 1 credit.

### INFO 101 Introduction to Informatics

INSTRUCTOR(S): Michelle Trim 3 Credits

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

### INFO 150 A Mathematical Foundation for Informatics

INSTRUCTOR(S): David Barrington 3 Credits

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.

### INFOSEC 690R Information Risk Management

INSTRUCTOR(S): Amanda Andress 3 Credits

This course reviews information governance, information technology policy, privacy and security, regulatory compliance and project management as a comprehensive program applicable to any corporate network system. Project management tailored to information security and privacy will also be a centerpiece of the case study and course content.

### INFOSEC 690S System Defense and Test

INSTRUCTOR(S): Parviz Kermani 3 Credits

This class trains students to detect and analyze weaknesses and vulnerabilities in target systems as a method of assessing the security of a system. We focus on tools and techniques that an attacker would employ but from the perspective of an ethical system administrator. Topics include tools and techniques for penetration testing and attacks, information gathering, social engineering, and defenses. Specific topics include malware, denial of service attacks, SQL injection, buffer overflow, session hijacking, and system hacking, network sniffing and scans, wireless encryption weaknesses and other WiFi issues, IDS/firewall evasion, metasploit tools, physical security, and setting up honeypots. 3 credits.