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): Meng-Chieh Chiu, Fatemeh Ghaffari, Ghazaleh Parvini, Allison Poh, Cole Reilly, Manan Talwar

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 15 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 127 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? This course is for everyone at all levels and with all interests. No programming or prerequisites are required. We focus on building skills to think analytically, broadly, and strategically, as well as to communicate effectively about complex problems with societal impact. Assignments will provide students multiple paths to success. Counts towards the IT minor. (Gen. Ed. SI) 4 credits.

CICS 160 Object-Oriented Programming

INSTRUCTOR(S): Gordon Anderson

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. (Gen. Ed. R2) Prerequisite: CICS 110 (previously INFO 190S) or COMPSCI 121. 4 credits.

CICS 210 Data Structures

INSTRUCTOR(S): Mordecai Golin, 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. (Gen. Ed. R2) Prerequisite: CICS 160 (previously INFO 190T). 4 credits.

CICS 256 Make: Physical Computing

INSTRUCTOR(S): 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: CICS 210 (or COMPSCI 187) and Basic Math Skills (R1). 4 credits.

CICS 291S S- CICS Second Year Pathways

INSTRUCTOR(S): Emma Anderson

This seminar will give students the opportunity to learn more about the different areas and career pathways in computer science, through guest lectures by professors and grad students, reading research papers, and coordination with CICS Careers. It is intended to be taken alongside any COMPSCI 200-level core course. 1 credit.

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, Elizabeth Gunther, Siobhan Meï, Justin Obara, Samuel Pulford, 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. CICS Primary Majors only. 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 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 198C P-Intro;C Programming Language

INSTRUCTOR(S): Meng-Chieh Chiu, 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). A required prerequisite for COMPSCI 230, effective Fall 2023. Prerequisite: CICS 160 (previously INFO 190T or COMPSCI 186) or COMPSCI 121 with a grade of B or better. 1 credit.

COMPSCI 220 Programming Methodology

INSTRUCTOR(S): Jaime Davila

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: CICS 210 (or 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: CICS 210 (or COMPSCI 187) and COMPSCI 198C. 4 credits.

COMPSCI 240 Reasoning Under Uncertainty

INSTRUCTOR(S): Shiting Lan, Mark Wilson

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 numbers, and central limit theorem. Probabilistic reasoning -- conditional probability and odds, Bayes' Law, Markov Chains, Bayesian Networks. Statistical topics such as estimation of parameters and linear regression, as time permits. Prerequisite: CICS 160 (previously INFO 190T or COMPSCI 187) and MATH 132. 4 credits.

COMPSCI 250 Introduction To Computation

INSTRUCTOR(S): David Barrington, Mordecai Golin

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: CICS 160 (previously INFO 190T or COMPSCI 187 or ECE 241) 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: CICS 210 or COMPSCI 187, and either COMPSCI 250 or MATH 455. 4 credits.

COMPSCI 320 Software Engineering

INSTRUCTOR(S): Yuriy Brun, Heather Conboy, 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 and counts as a CS Elective for the 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 a CS Elective toward the CS Major and as a Required Core for the INFORM Major. Prerequisite: CS Majors: CICS 210 or COMPSCI 187; INFORM Majors: INFO 248 and CICS 160 (previously INFO 190T or 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 and an Elective for both the CS and INFORM Majors. Prerequisite: CS Majors: COMPSCI 220 (or COMPSCI 230); INFORM Majors: INFO 248 and CICS 160 (previously INFO 190T or COMPSCI 186 or COMPSCI 187). 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 345 Pract + Appl of Data Managemnt

INSTRUCTOR(S): Jaime Davila

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. This course counts as a CS Elective toward the CS Major. Students who have completed COMPSCI 445 are not eligible to take this course without instructor permission. Prerequisite: CS Majors: CICS 210 or COMPSCI 187; INFORM Majors: INFO 248 and CICS 160 (previously INFO 190T or COMPSCI 186 or COMPSCI 187). 3 credits.

COMPSCI 348 Principles of Data Science

INSTRUCTOR(S): David Jensen

Data science uses various concepts, practices, algorithms, and systems to extract knowledge and insights from data. It encompasses techniques from machine learning, statistics, databases, visualization, and several other fields. When properly integrated, these techniques can help human analysts make sense of vast stores of digital information. This course presents the fundamental principles of data science, familiarizes students with the technical details of representative algorithms, and connects these concepts to applications in industry, science, and government, including fraud detection, marketing, scientific discovery, and web mining. The course assumes that students are familiar with basic concepts and algorithms from probability and statistics. This course counts as a CS Elective toward the CS Major. Prerequisites: CICS 210 (or COMPSCI 187), COMPSCI 240, and COMPSCI 250 (or MATH 455). 3 credits.

COMPSCI 360 Intro Computr + Ntwrk Security

INSTRUCTOR(S): Shiqing Ma

This course provides an introduction to the principles and practice of computer and network security. A focus on both fundamentals and practical information will be stressed. The three key topics of this course are cryptography, privacy, and network security. Subtopics include ciphers, hashes, key exchange, security services (integrity, availability, confidentiality, etc.), security attacks, vulnerabilities, anonymous communications, and countermeasures. This course counts as a CS Elective for the CS Major. Prerequisite: COMPSCI 230. 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. This course counts as a CS Elective for the CS Major. Prerequisite: COMPSCI 240 or 383. 3 credits.

COMPSCI 373 Intro to Computer Graphics

INSTRUCTOR(S): Rui Wang

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: CICS 210 (or 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 for the CS Major. Prerequisites: COMPSCI 230. 4 credits.

COMPSCI 383 Artificial Intelligence

INSTRUCTOR(S): Matthew Rattigan

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 an Elective toward the CS and INFORM Majors. Prerequisite: COMPSCI 220 (or COMPSCI 230) and COMPSCI 240 (or STATISTC 515). 3 credits.

COMPSCI 389 Intro to Machine Learning

INSTRUCTOR(S): Cooper Sigrist, Philip Thomas

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 STATISTC 515), and MATH 233. 3 credits.

COMPSCI 390B HarnessingDataSci;SocietalGood

INSTRUCTOR(S): Abhidip Bhattacharyya

This is a project-based course in which students will explore using large-scale datasets and data analysis to address real-world societal problems in domains such as sustainability, health, and work with different techniques of data analysis and processing. Students will address problems of societal or industrial relevance. Each semester, the course will offer one or more real-world datasets and a selection of sample problems and students will define a project based on these datasets to address a real-word problem in a group setting. Students will collaborate in groups for their project. Students will explore modern data processing tools and software systems to build data processing pipelines for their chosen project. Throughout the course, students will be expected to present their project ideas, develop project proposals outlining their implementation plans, and conclude with a final presentation and report submission. This course provides students with the opportunity to integrate their analytical and collaborative skills for real-world problem-solving. This course counts as a CS Elective toward the CS Major. 3 credits.

COMPSCI 390R ReverseEngin+VulnrbltyAnalysis

INSTRUCTOR(S): Steven Rossi

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, but does not count as an INFORM Elective. Prerequisites: COMPSCI 230 (or E&C-ENG 322 or E&C-ENG 373) or permission of instructor. 3 credits.

COMPSCI 391M S-Make: Audio Circuit Analysis

INSTRUCTOR(S): Brian Levine, Rui Wang

This course is designed for students who wish to expand on knowledge and experience gained previously from a Maker course by studying solid state electronics that are used for audio signal processing. Specifically, we will examine circuit designs that are behind effects used in contemporary musical compositions. Our focus will be on the theory that explains how electronics circuits work to process audio signals, such as "clip" waveforms (commonly known as distortion, fuzz, or overdrive). Students will gain experience in electronics theory, circuit design, sound processing, and circuit assembly/soldering skills. Does not count as a CS Elective. Prerequisite: CICS 256 or COMPSCI 335. 1 credit.

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 a CS Elective for the CS Major. Prerequisite: COMPSCi 320 (or COMPSCI 326). 3 credits.

COMPSCI 429 Software Engin Proj Management

INSTRUCTOR(S): Yuriy Brun, Heather Conboy, 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. This course counts as a CS Elective for the CS Major. 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 B or better. 3 credits.

COMPSCI 445 Information Systems

INSTRUCTOR(S): Trek Palmer

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): Ali Montazeralghaem

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 a CS Elective for the CS Major. Prerequisite: COMPSCI 240 or COMPSCI 383. 3 credits.

COMPSCI 453 Computer Networks

INSTRUCTOR(S): Arun Venkataramani

This course provides an introduction to fundamental concepts in the design and implementation of computer networks, their protocols, and applications with a particular emphasis on the Internet's TCP/IP protocol suite. Topics to be covered include: overview of network architectures, applications, network programming interfaces (e.g., sockets), transport, congestion, routing, and data link protocols, addressing, local area networks, wireless networks, network security, and network management. There will be five or six homeworks, two programming projects, several hands-on labs (that require an Internet-connected personal computer) and two exams. This course counts as a CS Elective for the CS Major. Prerequisite: 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 485 Applications of NLP

INSTRUCTOR(S): Brendan O'Connor

This course will introduce NLP methods and applications, such as text classification, sentiment analysis, machine translation, and other applications to identify and use the meaning of text. During the course, students will (1) learn fundamental methods and algorithms for NLP; (2) become familiar with key 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 use, implement, experiment with, and improve NLP tools. This course counts as a CS Elective for the CS Major. Prerequisite: COMPSCI 220 and COMPSCI 240, or LINGUIST 429B (previously LINGUIST 492B). 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. This course counts as a CS Elective for the CS Major. Prerequisites: MATH 132, MATH 235, and either COMPSCI 240 or STATISTC 515. 3 credits.

COMPSCI 491G S-Computer Networking Lab

INSTRUCTOR(S): Parviz Kermani

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 520 Thry + Practice;Software Engin

INSTRUCTOR(S): Juan Zhai

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 a CS Elective for the CS Major. Undergraduate Prerequisites: COMPSCI 320 (or COMPSCI 220 and COMPSCI 326). 3 credits.

COMPSCI 528 Mobile and Ubiquit. Computing

INSTRUCTOR(S): Phuc Nguyen

This course will introduce students to the field of mobile sensing and ubiquitous computing (Ubicomp) – an emerging CS research area that aims to design and develop disruptive technologies with hardware and software systems for real-world messy, noisy and mobile scenarios. The students will learn how to build mobile sensing systems, how to implement it with ubiquitous computing tools, how to make sense of the sensor data and model the target variables. Lastly, the students will learn how to critically think about problems in many application areas including Human-Computer Interaction, Medicine, Sustainability, Transportation, Psychology and Economics, and subsequently practice to find appropriate Ubicomp solutions. There is no exam in this course. The student is expected to work on different hands-on assignments, critique writing, and a final project. This course counts as an Elective toward the CS Major. Undergraduate Prerequisites: COMPSCI 230 and COMPSCI 240. 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 a CS Elective for 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. Semester team project to design an architecture and develop a software simulation of it. This course counts as a CS Elective for the CS Major. Undergraduate Prerequisites: COMPSCI 335. 3 credits.

COMPSCI 546 Applied Information Retrieval

INSTRUCTOR(S): Hamed Zamani

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 (or COMPSCI 326) and either COMPSCI 383, COMPSCI 446, COMPSCI 485, 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 Prerequisites: CICS 210 (or COMPSCI 187) and STATISTC 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. Previously INFOSEC 690S. This course counts as an Elective toward the CS Major. Undergraduate Prerequisites: COMPSCI 360 (previously COMPSCI 460) or COMPSCI 560/597N or COMPSCI 660) and COMPSCI 453. 3 credits.

COMPSCI 564 Cyber Effects

INSTRUCTOR(S): Jeffrey Hamalainen, Seth Landsman, Nick Merlino, Edward Walters, Adam Woodbury

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 Prerequisites: COMPSCI 230 (or E&C-ENG 322) and COMPSCI 360 (previously COMPSCI 460 or COMPSCI 365, or COMPSCI 390R, or COMPSCI 466, or E&C-ENG 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. Undergraduate Prerequisites: B or better in COMPSCI 311, COMPSCI 383, and 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 a CS Elective for the CS Major. Undergraduate Prerequisites: MATH 545 and COMPSCI 240 and STATISTC 515 C or better. MATH 545 can be skipped by students who have taken MATH 235 and MATH 233 both with B+ or better. STATISTC 515 can be skipped by students who have taken COMPSCI 240 with a B+ or better. 3 credits.

COMPSCI 590AB Quantum Cryptography and Comm

INSTRUCTOR(S): Filip Rozpedek

The ability to transmit quantum information over long distances will enable implementation of many fascinating quantum communication tasks and provide us with novel capabilities that reach beyond what we can do over classical Internet alone. Examples of such tasks include blind quantum computing, clock synchronization or distributed quantum computing. Quantum cryptography is one family of such tasks with the most famous one being quantum key distribution. This task, which is currently the most mature quantum technology, enables distribution of shared keys through a protocol that is information-theoretically secure and whose security remarkably is guaranteed by the laws of quantum physics. Such unconditional security cannot be achieved in the classical world. In the first part, the course will introduce the world of quantum cryptographic protocols and describe how the power of quantum mechanics can enable distribution of shared secret keys even with untrusted devices. It will also introduce many other fascinating quantum protocols beyond quantum key distribution. In the second part we will learn about the uniquely quantum challenges of transmitting quantum information over long distances. We will then study how to overcome them using different types of the so-called "quantum repeaters". We will finish by investigating the fundamental limits of quantum communication over practical noisy channels and we will use this framework for assessing quantum repeater performance. This course counts as a CS Elective for the CS Major. Undergraduate Prerequisites: MATH 132 AND MATH 235 AND COMPSCI 240 (or STATISTC 515 or PHYSICS 281 or PHYSICS 287). 3 credits

COMPSCI 590AE Mobile and Wireless Networks

INSTRUCTOR(S): James Kurose

This course covers wireless networking and mobility principles and practice. The focus of "practice" is primarily 802.11 (WiFi) and 4G/5G cellular networks. Unlike many other wireless networking courses, this course focuses primarily on the networking and systems aspects of wireless and mobile networks, rather than on the wireless channel aspects. This course counts as a CS Elective for the CS Major. Undergraduate Prerequisites: COMPSCI 453 or ECE 374. 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): Andrew McGregor

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

COMPSCI 614 RandomizedAlgorithms;App to DS

INSTRUCTOR(S): Cameron Musco

Randomness has proven itself to be a useful resource for developing provably efficient algorithms and protocols for large scale data processing. As a result, the study of randomized algorithms has become a major research topic in recent years. This course will explore a collection of techniques for effectively using randomization and for analyzing randomized algorithms, as well as examples from a variety of settings and problem areas. The course is a natural follow on both COMPSCI 514: Algorithms for Data Science and COMPSCI 611: Advanced Algorithms. 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 645 DatabaseDesign+Implementation

INSTRUCTOR(S): Alexandra Meliou

This course covers the design and implementation of traditional relational database systems as well as advanced data management systems. The course will treat fundamental principles of databases such as the relational model, conceptual design, and schema refinement. We will also cover core database implementation issues including storage and indexing, query processing and optimization, and transaction management. Additionally, we will address challenges in modern networked information systems, including data mining, provenance, data stream management, and probabilistic databases. 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 670 Computer Vision

INSTRUCTOR(S): Grant Van Horn

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 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. 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 690AB Systems for Deep Learning

INSTRUCTOR(S): Hui Guan

This course is designed to provide a comprehensive understanding of computer systems architecture that supports deep learning workloads. It assumes students have prior knowledge on computer systems, algorithms, and Python/C/C++ programming background. In the course, we will study the full-stack system design to support deep learning, covering topics from the high-level programming frameworks to low-level kernel implementations. We will also introduce cutting-edge research on efficient and scalable deep learning model training, inference, and serving. 3 credits.

COMPSCI 690G Security;Large-Scale Systems

INSTRUCTOR(S): Pubali Datta

This course provides an in-depth examination of the issues in system security, and assumes prior knowledge of operating system concepts. We will start with learning the classic approaches to security attacks and defenses from the perspective of a single host system. Topics include access control, information flow control, system auditing, privilege separation, security policies, host-based intrusion detection etc. Then we will study how these classic mechanisms are expanded and adapted to modern distributed system platforms such as cloud platforms, Internet of Things platforms, and large enterprises. 3 credits.

COMPSCI 690L Deep Generative Models

INSTRUCTOR(S): Sajjad Amini

This course offers an introduction to the probabilistic foundations and learning algorithms of generative models, with a focus on deep learning architectures. We will delve into generative models as conditional probability distributions represented by p(x|y), where 'x' is a high-dimensional random vector, and 'y' can be either high or low-dimensional. The curriculum encompasses various facets of generative models, including sampling, density estimation, training techniques, the exploration of latent spaces, and architectural considerations. We'll also examine the practical applications of these models in tasks such as data generation, imputation, and latent space interpolation. Accompanying the course are hands-on projects and exercises that allow students to investigate the scalability of various methods in real-world scenarios, as well as their theoretical underpinnings. 3 credits.

COMPSCI 690R Computing;Human Mvmt Analysis

INSTRUCTOR(S): Sunghoon Lee

Computer science has played a pivotal role in developing innovative technologies to monitor patients' behaviors and behavioral phenotypes beyond the traditional laboratory or clinical environments. A profound understanding of patient behaviors has the potential to unlock a myriad of applications, including the implementation of targeted behavioral interventions to drive substantial improvements in health-related outcomes, monitoring the progress of individuals undergoing rehabilitation, and assessing the efficacy of emerging therapeutic interventions. In this course, we will delve into the application of machine learning and mobile technologies in the analysis of human movement and behavior. More specifically, the course curriculum will encompass the fundamentals of human movement analysis, mobile and wearable sensing technologies to support remote human movement monitoring, fundamentals of signal processing techniques, and state-of-the-art machine learning techniques, all with the overarching goal of improving our understanding of human behaviors and behavioral phenotypes. In addition, we will cover essential topics such as human subject study design (e.g., randomized controlled trials) and hypothesis testing (e.g., t-test, ANOVA, correlation test, etc.). These skills are crucial for their application within the realm of clinical sciences, especially when it comes to assessing the effectiveness of emerging healthcare and wellness technologies. This course contains lectures, assignments, a final project, paper presentations, and critical discussions. 3 credits.

COMPSCI 690S Human-Centric Machine Learning

INSTRUCTOR(S): Scott Niekum

This course will focus on modern machine learning approaches to learn from human demonstrations, preferences, feedback, and other multimodal signals, with the goal of aligning agent goals and behaviors with human values and desires. For the purposes of both safety and practicality, it is increasingly important for AI systems to be well-aligned with human users as their capabilities improve and they are deployed more frequently in real-world settings. This course will provide the basic tools to address these important issues, covering topics such as behavioral cloning, inverse reinforcement learning, preference elicitation, active learning, learning from feedback, value alignment, bounded rationality, and best practices for human studies. We will examine applications including robotics, large language models, and self-driving cars. 3 credits.

COMPSCI 690U Computational Bio + Bioinform

INSTRUCTOR(S): Anna Green

This course is designed to provide computer scientists with a comprehensive introduction to the field of computational biology. The course will cover the application of computational techniques to modern research challenges in biology, discussing both foundational algorithms and newly introduced methods. The necessary background on biology will be provided in order to contextualize the methods. The primary focus will be analysis of genomic data, including DNA read assembly algorithms, genome annotation, sequence search, sequence alignment, phylogeny construction, mutation effect prediction, population genetics, and genotype-phenotype association studies. We will also cover gene expression analysis (RNA-seq and single-cell RNA-seq) and protein structure analysis and prediction. Throughout the course, we will emphasize the unique challenges to working with biological data. Through lectures and hands-on programming problem sets, students will develop the necessary skills to tackle computational challenges in the field of biology. 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 692Z S-Topics in Quantum Networks

INSTRUCTOR(S): Kenneth Goodenough, Filip Rozpedek, Donald Towsley

This seminar will cover topics in quantum networks. This includes topics in quantum network architectures including one-way vs. two-way architectures and terrestrial vs. satellite networks; quantum repeater and switch design; techniques for overcoming noise in a network including entanglement distillation and quantum error correction; resource allocation including routing and path selection, and scheduling. Other topics include distributed quantum computing; performance modeling and analysis of quantum networks with an emphasis on capacity and on the quality of quantum states delivered to end users; quantum network measurement including quantum network tomography. All students taking the seminar for credit will be expected to present papers on these topics and to participate in classroom discussion. Students taking the seminar for 3 credits will be expected to propose and complete a project. The two objectives of this seminar are to bring everyone, including instructors, up to speed regarding the state of the art in quantum networks. The second objective is to identify MS and PhD level research problems in quantum networks. 1 or 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, 701 followed by 701Y.
The 6 credit option is for a project that will be completed over two semesters with enrollment in only one semester.

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 791U S-Adv Topics in Info Retrieval

INSTRUCTOR(S): James Allan, Hamed Zamani

A seminar in which students will read, present, and discuss research papers on recent and advanced topics in Information Retrieval (IR). Students are expected to read up to two papers per week. For one or more sessions in the semester, students are expected to make summary presentations and lead discussion of the papers. Students should have taken COMPSCI 646, Information Retrieval, or a comparable course. This semester, the seminar will primarily cover the following main topics: Retrieval-Enhanced Machine Learning, Large Language Models for IR, Generative Information Retrieval, Multi-Modal Information Retrieval, and Information Retrieval for the Common Good. Along with studying research papers on the aforementioned topics, students registered for three credit hours should work with the instructor to complete a mutually agreed-upon research project. This seminar assumes prior knowledge of fundamental information retrieval concepts. 1 or 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 two 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): Andrew McGregor

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 seminar 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. Prerequisite: 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. Prerequisite: Students must be enrolled in or have completed COMPSCI 311. 1 credit.

COMPSCI H389 Intro;Machine Learning colloq

INSTRUCTOR(S): Philip Thomas

This colloquium will dive deeper into issues related to the safety and fairness of machine learning algorithms. You will study examples of misbehaving machine learning systems, and machine learning algorithms designed to avoid these undesirable behaviors. The colloquium will culminate with your training a machine learning model that incorporates high-confidence safety and/or fairness guarantees. Prerequisite: Students must be enrolled in or have completed COMPSCI 389. 1 credit.

COMPSCI H453 Computer Networks (colloq)

INSTRUCTOR(S): Parviz Kermani

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

COMPSCI H589 Machine Learning (colloq)

INSTRUCTOR(S): Benjamin Marlin

This colloquium will enrich the primary course by focusing on reading, presenting, and discussing foundational and recent research papers from the machine learning literature. Students will write weekly reading responses, and lead one to two group discussions over the course of the semester. Prerequisite: Students must be enrolled in or have completed COMPSCI 589. 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): Mark Wilson

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): Mohammadhassan Hajiesmaili

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: CICS 110 (or CICS 160 or COMPSCI 119 or COMPSCI 121) and either PSYCH 240, OIM 240, STATISTIC 240, RES-ECON 212, SOCIOL 212, or STATISTC 515. 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 Informatics major. Prerequisites: INFO 248 (or COMPSCI 240) and CICS 210 (or COMPSCI 186 or COMPSCI 187). 4 credits.