

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

INSTRUCTOR(S): STAFF, David Barrington, Cole Reilly, Timothy Richards, Rui Wang

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 160 Object-Oriented Programming**

INSTRUCTOR(S): Jaime Davila

This course will expose students to programming practices beyond the introductory level, concentrating on Object Oriented Programming techniques and an introduction to Data Structures. Students will also study and analyze the complexity of both the algorithms presented in class and of the algorithms they develop. This course also provides experience with the development and analysis of recursive algorithms and programs. Before taking this course, students are expected to have been exposed to the following concepts through a college-level course or equivalent in some high level computer programming language: input and output operations, conditional statements, loops, arrays, recursion, and functions/methods. The course places an emphasis on the careful design and testing of programs. (Gen. Ed. R2) Prerequisite: CICS 110 (previously INFO 190S) or COMPSCI 121. 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): Marc Liberatore

An introduction to the design, analysis, and implementation of data structures. This course teaches you how to build, test, debug, document, and evaluate objects that encapsulate data and their associated operations using programming constructs and data abstractions of a modern programming language. Concepts and techniques covered include linear and non-linear structures, recursive structures and algorithms, traversal algorithms, binary search trees, balanced trees, priority queues, union-find, hash tables, bloom filters, and graphs. We will also informally compare and contrast the run time efficiency of algorithms and their performance characteristics including the concept of worst-case running time analysis and the classification of algorithms in terms of constant, logarithmic, linear, log linear, quadratic, and exponential time using Big-O notation. (Gen. Ed. R2) Prerequisite: CICS 160 (previously INFO 190T). 4 credits.

### **CICS 256 Make: Physical Computing**

INSTRUCTOR(S): Md Farhan Tasnim Oshim

Inspired by the Maker movement, this course provides a hands-on introduction to physical computing: sensing and responding to the physical world using computers. Specific topics include: basic electronics and circuit design, microcontroller programming using Arduinos, sensing and responding to the physical world, rapid prototyping (3D printing and laser cutting etc.), soft circuits and wearable electronics. The course will encourage and empower students to invent, design, and build practical hardware projects that interact with the physical world. This course has a required lab section, and counts as one of the CS Lab Science Requirement courses for the BS-CS. Prerequisite: CICS 210 (or COMPSCI 187) and Basic Math Skills (R1). 4 credits.

### **CICS 290A 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 291T S-CICS Transfer Success**

INSTRUCTOR(S): Emma Anderson

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

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

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

Through a careful analysis and discussion of a range of computing issues, topics, and 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): Cole Reilly

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

## **COMPSCI 198C P-Intro; C Programming Language**

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

This practicum assumes general background and experience in computer programming (such as that provided by COMPSCI 121 or a similar introductory programming course) and some knowledge of data structures. Content will include basic C data types, declarations, expressions, statements, and functions; simple use of macros; some common library calls (such as formatted input/output); basic pointer manipulation using linked lists; and introduction to using standard tools (gcc and make). Required prerequisite for COMPSCI 230, effective Fall 2023. Prerequisite: CICS 160 (previously INFO 190T) or COMPSCI 121 or COMPSCI 186. 1 credit.

## **COMPSCI 220 Programming Methodology**

INSTRUCTOR(S): Marius Minea

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

## **COMPSCI 230 Computer Systems Principles**

INSTRUCTOR(S): Meng-Chieh Chiu

Large-scale software systems like Google - deployed over a world-wide network of hundreds of thousands of computers - have become a part of our lives. These are systems success stories - they are reliable, available ("up" nearly all the time), handle an unbelievable amount of load from users around the world, yet provide virtually instantaneous results. On the other hand, many computer systems don't perform nearly as well as Google - hence the now-cliché "the system is down." In this class, we study the scientific principles behind the construction of high-performance, scalable systems. The course begins with a discussion of C data representation, and moves up the stack from there to the features of modern architectures, assembly languages, and operating system services such as I/O, process, and synchronization. This class assumes students have either taken COMPSCI 198C or have equivalent experience in the C programming language. Prerequisite: CICS 210 (or COMPSCI 187) and COMPSCI 198C. 4 credits.

---

## **COMPSCI 240 Reasoning Under Uncertainty**

INSTRUCTOR(S): Peter Haas, STAFF

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 Network. Prerequisite: CICS 160 (previously INFO 190T or COMPSCI 187) and MATH 132. 4 credits.

## **COMPSCI 250 Introduction To Computation**

INSTRUCTOR(S): David Barrington, Ghazaleh Parvini

Basic concepts of discrete mathematics useful to computer science: set theory, strings and formal languages, propositional and predicate calculus, relations and functions, basic number theory. Induction and recursion: interplay of inductive definition, inductive proof, and recursive algorithms. Graphs, trees, and search. Finite-state machines, regular languages, nondeterministic finite automata, Kleene's Theorem. Problem sets, 2 midterm exams, timed final. Prerequisite: CICS 160 (previously INFO 190T or COMPSCI 187 or ECE 241) and MATH 132. 4 credits.

## **COMPSCI 311 Introduction to Algorithms**

INSTRUCTOR(S): Ghazaleh Parvini, Daniel Sheldon

This course will introduce you to a variety of techniques to design algorithms, such as divide and conquer, greedy, dynamic programming, and network flow. You will learn to study the performance of various algorithms within a formal, mathematical framework. You will also learn how to design very efficient algorithms for many kinds of problems and recognize problems that currently do not have efficient algorithms. Assignments may include programming: you should be able to program in Java, C, or some other closely related language. Mathematical experience (as provided by COMPSCI 250) is required. This course is required for the CS Major (BS) and counts as an Elective toward the CS Major (BA). Prerequisite: CICS 210 or COMPSCI 187, and either COMPSCI 250 or MATH 455. 4 credits.

## **COMPSCI 320 Software Engineering**

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

In this course, students learn and gain practical experience with software engineering principles and techniques. The practical experience centers on a semester-long team project in which a software development project is carried through all the stages of the software life cycle. Topics in this course include requirements analysis, specification, design, abstraction, programming style, testing, maintenance, communication, teamwork, and software project management. Particular emphasis is placed on communication and negotiation skills and on designing and developing maintainable software. Use of computer required. Several written assignments, in-class presentations, and a term project. This course satisfies the IE Requirement and counts as a CS Elective for the CS Major. Prerequisite: COMPSCI 220. 4 credits.

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

INSTRUCTOR(S): Cheryl Swanier

Human-Computer Interaction design is "design for human use". Computers are a ubiquitous part of many interactions in our lives, from the mundane everydayness of light switches and "smart" vending machines to entertainment and education to sophisticated instruments and complex energy and defense systems. In this course, we will challenge you to broaden your grasp of what a user interface can and should be, and try your hand at doing better yourself. It is a fast-paced, hands-on, project-based experience that will challenge many of your ideas of what computer science is and can be. It is designed around active lecture sessions supported by readings, working classes, and team projects, where students practice and explore the concepts introduced in lecture, and go well beyond them to learn and apply HCI techniques that build into group projects. More specifically, the course adopts a human-centered design (HCD) approach and teaches a highly iterative process called design thinking. The design thinking process draws heavily on the fundamentals of human-computer interaction (HCI) methods. I also cover design methodologies, evaluation methodologies (both quantitative and qualitative), human information processing, cognition, and perception. This course counts as a CS Elective toward the CS Major and as a Required Core for the INFORM Major. Prerequisite: CS Majors: CICS 210 or COMPSCI 187; INFORM Majors: INFO 248 and CICS 160 (previously INFO 190T or COMPSCI 186 COMPSCI 187). 3 credits.

---

## **COMPSCI 326 Web Programming**

INSTRUCTOR(S): Meng-Chieh Chiu

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

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

INSTRUCTOR(S): Deepak Ganesan

In recent years, the ability to continuously monitor activities, health, and lifestyles of individuals using sensor technologies has reached unprecedented levels. The typical smartphone comes equipped with a plethora of sensors for monitoring activity, speech patterns, social interactions, and location. In addition, mobile accessories such as wearable wristbands and chestbands now enable routine and continuous monitoring of a host of physiological signals (e.g., heart rate, respiratory rate, skin Conductance, and others.). In conjunction, these sensors can enable higher-order inferences about more complex human activities/behavioral states (e.g., activity patterns, stress, sleep, etc.). Such ubiquitous sensing in daily life, referred to as mobile health sensing and monitoring, promises to revolutionize our understanding of human activities and health conditions. This course is an introduction to personal health sensing and monitoring through mobile phones and on-body sensors and addresses several aspects including mobile devices and applications for health, sensor data quality and reliability challenges, inference of key health assessments from sensor data including such as activity patterns, sleep patterns, or stress, sensor data visualization and feedback, and practical considerations such as battery lifetime. This course counts as an Elective for the CS Major. Prerequisite: CS Majors: CICS 210 or COMPSCI 187; INFORM Majors: INFO 248 and CICS 160 (previously INFO 190T or COMPSCI 186 or COMPSCI 187). 3 credits.

## **COMPSCI 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. Prerequisite: COMPSCI 220 (or COMPSCI 230). 3 credits.

## **COMPSCI 345 Pract + Appl of Data Managemnt**

INSTRUCTOR(S): Gordon Anderson, Marco Serafini

Computing has become data-driven, and databases are now at the heart of commercial applications. The purpose of this course is to provide a comprehensive introduction to the use of data management systems within the context of various applications. Some of the covered topics include application-driven database design, schema refinement, implementation of basic transactions, data on the web, and data visualization. The class will follow a flipped classroom model; students will be required to review materials in preparation for each week and they will work collaboratively on practical problems in class. This course counts as a CS Elective toward the CS Major. Students who have completed COMPSCI 445 are not eligible to take this course without instructor permission. Prerequisite: CS Majors: CICS 210 or COMPSCI 187; INFORM Majors: INFO 248 and CICS 160 (previously INFO 190T or COMPSCI 186 or COMPSCI 187). 3 credits.

## **COMPSCI 360 Intro Computr + Ntwrk Security**

INSTRUCTOR(S): Christopher Misra

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

---

## **COMPSCI 365 Digital Forensics**

INSTRUCTOR(S): Brian Levine

The goal of forensics is to gather artifacts for refinement into evidence that supports or refutes a hypothesis about an alleged crime or policy violation. Done correctly, forensics represents the application of science to law. The techniques can also be abused to thwart privacy. This course is a broad introduction to forensic investigation of digital information and devices. We cover the acquisition, analysis, and courtroom presentation of information from file systems, operating systems, networks, mobile applications, and the like. Students do not need experience with these systems. We review the use of some professional tools, however, the primary goal of the course is to understand why and from where artifacts are recoverable in these systems. Most assignments involve coding forensic tools. For a small portion of the class, we cover some relevant issues from the law, privacy, and current events. Thus, the class serves the well-rounded student who is eager to participate in class discussion on a variety of technical and social issues. This course counts as an Elective toward the CS and INFORM Majors. Prerequisite: COMPSCI 230. 3 credits.

## **COMPSCI 377 Operating Systems**

INSTRUCTOR(S): Peter Klemperer

In this course we examine the important problems in operating system design and implementation. The operating system provides a well-known, convenient, and efficient interface between user programs and the bare hardware of the computer on which they run. The operating system is responsible for allowing resources (e.g., disks, networks, and processors) to be shared, providing common services needed by many different programs (e.g., file service, the ability to start or stop processes, and access to the printer), and protecting individual programs from one another. The course will start with a brief historical perspective of the evolution of operating systems over the last fifty years, and then cover the major components of most operating systems. This discussion will cover the tradeoffs that can be made between performance and functionality during the design and implementation of an operating system. Particular emphasis will be given to three major OS subsystems: process management (processes, threads, CPU scheduling, synchronization, and deadlock), memory management (segmentation, paging, swapping), file systems, and operating system support for distributed systems. This course counts as a CS Elective for the CS Major. Prerequisites: COMPSCI 230. 4 credits.

## **COMPSCI 383 Artificial Intelligence**

INSTRUCTOR(S): Scott Niekum

The course explores key concepts underlying intelligent systems, which are increasingly deployed in consumer products and online services. Topics include problem solving, state-space representation, heuristic search techniques, game playing, knowledge representation, logical reasoning, automated planning, reasoning under uncertainty, decision theory and machine learning. We will examine the use of these concepts in the design of intelligent agents in the context of several applications. Students should be comfortable programming in Python. This course counts as an Elective toward the CS and INFORM Majors. Prerequisite: COMPSCI 220 (or COMPSCI 230) and COMPSCI 240 (or STATISTC 515). 3 credits.

## **COMPSCI 391L S-Cmptr Crime Law:Invstg;Prvcy**

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. Prerequisite: COMPSCI 230 and ENGLWRIT 112. 3 credits.

## **COMPSCI 420 Software Entrepreneurship**

INSTRUCTOR(S): Neena Thota

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

---

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

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

The purpose of this course is to provide students with practical experience in the management of software development projects. Students in this course will gain this experience by serving as software development team technical managers for teams of software engineering students in COMPSCI 320. As project managers, the students in COMPSCI 429 will be responsible for: supervising and managing the work of teams of COMPSCI 320 students; interfacing with the other COMPSCI 429 students managing other teams in the course; interfacing with the course instructor, course TA, and course customer. COMPSCI 429 students will be assigned readings in software engineering project management to provide a theoretical basis for their work in this course. But the majority of work in the course will be related to the actual management of assigned development teams. As team managers, COMPSCI 429 students will set goals and schedules for their teams, track and report team progress, negotiate with leaders of other teams and the course customer, and evaluate the work of members of their teams. COMPSCI 429 course assignments may include: written team goals, plans and schedules; periodic reports on team progress; documentation of agreements reached with other team leaders and customers; evaluations of the applicability of theoretical papers to the work of this course. This course will meet at the same times and places as COMPSCI 320. Additional meetings with team members and other students in COMPSCI 429 are also expected to be arranged by mutual agreement. An additional one hour weekly meeting of all of the students in COMPSCI 429 is required. 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 446 Search Engines**

INSTRUCTOR(S): Ali MontazerAlghaem

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

## **COMPSCI 453 Computer Networks**

INSTRUCTOR(S): Parviz Kermani

Introduction to computer communication networks and protocols. Fundamental concepts in the design and analysis of computer networks. Topics include: layered network architectures, networked applications, network programming interfaces, transport, congestion, routing, data link protocols, local area and data center networks, network security, and wireless networks. Examples drawn from the Internet (e.g., TCP, UDP, and IP) protocol suite. Homework assignments involve programming and written tasks. In Fall 2020, this course will be taught as a primarily online class, without in-class lectures, but with extensive on-line material including recorded video material, interactive exercises, and online discussion. There will be an optional weekly, one-hour in-class and on-line open discussion period. This course counts as a CS Elective for the CS Major. Prerequisite: Experience programming; COMPSCI 230 or COMPSCI 377. 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. TThis course counts as a CS Elective for the CS Major, as well an Any 2 menu choice for the former Security & Privacy track. Prerequisite: COMPSCI 326, COMPSCI 345, COMPSCI 377, COMPSCI 453, or COMPSCI 497P. 3 credits.

## **COMPSCI 490A App;NaturalLanguageProcessing**

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. An alternate prerequisite of LINGUIST 492B is acceptable for Linguistics majors. 3 credits.

---

## **COMPSCI 490STA Scalable Web Systems**

INSTRUCTOR(S): Timothy Richards

The web has become a large and complex area for application development. Access to an abundance of open source languages, libraries, and frameworks has led to the quick and easy construction of a variety of applications with several moving parts working in coordination to present to the user the illusion of a single program. In reality, web applications are extremely difficult to get right. They involve a large collection of coordinated services, multiple databases, complicated user interfaces, security and performance issues, and ever changing 3rd party services, spread across physical and virtual machines. These complications are further stressed by the large number of concurrent users that access these applications every second. This course will investigate several well known web-based applications and the technology and software architecture used to scale these applications. We will also study a specific topic related to scalability in software design in the context of web application architecture. This course counts as an Elective for the CS and INFORM Majors. Prerequisite: COMPSCI 320 or COMPSCI 326. 3 credits.

## **COMPSCI 513 Logic in Computer Science**

INSTRUCTOR(S): Marius Minea

Rigorous introduction to mathematical logic from an algorithmic perspective. Topics include: Propositional logic: Horn clause satisfiability and SAT solvers; First Order Logic: soundness and completeness of resolution, compactness theorem. We will use the Coq theorem prover and Datalog. This course counts as a CS Elective for the CS Major. Undergraduate Prerequisite: COMPSCI 250 and COMPSCI 311. 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 515) and COMPSCI 311 both with a grade of B+ or better, OR (COMPSCI 240 and STATISTCS 515 and COMPSCI 311 and MATH 233 and MATH 235, all with a C or better). 3 credits

## **COMPSCI 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. This course counts as a CS Elective for the CS Major. Undergraduate Prerequisite: COMPSCI 240 and COMPSCI 250. 3 credits.

## **COMPSCI 520 Thry + Practice;Software Engin**

INSTRUCTOR(S): Heather Conboy

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

## **COMPSCI 524 Health Informatics + Data Sci**

INSTRUCTOR(S): Sunghoon Lee

This course aims to introduce the discipline of Clinical Health Informatics and data to prepare our students as forerunners of the future of digital health care systems. Followed by an overview of the health informatics industry, it covers a broad range of introductory topics, such as the structure of current health care systems, various types of health data, theoretical framework and practice use of quantitative analytic methodologies, and ethics. More specifically, this course will teach important health informatics technologies and standards, such as electronic health records, medical claims data, imaging/free-text clinical notes, patient-reported outcomes, traditional and machine learning-based analytic algorithms, data visualization, and clinical research and experimental procedures. It is noteworthy that the course is not designed to introduce new types of machine learning or artificial intelligence algorithms for health-related data. Instead, this course aims to teach students how to comprehend and process health data and apply advanced computational models to solve important clinical problems. This course counts as a CS Elective for the CS Major. Undergraduate Prerequisite: COMPSCI 240 or STATISTCS 515. 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. 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. Undergraduate Prerequisite: COMPSCI 453 or E&C-ENG 374. 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. Undergraduate Prerequisite: COMPSCI 311, COMPSCI 383, or COMPSCI 360 (previously 460). 3 credits.

## **COMPSCI 578 Distributed Computing + Systems**

INSTRUCTOR(S): Arun Venkataramani

This course will teach the principles and practice of distributed systems as applied in today's cloud computing environments. The course will cover fundamental concepts in distributed computing including distributed clocks, consistency, fault tolerance, and consensus. The course will also cover popular cloud computing service models, related programming models, datacenter architectures, software-defined networking, and security and privacy issues in public clouds. The course will expose students to public cloud platforms such as Amazon EC2, Google Cloud Engine, Microsoft Azure etc. The assignments in the course will include a nontrivial programming component that assumes familiarity with multithreaded programming and network programming. This course counts as a CS Elective for the CS Major. Undergraduate Prerequisite: COMPSCI 377 or an equivalent first course in operating systems. 3 credits.

## **COMPSCI 589 Machine Learning**

INSTRUCTOR(S): Hui Guan

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

## **COMPSCI 611 Advanced Algorithms**

INSTRUCTOR(S): Hung Le

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



---

## **COMPSCI 613 Adv Logic in Computer Science**

INSTRUCTOR(S): Marius Minea

Rigorous introduction to mathematical logic from an algorithmic perspective. Topics include: Propositional logic: Horn clause satisfiability and SAT solvers; First Order Logic: soundness and completeness of resolution, compactness theorem, automatic theorem proving, model checking. We will learn about and use the Coq theorem prover, Datalog, a Model Checker, and SAT and SMT solvers. Prerequisites: Students taking this course should have undergraduate preparation in discrete math and algorithms. Requirements will include readings, class participation, weekly problem sets, a midterm and a final project. 3 credits.

## **COMPSCI 635 Modern Computer Architecture**

INSTRUCTOR(S): Charles Weems

This course examines the structure of modern computer systems. We explore recent research results that are influencing modern machine organizations, then consider specific features and their impact on software and performance. These may include superscalar issue, caches, pipelines, branch prediction, and parallelism. Midterm and final exams, individual projects, homework, in-class exercises. Prerequisites: COMPSCI 535 or equivalent. 3 credits.

## **COMPSCI 646 Information Retrieval**

INSTRUCTOR(S): Hamed Zamani

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. 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. 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. 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. 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". 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 TensorFlow. Required background includes Linear Algebra, Probability and Statistics, and Multivariate Calculus. All assignments will be in the Python programming language. 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). 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. 3 credits.

## **COMPSCI 690C Foundations Applied 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. 3 credits.

## **COMPSCI 690F Responsible AI**

INSTRUCTOR(S): Przemyslaw Adam Grabowicz

The real-world deployment of machine learning models faces a series of lateral challenges affecting model trustworthiness, such as domain generalization, dataset shifts, causal validity, explainability, fairness, representativeness, and transparency. These challenges become increasingly important in techno-social systems affecting human high-stake decision making, which is often regulated by law. In this course, students will learn techniques for robust model evaluation, model selection, causal discovery, explainable and fair artificial intelligence, and interpretable models. In addition, students will reason about representativeness, transparency, and legal aspects of techno-social systems. The course will review both cutting-edge research and relevant portions of recent open-access textbooks. Coursework includes reading recent research papers, programming assignments, and a final group project. After completing the course, students should be able to develop, investigate, evaluate, and deploy artificial intelligence systems more responsibly. 3 credits.

---

## **COMPSCI 691DD S-Res Methods in 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. 3 credits.

## **COMPSCI 692L S-Natural Language Processing**

INSTRUCTOR(S): Mohit Iyer

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):

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): David Barrington

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): Ghazaleh Parvini

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.

---

### **INFO 101 Introduction to Informatics**

INSTRUCTOR(S): Cheryl Swanier

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

### **INFO 150 Mathmtcl Fndtn for Informatics**

INSTRUCTOR(S): Cole Reilly

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 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. Prerequisite: INFO 248 (or STATISTCS 515 or COMPSCI 240). 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. Prerequisite: INFO 248 and CICS 210 (or COMPSCI 186 or COMPSCI 187). 3 credits.