CICS 109 Intro to Data Analysis in R

INSTRUCTOR(S): Jasper McChesney

An introduction to data analysis in the open-source R language, with an emphasis on practical data work. Topics will include data wrangling, summary statistics, modeling, and visualization. Will also cover fundamental programming concepts including data types, functions, flow of control, and good programming practices. Intended for a broad range of students outside of computer science. Some familiarity with statistics is expected. 1 credit.

CICS 110 Foundations of Programming

INSTRUCTOR(S): Eduardo Calle Ortiz, Jakob Falus, Allison Poh, Cole Reilly, Jared Yeager

An introduction to computer programming and problem solving using computers. This course teaches you how real-world problems can be solved computationally using programming constructs and data abstractions of a modern programming language. Concepts and techniques covered include variables, expressions, data types, objects, branching, iteration, functions, classes, and methods. We will also cover how to translate problems into a sequence of instructions, investigate the fundamental operation of a computational system and trace program execution and memory, and learn how to test and debug programs. No previous programming experience required. (Gen. Ed. R2) 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.

CICS 160 Object-Oriented Programming

INSTRUCTOR(S): Jaime Davila

This course will expose students to programming practices beyond the introductory level, concentrating on Object Oriented Programming techniques and an introduction to Data Structures. Students will also study and analyze the complexity of both the algorithms presented in class and of the algorithms they develop. This course also provides experience with the development and analysis of recursive algorithms and programs. Before taking this course, students are expected to have been exposed to the following concepts through a college-level course or equivalent in some high level computer programming language: input and output operations, conditional statements, loops, arrays, recursion, and functions/methods. The course places an emphasis on the careful design and testing of programs. Prerequisites: CICS 110 OR INFO 190S. 4 credits.

CICS 210 Data Structures

INSTRUCTOR(S): Marc Liberatore

An introduction to the design, analysis, and implementation of data structures. This course teaches you how to build, test, debug, document, and evaluate objects that encapsulate data and their associated operations using programming constructs and data abstractions of a modern programming language. Concepts and techniques covered include linear and non-linear structures, recursive structures and algorithms, traversal algorithms, binary search trees, balanced trees, priority queues, union-find, hash tables, bloom filters, and graphs. We will also informally compare and contrast the run time efficiency of algorithms and their performance characteristics including the concept of worst-case running time analysis and the classification of algorithms in terms of constant, logarithmic, linear, log linear, quadratic, and exponential time using Big-O notation. Prerequisite: CICS 160 (or INFO 190T). 4 credits.

CICS 256 Make: Physical Computing

INSTRUCTOR(S): Donghyun Kim, Md Farhan Tasnim Oshim

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 290P Intro to Public Interest Tech

INSTRUCTOR(S): Francine Berman

Today's world is complex and tech-driven. How do we use the tools of information technology to solve problems in a socially responsible way, i.e. in a way that both empowers us and promotes the well-being of the communities in which we live? In this course, we describe the socio-technical world and pragmatic strategies for promoting personal and social responsibility. We explore the questions: What is the public interest in a socio-technical world? What strategies can we use to promote social responsibility in the public sector, private sector and general public? What can each of us do to make the world a better place? 3 credits.

CICS 291T S-CICS Transfer Success

INSTRUCTOR(S): Emma Anderson

This seminar is intended to help you become fully prepared to succeed in CICS at UMass. Students in this seminar will be led by an instructor with a detailed understanding of the transfer student experience, and supported by various staff members in CICS. You will learn about which campus and College resources will be most helpful to you, how to best utilize these resources, and where you can look for other opportunities to connect. 1 credit.

CICS 305 Social Issues in Computing

INSTRUCTOR(S): Erin Butler, Oscar Garcia, Elizabeth Gunther, Siobhan Meï, Justin Obara, Christina Sutcliffe, Michelle Trim

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: CS Majors: ENGLWRIT 112 (or English Writing waiver), COMPSCI 220, COMPSCI 230 and COMPSCI 240 (or 250); INFORM Majors: ENGLWRIT 112 (or English Writing waiver) AND INFO 248. 3 credits.

COMPSCI 119 Intro to Programming

INSTRUCTOR(S): Cole Reilly

A complete introduction to computer programming using the Python language. Topics include coverage of all the supported data types and program code structures, functions (up through lambda expressions and recursion), reasoning about and debugging existing code, implementation of custom libraries, selection of data structures, and the fundamentals of object-oriented programming. Students will create, debug, and run Python 3 programs that explore each of these topics in turn, from simple loops up through the processing of large data sets, and eventually to the creation of professional-quality libraries to synthesize graphics images and audio files. No prior programming experience expected. Not open to Computer Science majors. 3 credits.

COMPSCI 187 Programming w;Data Structures

INSTRUCTOR(S): 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 198C P-Intro;C Programming Language

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

This practicum assumes general background and experience in computer programming (such as that provided by COMPSCI 121 or a similar introductory programming course) and some knowledge of data structures. Content will include basic C data types, declarations, expressions, statements, and functions; simple use of macros; some common library calls (such as formatted input/output); basic pointer manipulation using linked lists; and introduction to using standard tools (gcc and make). It is recommended that students planning to take COMPSCI 230 take this class along with COMPSCI 187. Prerequisite: COMPSCI 121 or COMPSCI 186 or INFO 190T/CICS 160. 1 credit.

COMPSCI 220 Programming Methodology

INSTRUCTOR(S): Marius Minea

Development of individual skills necessary for designing, implementing, testing and modifying larger programs, including: design strategies and patterns, using functional and object-oriented approaches, testing and program verification, code refactoring, interfacing with libraries. There will be significant programming and mid-term and final examinations. Prerequisite: COMPSCI 187. 4 credits.

COMPSCI 230 Computer Systems Principles

INSTRUCTOR(S): 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 data representation, and moves up the stack from there to the features of modern architectures, assembly languages, and operating system services such as I/O, process, and synchronization. This class assumes students have either taken COMPSCI 198C or have equivalent experience in the C programming language. Prerequisite: COMPSCI 187. 4 credits.

COMPSCI 240 Reasoning Under Uncertainty

INSTRUCTOR(S): Shiting Lan, Sunghoon Lee

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 (or INFO 190T/CICS 160) and MATH 132. 4 credits.

COMPSCI 250 Introduction To Computation

INSTRUCTOR(S): David Barrington, Ghazaleh Parvini

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: COMPSCI 187 (or ECE 241 or INFO 190T/CICS 160) and MATH 132. 4 credits.

COMPSCI 311 Introduction to Algorithms

INSTRUCTOR(S): Marius Minea, Ghazaleh Parvini

This course will introduce you to a variety of techniques to design algorithms, such as divide and conquer, greedy, dynamic programming, and network flow. 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 and recognize problems that currently do not have efficient algorithms. Assignments may include programming: you should be able to program in Java, C, or some other closely related language. Mathematical experience (as provided by COMPSCI 250) is required. This course is required for the CS Major (BS) and counts as an Elective toward the CS Major (BA). Prerequisite: COMPSCI 187 and either COMPSCI 250 or MATH 455. 4 credits.

COMPSCI 320 Software Engineering

INSTRUCTOR(S): Gordon Anderson, Heather Conboy, Jaime Davila, Matthew Rattigan

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 for CS Major. Prerequisite: COMPSCI 220. 4 credits.

COMPSCI 325 Intro;Human-Comp Interaction

INSTRUCTOR(S): Cheryl Swanier

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. This course counts as an Elective toward the CS and INFORM Majors. Prerequisite: COMPSCI 186 (or COMPSCI 187). 3 credits.

COMPSCI 326 Web Programming

INSTRUCTOR(S): Timothy Richards

The web is arguably today's most important application platform. Web browsers run on practically every device, and even many phone applications are in fact web applications under the covers. This course will cover a broad range of client-side web technologies, including HTTP itself, HTML5, CSS, and JavaScript; it will additionally cover key concepts for the server side of web applications, including key value stores and SQL servers. This course will also cover key concepts and technologies including AJAX, JavaScript libraries (e.g., jQuery), and web security. This course is hands-on and heavily project-based; students will construct a substantial dynamic web application based on the concepts, technologies, and techniques presented during lectures and in readings. This course satisfies the IE Requirement for CS and INFORM Majors. Prerequisites: CS Majors: COMPSCI 220 (or COMPSCI 230); INFORM Majors: INFO 248 and (COMPSCI 186 or COMPSCI 187 or INFO 190T/CICS 160). Note: as the name 'web programming' denotes, programming is a key component of this class. Previous background in JavaScript is strongly recommended. 4 credits.

COMPSCI 328 Mobile Health Sensing+Analytcs
INSTRUCTOR(S): Deepak Ganesan

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. Prerequisite: CS Majors: COMPSCI 187; INFORM Majors: INFO 248 and (COMPSCI 186 or COMPSCI 187 or INFO 190T/CICS 160). 3 credits.

COMPSCI 345 Pract + Appl of Data Managemnt
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 an Elective toward the CS and INFORM Major. Students who have completed COMPSCI 445 are not eligible to take this course without instructor permission. Prerequisite: CS Majors: COMPSCI 187; INFORM Majors: INFO 248 and (COMPSCI 186 or COMPSCI 187 or INFO 190T/CICS 160). 3 credits.

COMPSCI 370 Intro to Computer Vision
INSTRUCTOR(S): Subhransu Maji

This introductory computer vision class will address fundamental questions about getting computers to "see" like humans. We investigate questions such as -What is the role of vision in intelligence? -How are images represented in a computer? -How can we write algorithms to recognize an object? -How can humans and computers "learn to see better" from experience? We will write a number of basic computer programs to do things like recognize handwritten characters, track objects in video, and understand the structure of images. Prerequisite: COMPSCI 240 or 383. 3 credits.

COMPSCI 373 Intro to Computer Graphics
INSTRUCTOR(S): Zachary While

This course introduces the fundamental concepts of 2D and 3D computer graphics. It covers the basic methods for modeling, rendering, and imaging. Topics include: image processing, 2D/3D modeling, 3D graphics pipeline, WebGL, shading, texture mapping, ray tracing, 3D printing. Throughout the class, we will teach students to learn modern graphics techniques, to model the visual world algorithmically, and to implement algorithms using JavaScript. Students who have taken COMPSCI 473 are not eligible to take this course. Students cannot take COMPSCI 497C after taking this course. This course counts as a CS Elective toward the CS Major. Prerequisites: COMPSCI 187 and MATH 235 (or INFO 150 or COMPSCI 240). 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. This course counts as a CS Elective toward the CS Major. Prerequisites: COMPSCI 230. 4 credits.

COMPSCI 383 Artificial Intelligence

INSTRUCTOR(S): Scott Niekum

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. Students should be comfortable programming in Python. This course counts as a CS Elective toward the CS Major. Prerequisites: COMPSCI 220 (or COMPSCI 230) and COMPSCI 240 (or STAT 515). 3 credits.

COMPSCI 389 Intro to Machine Learning

INSTRUCTOR(S): Cooper Sigrist

The course provides an introduction to machine learning algorithms and applications. Machine learning algorithms answer the question: "How can a computer improve its performance based on data and from its own experience?" The course is roughly divided into thirds: supervised learning (learning from labeled data), reinforcement learning (learning via trial and error), and real-world considerations like ethics, safety, and fairness. Specific topics include linear and non-linear regression, (stochastic) gradient descent, neural networks, backpropagation, classification, Markov decision processes, state-value and action-value functions, temporal difference learning, actor-critic algorithms, the reward prediction error hypothesis for dopamine, connectionism for philosophy of mind, and ethics, safety, and fairness considerations when applying machine learning to real-world problems. This course counts as an Elective toward the CS and INFORM Majors. Prerequisite: COMPSCI 220 (or COMPSCI 230), COMPSCI 240 (or STAT 515), and MATH 233. 3 credits.

COMPSCI 390R ReverseEngin+VulnrbltyAnalysis

INSTRUCTOR(S): Brian Levine

Many software developers aren't aware of how to properly write secure code. This course covers practical skills in reverse engineering and binary exploitation, and examines the techniques used by hackers in recent major security incidents. The course objective is to provide students with a strong understanding of attack patterns, and to ensure students implement more secure coding practices in their own code. This course begins with an introduction to Intel-based assembly, reverse engineering, vulnerability analysis, and various forms of Linux-focused binary exploitation. The course then covers stack, heap and Linux kernel-based exploitation, and dive into common defensive mitigations such as ASLR, NX and Stack Cookies alongside techniques to bypass each of them. This course is focused on low-level software written in C. COMPSCI 230 is sufficient for demonstrating knowledge of C and that the student has been introduced to assembly. Students who have taken 198C (or can demonstrate a proficiency in C) and can demonstrate a familiarity with assembly can request an override from the instructor. This course counts as an Elective for the CS Major. Prerequisites: COMPSCI 230 (or E&C-ENG 322 or E&C-ENG 373) or permission of instructor. 3 credits.

COMPSCI 420 Software Entrepreneurship

INSTRUCTOR(S): Neena Thota

This course is geared towards students interested in developing software that moves from early stage proof-of-concept ideas towards marketable products with societal benefit. The course leverages the expertise of the Entrepreneurs in Residence (EIR) of the Ventures @ CICS initiative at CICS. The course is grounded in Challenge Based Learning (CBL), an active, student-directed instructional framework that was developed by Apple Inc. and educators. This course counts as an Elective toward the CS major. Prerequisites: COMPSCi 320 (or COMPSCI 326). 3 credits.

COMPSCI 429 Software Engin Proj Management

INSTRUCTOR(S): Gordon Anderson, Heather Conboy, Jaime Davila, Matthew Rattigan

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 429 will be responsible for: supervising and managing the work of teams of COMPSCI 320 students; interfacing with the other COMPSCI 429 students managing other teams in the course; interfacing with the course instructor, course TA, and course customer. COMPSCI 429 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 429 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 429 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 429 are also expected to be arranged by mutual agreement.   An additional one hour weekly meeting of all of the students in COMPSCI 429 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 445 Information Systems

INSTRUCTOR(S): Marco Serafini

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. This course counts as an Elective toward the CS Major. Prerequisites: COMPSCI 220 (or 230) and COMPSCI 311 and COMPSCI 345. 3 credits.

COMPSCI 446 Search Engines

INSTRUCTOR(S): James Allan

This course provides an overview of the important issues in information retrieval, and how those issues affect the design and implementation of search engines. The course emphasizes the technology used in Web search engines, and the information retrieval theories and concepts that underlie all search applications. Mathematical experience (as provided by COMPSCI 240) is required. You should also be able to program in Java (or some other closely related language). This course counts as an Elective toward the CS Major. Prerequisite: COMPSCI 240 or COMPSCI 383. 3 credits.

COMPSCI 453 Computer Networks

INSTRUCTOR(S): Phuthipong Bovornkeeratiroj, James Kurose

Introduction to computer communication networks and protocols. Fundamental concepts in the design and analysis of computer networks. Topics include: layered network architectures, networked applications, network programming interfaces, transport, congestion, routing, data link protocols, local area and data center 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. In Fall 2020, this course will be taught as a primarily online class, without in-class lectures, but with extensive on-line material including recorded video material, interactive exercises, and online discussion. There will be an optional weekly, one-hour in-class and on-line open discussion period. This course counts as an Elective toward the CS Major. Prerequisites: Experience programming; COMPSCI 230 or COMPSCI 377. 3 credits.

COMPSCI 466 Applied Cryptography

INSTRUCTOR(S): Adam O'Neill

This is an undergraduate-level introduction to cryptography. It is a theory course with a significant mathematical component. However, our viewpoint will be "theory applied to practice" in that we will aim to treat topics in a way of applied value. We will discuss cryptographic algorithms 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 principles and uncover design weaknesses. The primary topics are: blockciphers, pseudorandom functions, symmetric-key encryption schemes, hash functions, message authentication codes, public-key encryption schemes, digital signature schemes, and public-key infrastructures. This course counts as an Elective toward the CS Major. Prerequisites: COMPSCI 311. 3 credits.

COMPSCI 490Q Quantum Information Science

INSTRUCTOR(S): Stefan Krastanov

Quantum information science (QIS) revolutionizes our understanding of the fundamental laws of the universe and promises world-altering improvements in a number of practical computational tasks. For theoretical computer scientists, QIS provides the means to unlock the ultimate computational powers available to us in this universe. For programmers and computer engineers, QIS offers the tools to run simulations and optimizations otherwise infeasible on classical computers. For theoretical physicists, QIS gives us hope of resolving paradoxes foundational to our understanding of Nature. And for experimentalists and engineers, QIS also enables the creation of exquisite sensors and novel communication tools, far outperforming what is classically permitted. This class will introduce the notion of quantum probability amplitudes, i.e., the "correct" probabilistic description of Nature, and describe how these quantum phenomena permit the creation of new types of computational machines. The introduction to foundational quantum information science will be followed by a few practical (and impractical) quantum algorithms, illustrating the counterintuitive computational powers of quantum mechanics. The latter half of the class would focus on the difficulties of creating such extremely fragile computational machines in our noisy and unforgiving real world. Prerequisites: MATH 132, MATH 235, and either COMPSCI 240 or STATISTC 515. This course counts as an elective for both CS and Informatics Majors. 3 credits.

COMPSCI 490U Intro to UX Research

INSTRUCTOR(S): Ya Xiong

As technology becomes more and more integrated with our work and life, it is increasingly critical to consider how we can better design technology to enhance the experience and behavior of its users. UX Research, or User Experience Research, is an interdisciplinary field that combines knowledge from computer science, design, statistics, and psychology to gather insights about the needs, attitudes, and behaviors of technology users to inspire and inform design. This course introduces you to the cycle of UX research and aims to prepare you to tackle real-world UX research challenges. You will learn about the principles and practices of UX research, how to conduct user experiments to explore user behaviors and motivations, how to use statistical software to analyze data, and how to present your findings in an effective manner. This course counts as an Elective toward the CS and INFORM Majors. Prerequisites: COMPSCI 325. 3 credits.

COMPSCI 491G S-Computer Networking Lab

INSTRUCTOR(S): Arun Venkataramani

In this course, students will learn how to put "principles into practice," in a hands-on-networking lab course. The course will cover router, switches and end-system labs in the areas of Single Segment IP Networks, Multiple Segment IP Networks and Static Routing, Dynamic Routing Protocols (RIP, OSPF and BGP), LAN switching, Transport Layer Protocols: UDP and TCP, NAT, DHCP, DNS, and SNMP. Students will also get engaged in evaluating power consumption of network components as an aid in the design of energy efficient (green) networks. This course counts as an Elective toward the CS Major. Prerequisite: COMPSCI 453. 3 credits.

COMPSCI 501 Formal Language Theory

INSTRUCTOR(S): David Barrington

Introduction to formal language theory. Topics include finite state languages, context-free languages, the relationship between language classes and formal machine models, the Turing Machine model of computation, theories of computability, resource-bounded models, and NP-completeness. This course counts as an Elective toward the CS Major. Undergraduate Prerequisites: COMPSCI 311 or equivalent. It is recommended that students have a 'B-' or better in 311 in order to attempt 501. 3 credits.

COMPSCI 508 Ethical Consideratns;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. Undergraduate CS Majors with permission of instructor (counts as an Elective toward the CS Major). 3 credits.

COMPSCI 514 Algorithms for Data Science

INSTRUCTOR(S): Andrew McGregor

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. This course counts as an Elective toward the CS Major. Undergraduate Prerequisites: COMPSCI 240 (or STATISTCS 515) and COMPSCI 311 both with a grade of B+ or better, OR (COMPSCI 240 and STATISTCS 515 and COMPSCI 311 and MATH 233 and MATH 235, all with a C or better). 3 credits

COMPSCI 520 Thry + Practice;Software Engin

INSTRUCTOR(S): Heather Conboy

Introduces students to the principal activities and state-of-the-art techniques involved in developing high-quality software systems. Topics include: requirements engineering, formal specification methods, design principles & patterns, verification & validation, debugging, and automated software engineering. This course counts as an Elective toward the CS Major. Undergraduate Prerequisites: COMPSCI 320. 3 credits.

COMPSCI 532 Systems for Data Science

INSTRUCTOR(S): Peter Klemperer

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 an Elective toward the CS Major. Undergraduate Prerequisites: COMPSCI 377 and COMPSCI 445. 3 credits.

COMPSCI 535 Computer Architecture

INSTRUCTOR(S): Charles Weems

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

COMPSCI 546 Applied Information Retrieval

INSTRUCTOR(S): Razieh Rahimi

COMPSCI 546 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 and assignments. 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 (e.g. basic deep learning models for information retrieval); and many more. This course counts as an Elective toward the CS Major. Undergraduate Prerequisites: COMPSCI 320 and either COMPSCI 383, COMPSCI 446, or COMPSCI 585. 3 credits.

COMPSCI 550 Introduction to Simulation

INSTRUCTOR(S): Peter Haas

How can we use computers to design systems and, more generally, make decisions, in the face of complexity and uncertainty? Simulation techniques apply the power of the computer to study complex stochastic systems when analytical or numerical techniques do not suffice. It is the most frequently used methodology for the design and evaluation of computer, telecommunication, manufacturing, healthcare, financial, and transportation systems, to name just a few application areas. Simulation is an interdisciplinary subject, incorporating ideas and techniques from computer science, probability, statistics, optimization, and number theory. Simulation models, which embody deep domain expertise, can effectively complement machine-learning approaches. This course will provide the student with a hands-on introduction into this fascinating and useful subject. This course counts as an Elective toward the CS Major. Undergraduate Prerequisite: COMPSCI 187 and STAT 515. 3 credits.

COMPSCI 561 System Defense and Test

INSTRUCTOR(S): Parviz Kermani

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. Was INFOSEC 690S. This course counts as an Elective toward the CS Major. Undergraduate Prerequisites: COMPSCI 360 (or COMPSCI 460 or COMPSCI 560/597N or COMPSCI 660) and COMPSCI 453. 3 credits.

COMPSCI 564 Cyber Effects

INSTRUCTOR(S): Seth Landsman, Nick Merlino, Daniel Walters, Edward Walters

This course covers a broad range of topics related to cyber security and operations. Our focus is on real world studies of reverse engineering, exploit analysis, and capability development within the context of computer network operations and attack. The course has an emphasis on hands-on exercises and projects. Topics covered include computer architecture and assembly language, principles of embedded security, the essentials of exploit development and analysis (including using industry standard tools such as Ghidra, and utilizing computer security databases such as CVE), and discussion of real-world events and techniques. This course counts as an Elective toward the CS Major. Undergraduate Prerequisite: COMPSCI 230 (or ECE 322) and COMPSCI 360 (or COMPSCI 460 or ECE 371). 3 credits.

COMPSCI 565 Adv Digital Forensic Systems

INSTRUCTOR(S): Peter Klemperer

This course introduces students to the principal activities and state-of-the-art techniques involved in developing digital forensics systems. Topics covered may include: advanced file carving and reconstruction, forensic analysis of modern filesystems, network forensics, mobile device forensics, memory forensics, and anti-forensics. This course counts as an Elective toward the CS Major. Undergraduate Prerequisites: COMPSCI 365 (or COMPSCI 377). 3 credits.

COMPSCI 574 Intelligent Visual Computing

INSTRUCTOR(S): Evangelos Kalogerakis

Intelligent visual computing is an emerging new field that seeks to combine modern trends in machine learning, computer graphics, computer vision to intelligently process, analyze and synthesize 2D/3D visual data. The course will start by covering 2D image and 3D shape representations, classification and regression techniques, and the fundamentals of deep learning. The course will then provide an in-depth background on analysis and synthesis of images and shapes with deep learning, in particular convolutional neural networks, recurrent neural networks, memory networks, auto-encoders, adversarial networks, reinforcement learning methods, and probabilistic graphical models. Students will complete 5 programming assignments in Matlab/Octave and work on a course project related to visual computing with machine learning. This course counts as a CS Elective toward the CS major (BA/BS). Course was previously COMPSCI 590IV. Undergraduate Prerequisites:COMPSCI 311, COMPSCI 383, COMPSCI 373 (or COMPSCI 473). 3 credits.

COMPSCI 589 Machine Learning

INSTRUCTOR(S): Bruno Castro da Silva

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 effectively using machine learning methods to solve real-world problems with an emphasis on model selection, regularization, and empirical evaluation. The assignments will involve both mathematical problems and implementation tasks. Knowledge of a high-level programming language is absolutely necessary. Python is most commonly used (along with standard libraries such as numpy, scipy, and scikit-learn), but languages such as Matlab, R, Scala, Julia would also be suitable. While this course has an applied focus, it still requires appropriate mathematical background in probability and statistics, calculus, and linear algebra. The prerequisites for undergrads were previously COMPSCI 383 and MATH 235 (COMPSCI 240 provides sufficient background in probability, and MATH 131/132 provide sufficient background in calculus). Graduate students can check the descriptions for these courses to verify that they have sufficient mathematical background for 589. Strong foundations in linear algebra, calculus, probability, and statistics are essential for successfully completing this course. Graduate students from outside computer science with sufficient background are also welcome to take the course. This course counts as an Elective toward the CS Major. Undergraduate Prerequisites: MATH 545 and COMPSCI 240 and STATS 515 C or better. MATH 545 can be skipped by students who have taken MATH 235 and MATH 233 both with B+ or better. STATS 515 can be skipped by students who have taken COMPSCI 240 with a B+ or better. 3 credits.

COMPSCI 591NR S-Neural Networks:Neurosci+Eng

INSTRUCTOR(S): Hava Siegelmann

This is a project based course, focusing on the science connecting the field of neural network with human brain mechanism, as well as advancements which are at the front of the field. We start by introducing a few neural network architectures with their learning paradigms, including deep feedforward and recurrent networks, Hopfield's memory, and Kohonen's self organizing maps. We continue by building on the top of these: clique structures and brain-like updatable architectures, explainable AI, combining symbolic with subsymbolic into one strong AI, robust AI, deep fake applications, and up to lifelong learning. The course is project based – with grades based on active class participation, presenting non-trivial topics, research project that will be done during the semester, and research paper written based on this work. I'll assist students to get their research submitted to publications, if their work will reach high level. Students in 591NR will be able to run simpler projects. COMPARISON BETWEEN COMPSCI 682 AND 591NR/691NR: 682 teaches the engineering techniques necessary to train current neural networks. 591NR/691NR provides far larger view of the field, focusing on the science – both relation with human and natural intelligence, and on the advancements which are in the forefront of the neural networks field. 3 credits. This course counts as an Elective toward the CS Major. Undergraduate Prerequisites: COMPSCI 311. 3 credits.

COMPSCI 603 Robotics

INSTRUCTOR(S): Hao Zhang

This course is intended to serve as an advanced overview of robotics spanning the complete autonomy loop: perception, planning, and control. We will study the theory, algorithms, and efficient implementations related to these topics, with focus on open discussions for how to do research to go beyond the state of the art. Students will gain hands-on experience in implementing, and extending such algorithms using simulations. 3 credits.

COMPSCI 611 Advanced Algorithms

INSTRUCTOR(S): Hung Le

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 621 Adv S;W Eng: Analysis and Eval

INSTRUCTOR(S): Yuriy Brun

Software has become ubiquitous in our society. It controls life-critical applications, such as air traffic control and medical devices, and is of central importance in telecommunication and electronic commerce. In this course, we will examine state-of-the-art practices for software testing and analysis to verify software quality. We will initially look at techniques for testing and analyzing sequential programs, and then examine the complexity that arises from distributed programs. The students will be required to complete regular homework assignments and exams, and carry out a group research project extending techniques described in class and/or applying them to new domains. No required text. Papers from the open literature will be assigned and made available. Prerequisites: COMPSCI 320, Introduction to Software Engineering (or equivalent course). 3 credits.

COMPSCI 625 Advanced Methods in HCI

INSTRUCTOR(S): Narges Mahyar

This is an advanced course in HCI. This course will provide a deeper treatment of some topics that are typically found in an undergraduate HCI course. For example, design methodologies, evaluation methodologies (both quantitative and qualitative), human information processing, cognition, and perception. This course will also introduce students to research frontiers in HCI. The course will cover topics of Universal Usability, CSCW, Digital Civics and fundamentals of designing interactive technology for people. 3 credits.

COMPSCI 627 Fixing Social Media

INSTRUCTOR(S): Ethan Zuckerman

Over the past decade, user-generated participatory media – social media – has emerged as the dominant model for content of the Internet. From Facebook to Twitter, YouTube to Wikipedia, content created by non-professionals and circulated for commercial and non-commercial motives underpins seven of the top 10 websites in the US, and has become an increasingly important component of the news ecosystem. While social media was initially hailed as a powerful tool for broadening civic participation, many problems have emerged with the rise of the medium, from questions of whether social media usage is bad for our individual mental health, to whether the fabric of our democracy is being damaged by disinformation, fragmentation and hyperpolarization. As legislators look to regulate these platforms and commentators propose shutting them down entirely, this course looks for an alternative: affirmative visions of social media that are good for individuals and society, which we could work towards building. This class examines possible problems with existing modes of social media, discusses ways in which social media could be a benefit to individuals and societies, develops case studies of successful and healthy online communities, and ultimately designs and builds tools to improve existing social media systems or replace them with novel models. Students will write reflectively about weekly readings and discussions and participate in multi-week projects, ultimately building teams to work on final projects. Meets with COMM 627 and SPP 627. 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 651 Optimization;Computer Science

INSTRUCTOR(S): Madalina Fiterau Brostean

Much recent work in computer science in a variety of areas, from game theory to machine learning and sensor networks, exploits sophisticated methods of optimization. This course is intended to give students an in-depth background in both the foundations as well as some recent trends in the theory and practice of optimization for computer science. The Optimization course covers these topics, which are critical to a large number of research projects conducted within the department. 3 credits.

COMPSCI 674 Intelligent Visual Computing

INSTRUCTOR(S): Evangelos Kalogerakis

Intelligent visual computing is an emerging new field that seeks to combine modern trends in machine learning, computer graphics, computer vision to intelligently process, analyze and synthesize 2D/3D visual data. The course will start by covering 2D image and 3D shape representations, classification and regression techniques, and the fundamentals of deep learning. The course will then provide an in-depth background on analysis and synthesis of images and shapes with deep learning, in particular convolutional neural networks, recurrent neural networks, memory networks, auto-encoders, adversarial networks, reinforcement learning methods, and probabilistic graphical models. Students will complete 5 programming assignments in Matlab/Octave and work on a course project related to visual computing with machine learning. Course was previously COMPSCI 690IV. 3 credits.

COMPSCI 677 Distributed+Operating Systems

INSTRUCTOR(S): Prashant Shenoy

This course provides an in-depth examination of the principles of distributed systems and advanced concepts in operating systems. Covered topics include client-server programming, distributed scheduling, virtualization, cloud computing, distributed storage, security in distributed systems, distributed middleware, ubiquitous computing, and applications such as the Internet of Things, Web and peer-to-peer systems. Prerequisites: Students should be able to easily program in a high-level language such as Java, C++ or Python, have had a course on data structures, be familiar with elements of computer architecture and have had previous exposure to the operating system concepts of processes, virtual memory, and scheduling. A previous course on uniprocessor operating systems (e.g., COMPSCI 377) will be helpful but not required. 3 credits.

COMPSCI 683 Artificial Intelligence

INSTRUCTOR(S): Yair Zick

In-depth introduction to Artificial Intelligence focusing on techniques that allow intelligent systems to reason effectively with uncertain information and cope limited computational resources. Topics include: problem-solving using search, heuristic search techniques, constraint satisfaction, local search, abstraction and hierarchical search, resource-bounded search techniques, principles of knowledge representation and reasoning, logical inference, reasoning under uncertainty, belief networks, decision theoretic reasoning, representing and reasoning about preferences, planning under uncertainty using Markov decision processes, multi-agent systems, and computational models of bounded rationality. 3 credits.

COMPSCI 685 Adv Natural Language Processng

INSTRUCTOR(S): Mohit Iyyer

This course covers a broad range of advanced level topics in natural language processing. It is intended for graduate students in computer science who have familiarity with machine learning fundamentals, and previous course or research experience in natural language processing. It may also be appropriate for computationally sophisticated students in linguistics and related areas. Topics include probabilistic models of language, computationally tractable linguistic representations for syntax and semantics, neural network models for language, and selected topics in discourse and text mining. After completing the course, students should be able to read and evaluate current NLP research papers. Coursework includes a research literature review, homework assignments, and a final project. 3 credits.

COMPSCI 688 Probabilistic Graphical Models

INSTRUCTOR(S): Daniel Sheldon

Probabilistic graphical models are an intuitive visual language for describing the structure of joint probability distributions using graphs. They enable the compact representation and manipulation of exponentially large probability distributions, which allows them to efficiently manage the uncertainty and partial observability that commonly occur in real-world problems. As a result, graphical models have become invaluable tools in a wide range of areas from computer vision and sensor networks to natural language processing and computational biology. The aim of this course is to develop the knowledge and skills necessary to effectively design, implement and apply these models to solve real problems. The course will cover (a) Bayesian and Markov networks and their dynamic and relational extensions; (b) exact and approximate inference methods; (c) estimation of both the parameters and structure of graphical models. Although the course is listed as a seminar, it will be taught as a regular lecture course with programming assignments and exams. Students entering the class should have good programming skills and knowledge of algorithms. Undergraduate-level knowledge of probability and statistics is recommended. 3 credits.

COMPSCI 689 Machine Learning

INSTRUCTOR(S): Brendan O'Connor

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 691NR S-Neural Networks:Neurosci+Eng

INSTRUCTOR(S): Hava Siegelmann

This course covers various aspect of neural networks, from fundamentals to advanced concepts. Topics include feed-forward neural networks, kernel-based approaches, deep learning, recurrent neural networks, Hopfield networks, Kohonen Self-Organized Maps, Grossberg Adaptive Resonance Theory, Helmholtz machines, MDL, Symbolic neural nets, and space-time neurodynamics, with links to computational neuroscience. Theoretical foundations of supervised, unsupervised, and reinforcement learning are described. Advanced machine learning applications include image processing, speech recognition, game playing, time series prediction, and neurocontrol. The course is self-contained, preliminary knowledge of neural networks basics is useful but not required. Students at the 691 level are expected to complete a project implementing a neural network to solve a pattern recognition task, while students at the 591 level will be evaluated based on conceptual designs. COMPARISON BETWEEN COMPSCI 682 AND 591NR/691NR: 682 teaches the engineering techniques necessary to train modern neural network architectures to achieve competitive performance on problems such as image classification and sequence prediction. This includes optimization techniques, hyperparameter setting, and debugging techniques for large neural networks. There is less emphasis on theory. 591NR/691NR introduces a broader array of neural network models beyond feed-forward networks (such as Kohonen nets, Hopfield nets, Boltzmann machines, Adaptive Resonance Theory) and analyzes some of their theoretical properties, such as guaranteed convergence and stability. In general, there is more emphasis on theoretical properties, and less on engineering issues. 3 credits.

COMPSCI 691O S-Tools;Explanatory+TutorSystm

INSTRUCTOR(S): Beverly Woolf

Artificial Intelligence will radically change education. Through machine learning, data mining, analytics, robotics, and user models, AI will replace false learning boundaries (e.g., learning places, time, level of study); personalize learning; make learning instantly available to everyone; connect learners with partners; provide multi-media; and augment human learning ability. This seminar examines recent work in explanatory and tutoring systems, presents theories about digital teaching and learning, and describes how to deliver personalized teaching in online systems. Such software supports people working alone or in collaborative inquiry to rapidly access and integrate global information. This course describes how to build tutors, stimulates awareness of research issues, and promotes sound analytic and design skills. Specific topics include systems that support collaboration, inquiry, natural language dialogue, authoring tools and user models. The course is appropriate for students from many disciplines (e.g., computer science, linguistics, education, and psychology), researchers, and practitioners from academia, industry, and government. No programming is required. Students will read and critique papers about AI tools (e.g., vision, natural language), methods, and will 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 on your chosen topic. Students present readings from the research literature and several working systems will be available for hands-on or virtual critique. 3 credits.

COMPSCI 692L S-Natural Language Processing

INSTRUCTOR(S): Mohit Iyyer

Weekly seminar requiring students to read an NLP paper and discuss and review it from a variety of perspectives. Some weeks will feature invited speakers instead of paper reviews. 1 credit.

COMPSCI 692U S- Quantum Error Correction

INSTRUCTOR(S): Stefan Krastanov, Donald Towsley

The first few weeks of the course will consist of introduction to classical and quantum error correction through lectures by the instructors. After the introduction is completed, we will transition to the seminar part of the course where students will be studying and presenting recent advancements in quantum error correction. The main focus would be on topological codes like the surface and toric codes, and on good quantum codes like the recently developed quantum LDPC codes. 1-3 credits.

COMPSCI 701 Advanced Topics Computer Sci

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 Computer Sci

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 879 TeachngAssist;TomorrowsFaculty

INSTRUCTOR(S): Ivon Arroyo

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. 2 credits.

COMPSCI 891M S-Theory of Computation

INSTRUCTOR(S): Hung Le

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 Intro To Computation (colloq)

INSTRUCTOR(S): David Barrington

This course is an honors colloquium for COMPSCI 250. 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. Prerequisites: Students must be enrolled in or have completed COMPSCI 250. 1 credit.

COMPSCI H311 Intro to Algorithms (colloq)

INSTRUCTOR(S): Marius Minea

The design and analysis of efficient algorithms for important computational problems. Emphasis on the relationships between algorithms and data structures and on measures of algorithmic efficiency. Advanced graph algorithms, dynamic programming applications, NP-completeness and space complexity, approximation and randomized algorithms. Experimental analysis of algorithms also emphasized. Use of computer required. Prerequisites: Students must be enrolled in or have completed COMPSCI 311. 1 credit.

COMPSCI H370 Intro;Computer Vision (colloq)

INSTRUCTOR(S): Subhransu Maji

The colloquium will focus on advanced topics and recent research topics related to computer vision. Students will participate in group discussions and carry out a group or individual project which will be an extension to the project work in COMPSCI 370. 1 credit.

COMPSCI H377 Operating Systems (colloq)

INSTRUCTOR(S): Timothy Richards

The design and operation of modern computer operating systems. Review of capabilities of typical computer hardware. Topics include command language interpreter (the shell), processes, concurrency, inter-process communication, linking and loading, memory management, transactions, file systems, distributed systems, security, and protection. Programming projects in Java and C. Students taking this 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.

COMPSCI H453 Computer Networks (colloq)

INSTRUCTOR(S): James Kurose

Students will meet with instructor in small group setting with the class instructor on a weekly basis to discuss related topics of interest, including but not limited to: Internet privacy, network neutrality, network source code implementation. 1 credit

INFO 101 Introduction to Informatics

INSTRUCTOR(S): Cheryl Swanier

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 Mathmtcl Fndtn for Informatics

INSTRUCTOR(S): Mohammadhassan Hajiesmaili

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.

INFO 203 A Networked World
INSTRUCTOR(S): Parviz Kermani

The course will cover the technical foundations of today's communication networks, particularly the Internet. It will also address key social, policy, economic and legal aspects of these networks, their use (and abuse), and their regulation. This course covers computer science topics, but all material will be presented in a way that is accessible to an educated audience with or without a strong technical background. Not intended for Computer Science majors – students interested in a CS majors-level treatment of this material should see COMPSCI 453. 3 credits.

INFO 248 Intro to Data Science
INSTRUCTOR(S): Gordon Anderson

This course is an introduction to the concepts and skills involved with the collection, management, analysis, and presentation of data sets and the data products that result from the work of data scientists. Privacy, algorithmic bias and ethical issues are also discussed. Students will work with data from the financial, epidemiological, educational, and other domains. The course provides examples of real-world data that students work with using various software tools. This course consists of two lecture meetings and one lab meeting per week. Readings will be assigned as preparation for each class meeting. A semester project will be assigned. Students work in pairs to develop their project over the semester. The project provides students with an opportunity to work collaboratively to explore the topics in more depth in a specialized domain. A midterm and final exam will be given. Grades are determined by a combination of scores on lab activities, projects, and exam scores. Software: all software is freely available. Prerequisites: COMPSCI 121 and PSYCH 240 (or OIM 240, or STAT 240, or STAT 515, or RES ECON 212, or SOCIOL 212). 4 credits.

INFO 490PI Personal Health Informatics
INSTRUCTOR(S): Ravi Karkar

This course will cover the design of personal health and wellness technologies. Using the personal health informatics model, we will learn various challenges in designing technologies for personal health data collection (e.g., step count, heart rate, or food intake etc.), integration, self-reflection, and behavior change. Going further, students will understand design issues in sharing personal health data and discuss design guidelines for collaborative data collection, reflection, and care. It is difficult to create health technologies that can successfully be integrated into people's daily life due to many obstacles in individuals' data collection, integration, self-reflection, and sharing practices. Understanding these challenges is an important part of designing Health Technologies. Therefore, this course will cover HCI and design thinking methods that students can leverage to understand the adoption and use of Health Technologies and to design effective Health Technologies. Moreover, visualizations facilitate people to gain insights from their data, so we will cover common visualization approaches used in the personal data contexts. Students will apply the design issues taught during lecture to a team-based semester-long personal health application design project. This course satisfies the IE requirement for Informatics majors and it also counts as an elective for all concentrations of the Informatic major. Prerequisites: INFO 248 and (COMPSCI 186 or COMPSCI 187). 4 credits.

INFO H490PI PersonalHealthInformatc colloq
INSTRUCTOR(S): Ravi Karkar

This colloquium will dive deeper into issues related to design of personal health technologies. You will study examples of existing platforms, read relevant publications, and will produce a critical reflection on the state and future direction of the field.