CICS 110 Foundations of Programming

INSTRUCTOR(S): Victor Chen, Evan Ciccarelli, Jaime Davila, Cole Reilly

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): Emily Nutwell

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 and is required for the PIT Undergraduate Certificate. (Gen. Ed. SI) 4 credits.

CICS 160 Object-Oriented Programming

INSTRUCTOR(S): Cole Reilly, Ella Tuson

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 with a grade of C or better. 4 credits.

CICS 190A Makerspace Studio: Phys Comp

INSTRUCTOR(S): Michael Pfeiffer

This course explores tools, techniques, and methods of modern making in the context of physical computing with a focus on digital fabrication and electronics. It will introduce concepts in 2D and 3D design, basic shop techniques, 3D printing, subtractive manufacturing with laser engraving and CNC milling, elementary microcontroller programming, basic electronic sensors, motors, and LED lighting. Students will learn a framework for working with tools that support hardware projects in computer science. Classes will meet once a week in the Physical Computing Makerspace for a hands-on experience, giving students a chance to work with tools and materials during each meeting. Materials will be provided. No experience necessary. 1 credit.

CICS 208 DefendingDemocracyDigitalWorld

INSTRUCTOR(S): Ethan Zuckerman

This course explores the significance of the public sphere - from pamphlets, newspapers and letters to radio, television, the internet and social media - and its relationship to participatory, democratic society. Moving back and forth between the history of the public sphere and contemporary debates about the tensions between media and democracy, students will learn why democracies prescribe protected roles of the media, how media manipulation plays a role in politics, and how media spaces serve as deliberative spaces. Students will write short reaction papers to the readings, which will be used to shape class discussions, and a longer final paper, focused on applying the theories of the public sphere to regulation of contemporary online spaces. This course does not count toward CS or INFORM Major requirements. Cross-listed with COMM/SPP 208. (Gen. Ed. SB) 3 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) with a grade of C or better. 4 credits.

CICS 237 Intro/ResearchintheDiscipline

INSTRUCTOR(S): Neena Thota

The ‘Introduction to Research in the Discipline’ course is part of the CICS Early Research Scholars Program (ERSP). It provides a group-based, dual-mentored research structure designed to provide a supportive and inclusive first research experience for a large number of early-career Computer Science and Informatics majors. 2 credits.

CICS 256 Make: Physical Computing

INSTRUCTOR(S): Phuc Nguyen

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) with a grade of C or better and completion of the R1 (Basic Math Skills) Gen. Ed. 4 credits.

CICS 291C S-Find Strengths/Design Career

INSTRUCTOR(S): Casey Maloney

This course is designed to prepare CICS students for their internship and job searches, improve their professional skills (both technical and soft) and help them approach professional development and/or advanced educational opportunities with confidence. 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, Siobhan Meï, Justin Obara, Christina Sutcliffe

This class satisfies the Junior Year Writing requirement by providing instruction in several different technical communication genres, including research writing and science writing. Writing contexts will include writing for municipal, public, and professional audiences. Through our writing, we will engage with computing topics and challenges facing our current global context and explore various impacts of computers on modern society. 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 with a grade of C or better (or English Writing waiver), COMPSCI 220, COMPSCI 230 and COMPSCI 240 (or 250); INFORM Majors: ENGLWRIT 112 with a grade of C or better (or English Writing waiver) and INFO 248. 3 credits.

COMPSCI 119 Intro to Programming

INSTRUCTOR(S): Bao Nguyen

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

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 CICS 210) with a grade of C or better, or COMPSCI 121 with a grade of B or better. 1 credit.

COMPSCI 220 Programming Methodology

INSTRUCTOR(S): Marius Minea, James Perretta

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) with a grade of C or better. 4 credits.

COMPSCI 230 Computer Systems Principles

INSTRUCTOR(S): Phuthipong Bovornkeeratiroj, 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) with a grade of C or better and COMPSCI 198C. 4 credits.

COMPSCI 240 Reasoning Under Uncertainty

INSTRUCTOR(S): Ghazaleh Parvini, Gayane Vardoyan

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) or CICS 210 and MATH 132, all with a grade of C or better. 4 credits.

COMPSCI 250 Introduction To Computation

INSTRUCTOR(S): Mordecai Golin, Mingda Qiao

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 E&C-ENG 241, or CICS 210) and MATH 132, all with a grade of C or better. 4 credits.

COMPSCI 311 Introduction to Algorithms

INSTRUCTOR(S): Ghazaleh Parvini, Daniel Sheldon, Hava Siegelmann

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 (You will learn about NP-Completeness). There is no programming assignment but you should know a programming language to understand the algorithms . Mathematical experience (as provided by COMPSCI 250) is required. This course is required for the CS Major (BS) and counts as a CS Elective for the CS Major (BA). Prerequisite: CICS 210 (OR COMPSCI 187) , and either COMPSCI 250 or MATH 455, all with a grade of C or better. 4 credits

COMPSCI 320 Software Engineering

INSTRUCTOR(S): Gordon Anderson, Madeline Endres, James Perretta

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 with a grade of C or better. 4 credits.

COMPSCI 325 Intro/Human-Comp Interaction

INSTRUCTOR(S): Ella Tuson

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. Open to juniors and seniors in Computer Science or Informatics. Prerequisite: Prerequisites: COMPSCI 187 (or CICS 210) with a grade of C or better OR INFO 248 and COMPSCI 186 (or 187 or CICS 160;INFO 190T) with a grade of C or better. 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. Note: as the name ‘web programming’ denotes, programming is a key component of this class. Previous background in JavaScript is strongly recommended. Prerequisite: CS Majors: COMPSCI 220 with a grade of C or better; INFORM Majors: INFO 248 and CICS 210, both with a grade of C or better. 4 credits.

COMPSCI 328 Mobile Health Sensing&Analytcs

INSTRUCTOR(S): Deepak Ganesan

Modern smartphones and wearables together form a powerful sensing platform that capture movement, speech, physiological signals (heart rate, respiratory rate, oxygen saturation, and more), and behavioral patterns continuously and unobtrusively in daily life. This convergence, referred to as mobile health sensing, promises to revolutionize our understanding of human health and behavior. This course is a hands-on introduction to personal health sensing through mobile phones, with an emphasis on conceptual depth over implementation mechanics. Students will develop a working understanding of signal processing, sensor physics, and machine learning as they apply to real health and behavioral sensing problems. All work is done in teams and centers on developing complete, end-to-end sensing applications: structured projects guide teams through problems such as activity recognition, physiological monitoring, and audio-based health screening, all grounded in data collected directly from smartphones. The course culminates in a self-conceived team project where teams are free to define their own sensing problem, formulate their approach, and build toward a solution of their choosing. Programming is in Python, and fluency with modern development tools and workflows including AI coding assistants is expected. Prerequisites: CS MAJORS: CICS 210 (or COMPSCI 187) with a grade of C or better; INFORM MAJORS: INFO 248 and CICS 160 (previously INFO 190T or COMPSCI 186 or COMPSCI 187), all with a grade of C or better. 3 credits.

COMPSCI 335 Inside the Box:How Cmps Work

INSTRUCTOR(S): Charles Weems

How does the computer actually work? In this course we peel away the layers of abstraction and look at how switches become logic circuits, how logic circuits do math, and how programs really execute. We will wire up some simple examples of logic, then move on to programming an embedded ARM processor in a mix of assembly language and C, interfacing with various I/O devices and sensors, to experience what happens when machine code executes. We will also see the impact of hidden acceleration mechanisms like caches, pipelines, and branch predictors. This course counts as a CS Elective for the CS Major. Open to senior and junior Computer Science majors only. Prerequisite: Prerequisite: COMPSCI 220 or 230 with a grade of 'C' or better. 3 credits.

COMPSCI 345 Pract & Appl of Data Managemnt

INSTRUCTOR(S): Gordon Anderson, Neha Makhija, 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. The emphasis is on relational databases, though non-relational databases are also introduced. Some of the covered topics include the relational data model, data retrieval, application-driven database design, schema refinement, implementation of basic transactions, database security. This course counts as a CS Elective for 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) with a grade of C or better. INFORM MAJORS: INFO 248 and CICS 160 (previously INFO 190T or COMPSCI
186 or COMPSCI 187), all with a grade of C or better. 3 credits.

COMPSCI 360 Intro Computr & Ntwrk Security

INSTRUCTOR(S): Pubali Datta

This course provides an introduction to the principles and practice of computer, network, and artificial intelligence security. We will focus on both fundamentals and practical information. Two key topics of this course are communication privacy and security on the web, mobile, and AI. Subtopics include ciphers, hashes, key exchange, security services (integrity, availability, confidentiality, etc.), mobile and web security, prompt injections and jailbreaking, and countermeasures. This course counts as a CS Elective for the CS Major. Prerequisite: COMPSCI 230 with a grade of C or better. 3 credits.

COMPSCI 363 Computer Crime Law

INSTRUCTOR(S): Marvin Cable

A study, analysis, and discussion of the legal issues related to crimes involving computers and networks, including topical actions by dissidents and governments. We will also study the technologies of forensic investigation, intelligence gathering, privacy enhancement, and censorship resistance. Our main legal topics will include recent and important case law, statutes, and constitutional clauses concerning authorization, access, search and seizure, wiretaps, the right to privacy, and FISA. Our technology topics will include methods of investigation and resistance in the context of the Internet and Cellular networks. Students are assumed to have no background in legal concepts. Students will be required to complete substantial legal readings, complete significant written analysis of rulings, learn about technologies in detail, and participate in lively class discussion. This course counts as a CS Elective for the CS Major.Open to senior and junior Computer Science majors only. Prerequisite: Prerequisite: COMPSCI 230 with a grade of 'C' or better AND either ENGLWRIT 112 (with a grade of 'C' or better) or the completion of the 'CW' General Education requirement. 3 credits.

COMPSCI 367 ReverseEngin&ExploitDevelopmnt

INSTRUCTOR(S): Lurene Grenier

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 a CS Elective for the CS Major, but does not count as an INFORM Elective. Prerequisite: COMPSCI 230 (or E&C-ENG 322) with a grade of C or better, or permission of instructor. 3 credits.

COMPSCI 377 Operating Systems

INSTRUCTOR(S): Meng-Chieh Chiu

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. Programming projects in C. This course counts as a CS Elective for the CS Major. Prerequisite: COMPSCI 230 with a grade of C or better. 4 credits.

COMPSCI 383 Artificial Intelligence

INSTRUCTOR(S): Matthew Rattigan

This course aims to give students a high level understanding of the prominent AI topics that are being employed in industry today. It will provide an introduction to each topic, an overview of its supporting algorithms, and examples of products powered by the technology. Particular emphasis will be had on Machine Learning and developing hands-on practical skills with this technology. Upon completion of this course, students will obtain a wider scope of understanding about modern AI trends in software technology and develop an intuition for how this software works. To succeed in this course, students will need a fundamental understanding of data structures and programming fundamentals. Graph and tree data structures will be used in particular. Programming assignments in this class will be done using Python. Experience in at least one programming language is required and it's strongly recommended you have some Python experience before starting. A mathematical foundation in statistics and linear algebra is not strictly necessary but will deepen understanding of course material. This course counts as an Elective for CS and INFORM Majors. Prerequisite: CS MAJORS: (CICS 210 or COMPSCI 187) and COMPSCI 240 (or STATISTC 315), both with a grade of C or better; INFORM MAJORS: INFO 348 and STATISTC 315, both with a grade of C or better. 3 credits.

COMPSCI 389 Intro to Machine Learning

INSTRUCTOR(S): Aline Weber

The course provides an introduction to machine learning algorithms and applications, and is intended for students with no prior experience with machine learning. Students with prior experience in machine learning or who are already passionate about the subject are encouraged to take COMPSCI 589 instead. 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 for the CS and INFORM Majors. Prerequisite: a grade of C or better in CICS 210 and COMPSCI 240 (or STATISTC 315 or STATISTC 240 or PSYCH 240 or OIM 240 or RES-ECON 212 or SOCIOL 212), and MATH 132. 3 credits.

COMPSCI 420 Software Entrepreneurship

INSTRUCTOR(S): Matthew Rattigan

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 Elective for the CS and INFORM Majors. Prerequisite: COMPSCi 320 (or COMPSCI 326) with a grade of C or better. 3 credits.

COMPSCI 429 Software Engin Proj Management

INSTRUCTOR(S): Gordon Anderson, Madeline Endres, James Perretta

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): 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. Prerequisite: COMPSCI 220 (or 230) and COMPSCI 311 and COMPSCI 345 with a grade of C or better. 3 credits.

COMPSCI 453 Computer Networks

INSTRUCTOR(S): Phuthipong Bovornkeeratiroj

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) with a grade of C or better. 3 credits.

COMPSCI 461 Secure Distributed Systems

INSTRUCTOR(S): Gregory Stone

This is a class devoted to the study of securing distributed systems, with blockchain-based cryptocurrencies serving as our real platform of interest. We'll start with the fundamentals of Lamport's, Fischer's, and Douceur's results that fence-in all consensus system, and discuss Byzantine fault tolerance. We'll also look at the efficiency of the network architectures for peer-to-peer/distributed system communication and attacks on their security, such as denial of service attacks. And we'll review relevant applied cryptography such as elliptic curves. We’ll discuss in detail the mechanisms of Bitcoin and Ethereum and we’ll program distributed applications for Ethereum. Other topics include economics and finance. Assignments will include programming projects and reading research papers. The grade is also based on exams and participation in discussion. The course is based on a flipped classroom and uses a hybrid instruction model. Some of the course content is delivered online, however students are required to attend weekly class meetings. This course counts as a CS Elective for the CS Major, as well an Any 2 menu choice for the former Security & Privacy track. Open to Computer Science majors only. Prerequisite: COMPSCI 326, COMPSCI 345, COMPSCI 377, COMPSCI 453, or COMPSCI 497P with a grade of C or better. 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 with a grade of C or better, or LINGUIST 429B (previously LINGUIST 492B) with a grade of C or better. 3 credits.

COMPSCI 491P S-Philosoph/AI & Consciousness

INSTRUCTOR(S): Charles Weems

This seminar will explore the philosophies that have contributed to our belief that Turing equivalent computers can achieve human-like intelligence and contrasting philosophies of consciousness, which challenge that belief. We will follow parallel tracks of reading along two branches of philosophy, dualism and monism, to gain a comparative understanding of each viewpoint. On the dualist side we will start from Descartes and Kant, working forward to 20th century mechanistic models of intelligence and some more recent work that reexamines this approach. On the monist side we will primarily rely on a text by the early 20th century Austrian philosopher, Rudolf Steiner, which surveys dualism and monist and proposes a new form of monism. Students will be expected to read roughly one paper and one book chapter per week, writing summaries of the reading and preparing discussion questions. This will involve four to eight hours of work outside of class per week. For the dualist track, each week, one or two students will prepare a paper more extensively and lead the discussion. Grading is thus based on in-person attendance, reading summaries and questions, and leading one discussion. There are no exams. The book and papers will be provided. Counts as a CS Elective for CS Majors. Prerequisites: COMPSCI 250 and COMPSCI 383 (or 389) with a grade of C or better in each. 3 credits.

COMPSCI 514 Algorithms for Data Science

INSTRUCTOR(S): Cameron Musco

With the advent of social networks, ubiquitous sensors, and large-scale computational science, data scientists must deal with data that is massive in size, arrives at blinding speeds, and often must be processed within interactive or quasi-interactive time frames. This course studies the mathematical foundations of big data processing, developing algorithms and learning how to analyze them. We explore methods for sampling, sketching, and distributed processing of large scale databases, graphs, and data streams for purposes of scalable statistical description, querying, pattern mining, and learning. This course counts as a CS Elective for the CS Major. Undergraduate Prerequisite: COMPSCI 240 (or STATISTCS 315/515) and COMPSCI 311 both with a grade of B+ or better, or (COMPSCI 240 and STATISTCS 315/515 and COMPSCI 311 and MATH 233 and MATH 235, all with a C or better). 3 credits

COMPSCI 515 Computational Social Choice

INSTRUCTOR(S): Yair Zick

Recent years have seen a dramatic rise in the use of algorithms for solving problems involving strategic decision makers. Deployed algorithms now assist in a variety of economic interactions: assigning medical residents to schools, allocating students to courses, allocating security resources in airports, allocating computational resources and dividing rent. We will explore foundational topics at the intersection of economics and computation, starting with the foundations of game theory: Nash equilibria, the theory of cooperative games, before proceeding to covering more advanced topics: matching algorithms, allocation of indivisible goods, and mechanism design. Open to junior and senoir Computer Science students. This course counts as a CS Elective for the CS Major. Undergraduate Prerequisite: COMPSI 240 and 250 with a grade of C or better in both. 3 credits.

COMPSCI 520 Thry & Practice/Software Engin

INSTRUCTOR(S): Yuriy Brun

Introduces students to the principal activities and state-of-the-art techniques involved in developing high-quality software systems. Topics include: 1. AI for software engineering including topics like automated testing, program repair and software verification; 2. Software engineering for AI like detecting and fixing fairness bugs in AI models; 3. How industry works. This course counts as a CS Elective for the CS Major. Undergraduate Prerequisite: COMPSCI 320 (or COMPSCI 220 and COMPSCI 326) with a grade of C or better. 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 Prerequisite: COMPSCI 377 and COMPSCI 445 both with a grade of C or better. 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 315 (previously STATISTC 515). 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 a CS Elective for the CS Major. Undergraduate Prerequisite: COMPSCI 365 or COMPSCI 377 with a grade of C or better. 3 credits.

COMPSCI 571 DataVisualization&Exploration

INSTRUCTOR(S): Narges Mahyar

In this course, students will learn the fundamental algorithmic and design principles of visualizing and exploring complex data. The course will cover multiple aspects of data presentation including human perception and design theory; algorithms for exploring patterns in data such as clustering, and dimensionality reduction. A wide range of statistical graphics and information visualization techniques will be covered. We will explore numerical data, relational data, temporal data, spatial data, graphs and text. Hands-on projects will be based on Python or JavaScript with D3. This course counts as an Elective toward the CS and INFORM Majors. Undergraduate Prerequisite: COMPSCI 220, or COMPSCI 230, or COMPSCI 326 with a grade of C or better. No prior knowledge of data visualization or exploration is assumed. 3 credits.

COMPSCI 575 Combntrcs&Graph Thry

INSTRUCTOR(S): Marius Minea

This course is a basic introduction to combinatorics and graph theory for advanced undergraduates in computer science, mathematics, engineering and science. Topics covered include: elements of graph theory; Euler and Hamiltonian circuits; graph coloring; matching; basic counting methods; generating functions; recurrences; inclusion-exclusion; and Polya's theory of counting. This course counts as an Elective toward the CS Major. Open to juniors and seniors. Undergraduate Prerequisites: either COMPSCI 250 or MATH 455 with a grade of B or better. Modern Algebra - MATH 411 - is helpful but not required. 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. The course will assume experience with the Python programming language. While this course has an applied focus, it still requires strong mathematical background in probability and statistics, calculus, and linear algebra. Graduate students should check the descriptions of the undergraduate prerequisite courses to verify that they have sufficient mathematical background for 589. 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 Prerequisite: MATH 545 and COMPSCI 240 and STATISTC 315/515 all with a grade of C or better. (MATH 545 can be skipped by students who have taken MATH 235 and MATH 233 both with B+ or better. STATISTC 315/515 can be skipped by students who have taken COMPSCI 240 with a B+ or better). 3 credits.

COMPSCI 590AF Reverse Engin & Exploit Dev

INSTRUCTOR(S): Lurene Grenier

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 malicious adversaries in 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 counts as a CS Elective for the CS Major. Undergraduate Prerequisite: COMPSCI 230 (or E&C-ENG 322) with a grade of C or better. 3 credits.

COMPSCI 590OP Applied Numerical Optimization

INSTRUCTOR(S): Shiting Lan

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

COMPSCI 590Q Quantum Information Systems

INSTRUCTOR(S): Donald Towsley

To become quantum literate, to understand mathematical representations of quantum computing systems and algorithms; to understand key quantum computing concepts including entanglement and teleportation; to learn “classical” quantum algorithms including Grover’s search and Shor’s factorization algorithms among others; to understand quantum information. This course counts as a CS Elective for the CS Major (BA or BS). Undergraduate Prerequisite: MATH 235 with a grade of C or better. 3 credits.

COMPSCI 590RM Research/Empirical CompSci

INSTRUCTOR(S): David Jensen

This course introduces concepts, practices, and tools for conducting effective research. You will learn how to read technical papers, interpret published research, assess the research frontier, select research topics, devise research questions and hypotheses, propose and plan research activities, analyze experimental results, and report those results. The course is structured around four activities: (1) Synchronous and asynchronous lectures on basic research strategies and techniques; (2) Synchronous activities that apply course concepts; (3) Reading and discussions of technical papers in computer science; and (4) An individual semester-long project that replicates an existing published study in computer science. The course requires significant reading, reviewing, and writing. Students are expected to participate actively in class activities. The course is intended for MS and advanced undergraduate students. It provides a grounding in research methods that will aid your entry into research-oriented industrial positions and PhD studies. For undergraduates considering graduate studies, this course will help inform and accelerate that direction. For undergraduates, this course can be used to satisfy the 499Y requirement for Departmental and Multidisciplinary Honors students whose theses or projects have a substantial empirical component. Undergraduates must obtain approval of the Computer Science Honors Program Director prior to registering. Does not count as a CS elective for the CS major (BA or BS).   3 credits.

COMPSCI 602 Research Methods/Empirical CS

INSTRUCTOR(S): David Jensen

This course introduces concepts, practices, and tools for conducting effective research. You will learn how to read technical papers, interpret published research, assess the research frontier, select research topics, devise research questions and hypotheses, propose and plan research activities, analyze experimental results, and report those results. The course is structured around five activities: (1) Synchronous and asynchronous lectures on basic research strategies and techniques; (2) Synchronous activities that apply course concepts; (3) Reading and discussions of technical papers in computer science; (4) An individual semester-long empirical research project; and (5) Review and feedback on other student's projects. The course requires significant reading, reviewing, and writing. Students are expected to participate actively in class activities and to provide meaningful comments on the work of other students. For PhD students, this course will help accelerate your current and future research. For MS students, this course will provide a grounding in research methods that will aid your entry into research-oriented industrial positions and PhD studies. For undergraduates considering graduate studies, this course will help inform and accelerate that direction. For undergraduates, this course can be used to satisfy the 499Y requirement for Departmental and Multidisciplinary Honors students whose theses or projects have a substantial empirical component. Undergraduates must obtain approval of the Computer Science Honors Program Director prior to registering. Open to graduate Computer Science students only. 3 credits.

COMPSCI 611 Advanced Algorithms

INSTRUCTOR(S): Hedyeh Beyhaghi

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. Open to graduate Computer Science students only. Prerequisite: 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): Juan Zhai

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. We study the foundations of software testing and program analysis. We also explore how large language models (LLMs) and software agents can be used to improve software development and evaluation, and examine emerging challenges in AI-assisted software engineering, including testing and evaluating software built with LLMs and autonomous agents. The students will be required to complete regular homework assignments, 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 646 Information Retrieval

INSTRUCTOR(S): Razieh Rahimi

The course will cover basic and advanced techniques for text-based information systems. Topics covered include retrieval models, indexing and text representation, browsing and query reformulation, data-intensive computing approaches, evaluation, and issues surrounding implementation. The course will include a substantial project such as the implementation of major elements of search engines and applications. Open to Masters and PhD Computer Science students only. 3 credits.

COMPSCI 648 Quantum Information Systems

INSTRUCTOR(S): Donald Towsley

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

COMPSCI 660 Advanced Information Assurance

INSTRUCTOR(S): Amir Houmansadr

This course provides an in-depth examination of the fundamental principles of information assurance. While the companion course for undergraduates is focused on practical issues, the syllabus of this course is influenced strictly by the latest research. We will cover a range of topics, including authentication, integrity, confidentiality of distributed systems, network security, malware, privacy, intrusion detection, intellectual property protection, and more. Open to Masters and PhD Computer Science students only. 3 credits.

COMPSCI 661 Secure Distributed Systems

INSTRUCTOR(S): Gregory Stone

This is a class devoted to the study of securing distributed systems, with blockchain-based cryptocurrencies serving as our real platform of interest. We'll start with the fundamentals of Lamport's, Fischer's, and Douceur's results that fence-in all consensus system, and discuss Byzantine fault tolerance. We'll also look at the efficiency of the network architectures for peer-to-peer/distributed system communication and attacks on their security, such as denial of service attacks. And we'll review relevant applied cryptography such as elliptic curves. We'll discuss in detail the mechanisms of Bitcoin and Ethereum and we'll program distributed applications for Ethereum. Other topics include economics and finance. Assignments will include programming projects and reading research papers. The grade is also based on exams and participation in discussion. The course is based on a "flipped classroom". Open to Masters and PhD Computer Science students and Electrical + Computer Engineering students. 3 credits.

COMPSCI 666 Theory & Practice/Cryptography

INSTRUCTOR(S): Adam O'Neill

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

COMPSCI 682 Neural Networks: Modern Intro

INSTRUCTOR(S): Subhransu Maji

This course will focus on modern, practical methods for deep learning with neural networks. The course will begin with a description of simple classifiers such as perceptrons and logistic regression classifiers, and move on to standard neural networks, convolutional neural networks, some elements of recurrent neural networks, and transformers. The emphasis will be on understanding the basics and on practical application more than on theory. Many applications will be in computer vision, but we will make an effort to cover some natural language processing (NLP) applications as well.The current plan is to use Python and associated packages such as Numpy and Pytorch. Required background includes Linear Algebra, Probability and Statistics, and Multivariate Calculus. All assignments will be in the Python programming language. Open to graduate Computer Science students only. 3 credits.

COMPSCI 687 Reinforcement Learning

INSTRUCTOR(S): Philip Thomas

This course provides a thorough introduction and overview of reinforcement learning. Reinforcement learning algorithms repeatedly answer the question "What should be done next?", and they can learn via trial-and-error to answer these questions even when there is no supervisor telling the algorithm what the correct answer would have been. Applications of reinforcement learning span across medicine (How much insulin should be injected next? What drug should be given next?), marketing (What ad should be shown next?), robotics (How much power should be given to the motor?), game playing (What move should be made next?), environmental applications (Which countermeasure for an invasive species should be deployed next?), and dialogue systems (What type of sentence should be spoken next?), among many others. Broad topics covered in this course will include: Markov decision processes, reinforcement learning algorithms (model-based/model-free, batch/online, value function-based, actor-critics, policy gradient methods, etc.), and representations for reinforcement learning. Special topics may include ensuring the safety of reinforcement learning algorithms, hierarchical reinforcement learning, theoretical reinforcement learning, multi-agent reinforcement learning, and connections to animal learning. This course assumes a very strong mathematical background in calculus, linear algebra, and strategies for proving theorems. We will emphasize hands-on experience in class and through assignments, which will require implementing and applying many of the algorithms we discuss. Therefore, a strong background in programming is also necessary, as we will require that students implement sophisticated learning algorithms using C++ and/or Python. Finally, we assume that students have a background in machine learning (COMPSCI 589 or 689) and artificial intelligence (COMPSCI 683). Open to graduate Computer Science students only. 3 credits.

COMPSCI 689 Machine Learning

INSTRUCTOR(S): Justin Domke

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. Open to graduate Computer Science students only 3 credits.

COMPSCI 690F Trustworthy and Responsible AI

INSTRUCTOR(S): Eugene Bagdasarian

In the era of intelligent assistants, autonomous agents, and self-driving cars we expect AI systems to not cause harm and withstand adversarial attacks. In this course you will learn advanced methods of building AI models and systems that mitigate privacy, security, societal, and environmental risks. We will go deep into attack vectors and what type of guarantees current research can and cannot provide for modern generative models. The course will feature extensive hands-on experience with model training and regular discussion of key research papers. The course will operate under the assumption that students have previously taken NLP, general ML, and security classes before taking this course. 3 credits.

COMPSCI 690K Adv Robot Dynamics and Control

INSTRUCTOR(S): Donghyun Kim

This advanced course focuses on the dynamics and control of robotic systems, concepts crucial for understanding how robots move and interact with their physical surroundings. The content covered will go into greater depth than the more general course, CompSci 603 Robotics. Students will learn the kinematics and dynamics of robots with multiple degrees of freedom, as well as the analysis and control of these systems. Subjects covered include Lie group-based kinematics, Lagrangian dynamics, whole-body control, contact simulation, and actuation mechanisms. The course will utilize Google Colab and Python programming to develop simulation and analysis tools. Expect in-class exercises, weekly assignments/quizzes, a midterm examination, and a final project. Key topics to be explored are: actuators, homogeneous transformations, forward and inverse kinematics, 3D orientation representation, Newtonian dynamics, Lagrangian dynamics, whole-body control, and contact dynamics. While this course builds upon some themes introduced in CompSci 603, students are not required to take CompSci 603 before enrolling in this course. We will cover the foundational concepts necessary for the advanced study in this course. Open to graduate students in Computer Science, Electrical + Computer Engineering, and Mechanical + Industrial Engineering. 3 credits.

COMPSCI 690S AI Alignment

INSTRUCTOR(S): Scott Niekum

This course will focus on modern machine learning approaches to align agent objectives 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 as their capabilities increase and they are deployed more frequently in real-world settings. While the standard ML paradigm assumes that objective functions are provided as part of the problem specification, alignment research examines the profound challenges associated with specifying or learning about such objectives. This course covers a core set of topics that examine AI Alignment from a variety of angles, including behavioral cloning, inverse reinforcement learning, reinforcement learning from human feedback, robustness, scalable oversight, and mechanistic interpretability. We will examine applications of alignment ranging from robotics to large language models. Thus, the course aims to provide a broad overview of how AI researchers and practitioners have historically tried to design objectives and control the behaviors of AI systems, rather than adhering to any particular definition of alignment.   3 credits.

COMPSCI 691WM S-Wearable&MobileSens/ClincSci

INSTRUCTOR(S): Sunghoon Lee

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

COMPSCI 692FD S-FlowsDiffusionsBayesianStuff

INSTRUCTOR(S): Justin Domke

This seminar will cover recent advances at the intersection of flow-based generative modeling, diffusion processes, and Bayesian methods. Topics will include normalizing flows, diffusion models, inference methods based on diffusion, flow matching, and neural samplers. We will focus on recent research with state of the art results, paying particular attention to the underlying fundamental concepts (e.g. stochastic differential equations). By the end of the semester, participants should have a unified mathematical picture of how continuous-time dynamics can be learned to transport between distributions for both generation and inference. 1 credit.

COMPSCI 692L S-Natural Language Processing

INSTRUCTOR(S): Brendan O'Connor

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 701 Advanced Topics Computer Sci

INSTRUCTOR(S): Hao Zhang

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): Hao Zhang

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): Timothy Richards

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 during the first semester of their 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 H230 Comp Syst Principles colloq

INSTRUCTOR(S): Meng-Chieh Chiu

Students will explore topics from the 230 curriculum in greater detail, focusing on low-level systems concepts, memory and process management, and concurrency. The course emphasizes deeper understanding of how systems behave at the machine level through additional reading, discussion, and implementation. Students will also gain hands-on experience with an alternative systems programming language (e.g., Rust), enabling them to compare design choices and safety models across system languages. 1 credit.

COMPSCI H250 Intro To Computation (colloq)

INSTRUCTOR(S): Mordecai Golin

This course is an honors colloquium for COMPSCI 250. It will introduce generating functions and their applications in multiple topics in discrete mathematics, e,g., analysis of algorithms, combinatorics, number theory and probability. The class will have short weekly homeworks. In addition, each student will make a short oral presentation towards the end of the class on a topic chosen together with the instructor. Although absolutely not required, comfort with some very basic calculus, linear algebra and/or discrete probability theory would be a plus. Prerequisite: Students must be enrolled in or have completed COMPSCI 250. 1 credit.

COMPSCI H335 InsidetheBox:HowCmpsWrk colloq

INSTRUCTOR(S): Charles Weems

Honors section students are expected to meet weekly with the instructor. These meetings can be a combination of lecture, student research presentations, project progress reports, discussions, demonstrations of work, and problem solving. Students define their own program of enrichment, which will typically be either a research project or a project to develop an application of embedded systems. Students may work individually or in teams. Grading is based upon participation in the weekly meetings and the quality of the finished project. Prerequisite: Students must be enrolled in COMPSCI 335. 1 credit.

INFO 101 Introduction to Informatics

INSTRUCTOR(S): Michelle Trim

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

INFO 150 Mathmtcl Fndtn for Informatics

INSTRUCTOR(S): Justin Clarke

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. Open to INFORM majors. Prerequisite: a grade of C or or above in the following courses: CICS 110 (or CICS 160 or COMPSCI 119 or COMPSCI 121) with a grade of C or above and either: PSYCH 240, OIM 240, STATISTC 240, RES-ECON 212, SOCIOL 212, OR STATISTC 315/515, OR COMPSCI 240, with a grade of C or above. 4 credits.

INFO 324 Intro/Clinical Health Data Sci

INSTRUCTOR(S): Sunghoon Lee

This course aims to introduce the fundamentals of Clinical Health Informatics to prepare students as forerunners of the future of digital health care systems. More specifically, this course aims to teach students the fundamentals of and tools for quantitative analysis of clinical health data and the practical application of the tools on various health data. The detailed components of the course are as follows. Following an overview of the clinical health informatics industry, the course covers a broad range of introductory topics, including the structure of current health care systems, types of health data, the theory and practical use of quantitative analytic methodologies, and ethics related to healthcare. More specifically, this course will introduce key health informatics technologies and standards, including electronic health records, medical claims data, imaging data, free-text clinical notes, patient-reported outcomes, traditional and machine learning-based analytic algorithms, data visualization, and clinical research and experimental procedures. Note, however, that the course is not designed to introduce new types of machine learning or artificial intelligence algorithms for health-related data. This course is taught in the same classroom with students from COMPSCI 524. However, students enrolled in INFO 324 will be evaluated independently of students from COMPSCI 524. This course fulfills a concentration core requirement for the Health and Life Sciences track, and it can be used to fulfill an elective requirement for the Data Science concentration of the Informatics major. Open to INFORM majors. Prerequisite: Prerequisite: INFO 248 (or STATISTIC 315;515 or COMPSCI 240) with a grade of C or better. 4 credits.

INFO 348 Data Analytics with Python

INSTRUCTOR(S): Jayant Taneja

The modern world is awash with data, and making sense of it requires specialized skills. This course will expose students to commonly used data analytics techniques. Topics include the acquisition, manipulation, and transformation of structured data, exploratory data analysis, data visualization, and predictive modeling. Students in this course will learn and use the Python programming language and tools for working with data. Analysis will be performed using real data sets. Does not count as a CS Elective (BA or BS). Satisfies one of the Data Science Concentration requirements and counts as an elective for the Health and Life Sciences Concentration for the Informatics major. Open to INFORM majors. Prerequisite: INFO 248 and CICS 160 (or INFO 190T or COMPSCI 186 or 187), both with a grade of C or better. 3 credits.