

### **CICS 110 Foundations of Programming**

INSTRUCTOR(S): Meng-Chieh Chiu, Ghazaleh Parvini, Cole Reilly, STAFF

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. Open to freshmen, sophomores, and juniors in Computer Science and Informatics. No previous programming experience required. (Gen. Ed. R2) Open to freshmen, sophomores, and juniors in Computer Science and Informatics. 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 160 Object-Oriented Programming**

INSTRUCTOR(S): Jaime Davila, Cole Reilly

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) Open to freshmen, sophomores, and juniors in Computer Science and Informatics. Prerequisite: CICS 110 (previously INFO 190S) or COMPSCI 121 with a grade of C or above. 4 credits.

### **CICS 208 Defending Democracy Digital World**

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, Timothy Richards

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) Open to freshmen, sophomores, and juniors in Computer Science and Informatics. Prerequisite: CICS 160 (previously INFO 190T) with a grade of C or better. 4 credits.

### **CICS 237 Intro; Research in the Discipline**

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): Rui Wang

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. Open to freshman and sophomore BS-CS students. Prerequisite: COMPSCI 187 (or CICS 210) with a grade of C or better and completion of the R1 (Basic Math Skills) Gen. Ed. 4 credits.

---

## **CICS 291T S-CICS Transfer Success**

INSTRUCTOR(S): Emma Anderson

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

## **CICS 305 Social Issues in Computing**

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

Through a careful analysis and discussion of a range of computing issues, topics, and policies, we will explore various impacts of computers on modern society. This class satisfies the Junior Year Writing requirement by providing directed practice and specific instruction in a range of writing genres. Students will produce approximately 20-25 pages of polished written work over the course of the semester. 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.

## **CICS 580 Intro;Numerical Comp w; Python**

INSTRUCTOR(S): Ali MontazerAlghaem

This course is an introduction to computer programming for numerical computing. The course is based on the computer programming language Python and is suitable for students with no programming or numerical computing background who are interested in taking courses in machine learning, natural language processing, or data science. The course will cover fundamental programming, numerical computing, and numerical linear algebra topics, along with the Python libraries that implement the corresponding data structures and algorithms. The course will include hands-on programming assignments and a project. No prior programming experience is required. Familiarity with undergraduate-level probability, statistics and linear algebra is assumed. Does not count toward graduate degrees. Open to Graduate students only. 1 credit.

## **COMPSCI 119 Intro to Programming**

INSTRUCTOR(S): Allison Poh

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. Open to freshmen and sophomores in any major EXCEPT Computer Science. 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. This course is open to Freshmen, Sophomores and Juniors. Prerequisite: COMPSCI 121 with a grade of B or better (or COMPSCI 186 or CICS 160;INFO 190T with a grade of C or better). 1 credit.

## **COMPSCI 220 Programming Methodology**

INSTRUCTOR(S): Marius Minea

Development of individual skills necessary for designing, implementing, testing and modifying larger programs, including: design strategies and patterns, using functional and object-oriented approaches, testing and program verification, code refactoring, interfacing with libraries. There will be significant programming and mid-term and final examinations. Open to Computer Science majors only. 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. Open to Computer Science majors only. Prerequisite: CICS 210 (or COMPSCI 187) with a grade of C or above and COMPSCI 198C. 4 credits.

---

## **COMPSCI 240 Reasoning Under Uncertainty**

INSTRUCTOR(S): Shiting Lan, Ghazaleh Parvini

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. Open to Computer Science majors only. Prerequisites: COMPSCI 187 (or CICS 160;INFO 190T) and MATH 132 with a grade of C or better.

## **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. Open to COMPSCI and Math majors only. Prerequisite: a grade of C or better in COMPSCI 187 (or E+C-ENG 241 or CICS160;INFO 190T) and MATH 132. 4 credits.

## **COMPSCI 311 Introduction to Algorithms**

INSTRUCTOR(S): Ghazaleh Parvini, 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. 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). Open to senior and junior Computer Science majors only. Prerequisite: COMPSCI 187 (or CICS 210) 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, Yuriy Brun, Heather Conboy, Jaime Davila

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. Open to senior and junior Computer Science majors only. Prerequisite: COMPSCI 220 with a grade of 'C' or better. 4 credits.

## **COMPSCI 325 Intro;Human-Comp Interaction**

INSTRUCTOR(S): Ravi Karkar

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. Open to juniors and seniors in Computer Science or Informatics. Prerequisite: COMPSCI 220 (or 230) with a grade of C or better, or INFO 248 and COMPSCI 186 (or 187 or INFO 190T;CICS 160) with a grade of C or better in each. Note: as the name 'web programming' denotes, programming is a key component of this class. Previous background in JavaScript is strongly recommended. 4 credits.

## **COMPSCI 328 Mobile Health Sensing+Analytcs**

INSTRUCTOR(S): Deepak Ganesan

The typical smartphone comes equipped with a plethora of sensors for monitoring activity, speech patterns, social interactions, and location. In addition, mobile accessories such as wearable wristbands now enable routine and continuous monitoring of a host of physiological signals (e.g., heart rate, respiratory rate, oxygen saturation, and others.). In conjunction, these sensors can enable higher-order inferences about more complex human activities/behavioral states (e.g., activity patterns, stress, sleep, social interactions, etc.). Such ubiquitous sensing in daily life, referred to as mobile health sensing, promises to revolutionize our understanding of human activities and health conditions. This course is a hands-on introduction to personal health sensing through mobile phones. Please note that this is a \*programming-heavy\* class so a solid programming background is required. All programming assignments are in Python, so programming experience with \*Python\* is recommended. This course counts as an Elective for the CS Major. Open to juniors and seniors in Computer Science or Informatics. Prerequisite: 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 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): Alexandra Meliou

Computing has become data-driven, and databases are now at the heart of commercial applications. The purpose of this course is to provide a comprehensive introduction to the use of data management systems within the context of various applications. Some of the covered topics include application-driven database design, schema refinement, implementation of basic transactions, data on the web, and data visualization. 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. 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 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. Open to senior and junior Computer Science majors only. 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 377 Operating Systems**

INSTRUCTOR(S): Phuthipong Bovornkeeratiroj

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. Open to senior and junior Computer Science majors only. Prerequisite: COMPSCI 230 with a grade of 'C' or better. 4 credits.

## **COMPSCI 383 Artificial Intelligence**

INSTRUCTOR(S): William McNichols

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. Open to senior and junior Computer Science majors only. 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 410 Compiler Techniques**

INSTRUCTOR(S): Hui Guan

This course explores the basic problems in the translation of programming languages focusing on theory and common implementation techniques for compiling traditional block structured programming languages to produce assembly or object code for typical machines. The course involves a substantial laboratory project in which the student constructs a working compiler for a considerable subset of a realistic programming language, within a provided skeleton. The lectures are augmented by a discussion section that covers details of the programming language used to build the compiler, the operating system, the source language, and various tools. Use of computer required. Text: Engineering a Compiler, Cooper and Torczon. Open to senior and junior Computer Science majors only. This course counts as an Elective toward the CS Major. Prerequisites: COMPSCI 230 and either COMPSCI 250 or MATH 455, all with a grade of C or better.. 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 CS Elective for the CS Major. Open to juniors and seniors in Computer Science or Informatics. Students enrolled in the Undergraduate Certificate in Public Interest Technology are eligible via override. 3 credits.

---

## **COMPSCI 429 Software Engin Proj Management**

INSTRUCTOR(S): Gordon Anderson, Yuriy Brun, Heather Conboy, Jaime Davila

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. Open to undergraduate Computer Science majors only. 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. Open to senior and junior Computer Science majors only. This course counts as an Elective toward the CS Major. Prerequisites: COMPSCI 220 (or 230) and COMPSCI 311 and COMPSCI 345 with a grade of C or better. 3 credits.

## **COMPSCI 446 Search Engines**

INSTRUCTOR(S): James Allan

This course provides an overview of the important issues in information retrieval, and how those issues affect the design and implementation of search engines. The course emphasizes the technology used in Web search engines, and the information retrieval theories and concepts that underlie all search applications. Mathematical experience (as provided by COMPSCI 240) is required. You should also be able to program in Java or Python (other closely related languages may be acceptable). Open to senior and junior Computer Science majors only. This course counts as a CS Elective for the CS Major. Prerequisite: Prerequisite: COMPSCI 240 or COMPSCI 383 with a grade of 'C' or better. 3 credits.

## **COMPSCI 453 Computer Networks**

INSTRUCTOR(S): Parviz Kermani

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. Open to senior and junior Computer Science majors only. This course counts as a CS Elective for the CS Major. Prerequisite: 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, or LINGUIST 429B (previously LINGUIST 492B). 3 credits.

## **COMPSCI 490Q Quantum Information Science**

INSTRUCTOR(S): Filip Rozpedek

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. Open to junior and senior Computer Science and Informatics majors. Prerequisite: Math 132 and 235 and either COMPSCI 240 or STATISTIC 315;515, all with a grade of C or better. 3 credits.

## **COMPSCI 514 Algorithms for Data Science**

INSTRUCTOR(S): Andrew McGregor, 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. Open to Junior and Senior COMPSCI students. Undergraduate Prerequisite: COMPSCI 240 (or STATISTCS 515) and COMPSCI 311 with a grade of B+ or better in both, OR COMPSCI 240, STATISTCS 515, COMPSCI 311, Math 233, and Math 235 with a grade of C or better in each. 3 credits

## **COMPSCI 515 Algorithms,GameTheory+Fairness**

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 senior Computer Science students. This course counts as a CS Elective for the CS Major. Undergraduate Prerequisite: COMPSCI 240 and 250 with a grade of C or better in both. 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. Open to junior and senior Computer Science students. Undergraduate Prerequisites: COMPSCI 320 (or COMPSCI 220 and 326) with a grade of C or better. 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. Open to senior and junior Computer Science majors only Undergraduate Prerequisites: COMPSCI 230 and 240 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. Open to junior and senior Computer Science students. Undergraduate Prerequisites: COMPSCI 377 and 445 with a grade of C or better in each. 3 credits.

## **COMPSCI 560 Intro;Comp + Network Security**

INSTRUCTOR(S): Parviz Kermani

This course provides an introduction to the principles and practice of computer and network security with a focus on both fundamental principles and practical applications through hands-on approach. Many of the principles are taught through examples. The key topics of this course are a brief introduction to computer networking; applied cryptography; protecting users, data, and services; network security, and common threats and defense strategies. Students will complete a number of practical lab assignments as well as auto-graded quizzes/assignments. This course counts as a CS Elective for the CS Major. Open to undergraduate COMPSCI and CS-ENG students Undergraduate Prerequisite: COMPSCI 453 or E&C-ENG 374 with a grade of C or better. 3 credits.

## **COMPSCI 563 Internet Law and Policy**

INSTRUCTOR(S): Marvin Cable

This course is meant for those looking for legal knowledge for use in computing- and Internet-related endeavors. The course will include topics related to security, policy, and the use of machine learning and related technologies. In addition, students will be assigned law review articles and will learn to do legal research so that they can remain updated after the course ends. Topics covered are all in the context of the ubiquity of the Internet and computing, and they include: basic legal principles, contract law, substantive laws, intellectual property law, ethics, dealing with third parties, policy issues, and topical issues such as implications of applying machine learning technology. This course was formerly numbered as INFOSEC 690L. This course counts as a CS Elective for the CS Major. Open to junior and senior Computer Science students. Undergraduate Prerequisite: either COMPSCI 311, 383, or 360 (previously 460) with a grade of C or better. 3 credits.

## **COMPSCI 575 Combntrcs+Graph Thry**

INSTRUCTOR(S): Mark Wilson

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 576 Game Programming**

INSTRUCTOR(S): Evangelos Kalogerakis

Game Programming introduces students to concepts of computer game development, including 2D and 3D modeling, character design, animation, game art, basic game AI, audio and video effects. The course will help students build the programming skills needed to turn ideas into games. Both runtime systems and the asset pipelines will be covered. Students will work on various game programming exercises with modern game engines and graphics APIs. This course counts as a CS Elective for the CS Major (BA or BS). Open to Computer Science juniors and seniors Undergraduate Prerequisites: COMPSCI 311 with a grade of C or better (or COMPSCI 250 with a grade of B+ or better) and received a grade of C or better in COMPSCI 220 and MATH 235. 3 credits.

## **COMPSCI 589 Machine Learning**

INSTRUCTOR(S): Justin Domke

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. Open to Sr/Jr CS majors. Undergraduate Prerequisites: a grade of C or better in MATH 545, STAT 315/515 & CS 240; OR a C or better in MATH 545 & B+ or better in CS 240; OR a C or better in STAT 315/515 & CS 240 and B+ or better in MATH 233 & 235; OR B+ or better in CS 240, MATH 233 & 235 3 credits.



---

## **COMPSCI 590X Decarbonization + Data Science**

INSTRUCTOR(S): Jayant Taneja

This course examines applications of Data Science in the decarbonization of energy systems. The course covers (i.) basic energy systems concepts and background with US and global examples, (ii.) an introduction to relevant methods in statistical and geospatial data analytics and machine learning, and (iii.) trends and challenges affecting decarbonization in the electricity sector and beyond, with a focus on end-uses of energy. This course incorporates a significant programming component, with assignments on electricity supply implications from fluctuating solar photovoltaic and wind generation; residential energy system planning including heat pumps, solar photovoltaic systems, and energy storage; and demand side management, including smart appliances and electric vehicles. This course counts as a CS Elective for the CS Major (BA or BS) and as an Elective for the INFORM Major. Open to junior and senior Computer Science and Informatics students. Prerequisites: Either COMPSCI 240 (or STATISTC 515) and CICS 210 (or COMPSCI 187), or INFORM 348. 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 610 Compiler Techniques**

INSTRUCTOR(S): Hui Guan

This course explores the basic problems in the translation of programming languages focusing on theory and common implementation techniques for compiling traditional block structured programming languages to produce assembly or object code for typical machines. The course involves a substantial laboratory project in which the student constructs a working compiler for a considerable subset of a realistic programming language, within a provided skeleton. The lectures are augmented by a discussion section that covers details of the programming language used to build the compiler, the operating system, the source language, and various tools. Use of computer required. Open to graduate Computer Science students only. Text: Engineering a Compiler, Cooper and Torczon. 3 credits.

## **COMPSCI 611 Advanced Algorithms**

INSTRUCTOR(S): Hung Le

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

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 653 Computer Networking**

INSTRUCTOR(S): Arun Venkataramani

The goals of this course are to teach advanced fundamental principles underlying computer network systems. The course will cover topics in the following categories: 1) routing and transport protocols, 2) resource management, 3) datacenter network design, 4) software defined networking 5) wireless networks, and 6) network security. Prerequisites: Introductory (undergraduate level) courses in computer networks (e.g., COMPSCI 453), and algorithms (e.g., COMPSCI 311). Some familiarity with probability, optimization theory, and operating systems will be helpful. Open to Masters and PhD Computer Science students only. 3 credits.

### **COMPSCI 655 Performance Evaluation**

INSTRUCTOR(S): Donald Towsley

This course will provide an introduction to the tools and techniques needed to construct and analyze performance models of computer systems, distributed systems, and communication networks. The course covers three topics: i) analytical methods including discrete and continuous time Markov chain models, queues in isolation, queueing networks, and fluid queues; ii) computer/communication system measurement methodology including statistical inference and estimation of pertinent performance metrics, optimal measurement design, and bias removal; iii) applications to solving real world problems including model validation against measurements and/or simulation, case studies will be drawn from the areas of parallel and distributed systems, and networks. The goal is to teach fundamentals with a long half-life. Students are expected to have taken probability theory at at least the undergraduate level. 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 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 strong background in probability, calculus, linear algebra, and Python. Prior experience in signal/image processing is useful but not required. Open to graduate Computer Science students only. 3 credits.

---

## **COMPSCI 677 Distributed+Operating Systems**

INSTRUCTOR(S): Peter Klempner

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. Open to Online MS in Computer Science students only 3 credits.

## **COMPSCI 682 Neural Networks: Modern Intro**

INSTRUCTOR(S): Chuang Gan, 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): Bruno Castro da Silva

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 688 Probabilistic Graphical Models**

INSTRUCTOR(S): Daniel Sheldon

Probabilistic graphical models provide an intuitive 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 common 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; (b) exact and approximate inference methods; (c) estimation of the parameters and structure of graphical models; (d) broader topics in probabilistic inference for statistics and machine learning. Students entering the class should have good programming skills and knowledge of algorithms. Undergraduate-level knowledge of probability and statistics and a prior machine learning course are recommended. Open to graduate Computer Science students only. 3 credits.

---

## **COMPSCI 689 Machine Learning**

INSTRUCTOR(S): Benjamin Marlin

Machine learning is the computational study of artificial systems that can adapt to novel situations, discover patterns from data, and improve performance with practice. This course will cover the popular frameworks for learning, including supervised learning, reinforcement learning, and unsupervised learning. The course will provide a state-of-the-art overview of the field, emphasizing the core statistical foundations. Detailed course topics: overview of different learning frameworks such as supervised learning, reinforcement learning, and unsupervised learning; mathematical foundations of statistical estimation; maximum likelihood and maximum a posteriori (MAP) estimation; missing data and expectation maximization (EM); graphical models including mixture models, hidden-Markov models; logistic regression and generalized linear models; maximum entropy and undirected graphical models; nonparametric models including nearest neighbor methods and kernel-based methods; dimensionality reduction methods (PCA and LDA); computational learning theory and VC-dimension; reinforcement learning; state-of-the-art applications including bioinformatics, information retrieval, robotics, sensor networks and vision. Prerequisites: undergraduate level probability and statistics, linear algebra, calculus, AI; computer programming in some high level language. Open to graduate Computer Science students only 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 692P S-HotTopics;SoftwareEnginRsrch**

INSTRUCTOR(S): Yuriy Brun, Juan Zhai

This seminar covers research spanning programming languages, software engineering, security and systems. Open to graduate Computer Science students only 1 credit.

## **COMPSCI 692X S-MachineLearning;Bio Seq Data**

INSTRUCTOR(S): Anna Green

A seminar in which students will read, present, and discuss research papers on recent and advanced topics in computational biology, specifically related to machine learning models fit to biological sequence data (proteins and DNA). This semester, the seminar will primarily cover the following topics: foundation models of DNA and protein sequences (including transformer-based models), predicting the effects of biological mutations, predicting the structure of proteins (including AlphaFold), and supervised vs. unsupervised learning on sequences. 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 690U Computational Biology and Bioinformatics, or have comparable background. 1 credit.

## **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 879 TeachngAssist;TomorrowsFaculty**

INSTRUCTOR(S): Neena Thota

Teaching Assistants as Tomorrow's Faculty prepares Teaching Assistants (TAs) at the College of Information and Computer Sciences to fulfill their duties in an effective and pedagogically sound manner. The 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): Hung Le

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

## **COMPSCI H446 Search Engines (colloq)**

INSTRUCTOR(S): James Allan

This course is an honors colloquium for COMPSCI 446. Students will explore and discuss topics from the 446 curriculum in greater detail, with an intended focus on contemporary issues related to search engines -- for example, large language models, fairness, and/or explainability. Students will also collaboratively design programming project that builds on the programming project from 446. All students will produce a final report and may implement the expanded programming project to replace part of that report. Required reading (available free via a UMass subscription): Michael D. Ekstrand, Anubrata Das, Robin Burke and Fernando Diaz (2022), "Fairness in Information Access Systems", Foundations and Trends® in Information Retrieval: Vol. 16, No. 1-2, pp 1-177. DOI: 10.1561/1500000079. Additional readings from open-source and freely available material may be used. 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): David Barrington

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. Open to INFORM majors. 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). Open to INFORM majors. 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. Open to INFORM majors. 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. Prerequisites: a grade of C or better in the following courses: COMPSCI 121 (or CICS 110, 160, or COMPSCI 119) and either PSYCH 240, OIM 240, STATISTC 240, RES-ECON 212, SOCIO 212, or STATISTC 515. 4 credits.

## **INFO 324 IntroClinicalHealthInformatics**

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): Matthew Rattigan

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.

## **INFO 390C Intro;Comp Bio + Bioinform**

INSTRUCTOR(S): Anna Green

This course is designed to provide Informatics students with a broad, practical introduction to the field of computational biology and bioinformatics. The course will discuss at a high level the models and algorithms used to analyze biological sequence data, as well as practical applications and data analysis. Background in biology is not assumed. The primary focus of the course will be analysis of genomic data, including sequence alignment, genome assembly, genome annotation, phylogeny construction, mutation effect prediction, population genetics, RNA-seq data analysis, and genotype-phenotype association studies. 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 biology. This course counts as a CS Elective toward the CS Major and as an Elective toward the INFORM Major. Open to juniors and seniors in Computer Science or Informatics. Prerequisites: A grade of C or better in INFO 248 or a grade of C or better in both CICS 210 and COMPSCI 240 3 credits.