
CICS 197R Special Topics - Introduction to Data Analysis in R

INSTRUCTOR(S): Jasper Mcchesney

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

CICS 256 Make: A Hands-on Introduction to Physical Computing

INSTRUCTOR(S): Donghyun Kim

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

CICS 291S Seminar - CICS Second Year Pathways

INSTRUCTOR(S): Emma Anderson

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

CICS 298A Practicum - Leadership: Communicating Across Expertise

INSTRUCTOR(S): Emma Anderson

No matter where you end up in tech, you will need to explain concepts, products and ideas to people with different technical backgrounds. This course is intended to help prepare you for these communication tasks. Through the lens of tutoring, we will work on explaining technical ideas clearly and compassionately to others. We will do some theoretical study, including a history of CS education as well as brain and learning science, and some practice, including tutoring beginning students in CS. This course is intended for a broad range of students looking to pursue careers in tech, but will be particularly useful for those who are currently UCAs or intending to apply for UCA positions in the future. Prerequisite: COMPSCI 186 or 187. 1 credit.

CICS 305 Social Issues in Computing

INSTRUCTOR(S): Elena Kalodner-Martin, Siobhan Mei, Justin Obara, 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. Prerequisite: ENGLWRIT 112 (or English Writing waiver), COMPSCI 220, COMPSCI 230 and COMPSCI 240 (or 250), OR ENGLWRIT 112 (or English Writing waiver) AND INFO 248. 3 credits.

CICS 490E EMBER: Introduction to Research Methods in Computing

INSTRUCTOR(S): Michelle Trim

The course will contribute to the overall goal of making research in computing more accessible to students from historically underrepresented groups in computer science. To that end, the workshop topics will include a combination of practical, social, and research-related topics. Examples of those topics include cultivating curiosity, building confidence and combatting imposter syndrome, summarizing findings, visualizing data, forming hypotheses, applications in computing, and developing collaboration strategies. These topics will dovetail with the more intensive four-week mini-research module offered by the various participating computing research labs and their faculty leads. This class is part of the EMBER (Energizing Mentoring and Broadening Exposure to Research) project in CICS. 1 credit.

COMPSCI 119 Introduction to Programming

INSTRUCTOR(S): William Verts

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 121 Introduction to Problem Solving with Computers

INSTRUCTOR(S): Eduardo Calle Ortiz, Jaime Davila, Collin Giguere, Peter Klemperer, Adam Kohan, Spencer Lane, Ghazaleh Parvini, Emily Pruc, Cole Reilly

COMPSCI 121 provides an introduction to problem solving and computer programming using the programming language Java. The course teaches how real-world problems can be solved computationally using the Java programming language. Concepts and techniques covered include data types, expressions, objects, methods, top-down program design, program testing and debugging, interactive programs, conditionals, iteration, and arrays. No previous programming experience is required; however, this course is intended for Computer Science majors or those who plan on applying to the major. Non-majors are strongly encouraged to take one of our programming courses designed for non-majors. Use of a laptop computer on which you can install software is required. Prerequisite: R1 (or a score of 20 or higher on the math placement test Part A), or one of the following courses: MATH 101&102 or MATH 104 or MATH 127 or MATH 128 or MATH 131 or MATH 132. 4 credits.

COMPSCI 145 Representing, Storing, and Retrieving Information

INSTRUCTOR(S): William Verts

An introductory course in the use of data in computer systems, a core course for the Information Technology certificate. Analog vs. Digital representations of information. Formats for representing text, numbers, sound, images, etc., as strings of bits. Equations of lines and curves, modeling of synthetic scenes (i.e., ray tracing), and exploring the frequency domain. Basic information theory, use and limitations of file compression and encryption. Structured databases and how to use them. Review of historical, pre-computer methods of information representation. Prerequisites: "Basic computer literacy", i.e., user-level familiarity with a modern operating system and some experience with application programs. Tier I math skills. Recommended for First Year and Sophomore Non-Majors. Prerequisite: R1. 3 credits.

COMPSCI 186 Using Data Structures

INSTRUCTOR(S): Marc Liberatore

COMPSCI 186 introduces foundational abstract data types and algorithms. The main focus is on the use of data structures in designing and developing programs to solve problems in a variety of domains. Specific topics include lists, sets, maps, graphs, stacks, queues, searching, and sorting. There will be weekly programming assignments, programming and written exercises in discussion sections, regular quizzes, and a cumulative final exam. This course is not a substitute for COMPSCI 187. If unsure of whether this course or COMPSCI 187 is more appropriate, contact instructor. Prerequisites: COMPSCI 121 and Basic Math Skills R1 (or a score of 20 or higher on the math placement test Part A), or one of the following courses: MATH 101&102 or MATH 104 or MATH 127 or MATH 128 or MATH 131 or MATH 132). 4 credits.

COMPSCI 187 Programming with Data Structures

INSTRUCTOR(S): Meng-Chieh Chiu, Neena Thota

The course introduces and develops methods for designing and implementing abstract data types using the Java programming language. The main focus is on how to build and encapsulate data objects and their associated operations. Specific topics include linked structures, recursive structures and algorithms, binary trees, balanced trees, and hash tables. These topics are fundamental to programming and are essential to other courses in computer science. The course involves weekly programming assignments, in-class quizzes, discussion section exercises, and multiple exams. Prerequisites: COMPSCI 121 (or equivalent Java experience). A grade of B or better in COMPSCI 121 (or a grade of C or better in COMPSCI 186 (or COMPSCI 190D) is required for students enrolling in COMPSCI 187 and Basic Math Skills (R1). Basic Java language concepts are introduced quickly; if unsure of background, contact instructor. 4 credits.

COMPSCI 198C Practicum - Introduction to the 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 (such as that provided by COMPSCI 187). 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). 1 credit.

COMPSCI 220 Programming Methodology

INSTRUCTOR(S): Jaime Davila, Marius Minea

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

COMPSCI 230 Computer Systems Principles

INSTRUCTOR(S): Meng-Chieh Chiu

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

COMPSCI 240 Reasoning Under Uncertainty

INSTRUCTOR(S): Shiting Lan

Development of mathematical reasoning skills for problems that involve uncertainty. Each concept will be illustrated by real-world examples and demonstrated through in-class and homework exercises. Counting and probability -- basic counting problems, probability definitions, mean, variance, binomial distribution, discrete random variables, continuous random variables, Markov and Chebyshev bounds, Laws of large number, and central limit theorem. Probabilistic reasoning -- conditional probability and odds, Bayes' Law, Markov Chains, Bayesian Network, Markov Decision Processes. Prerequisites: COMPSCI 187 and MATH 132. 4 credits.

COMPSCI 250 Introduction to Computation

INSTRUCTOR(S): David Barrington, Kyle Doney

Basic concepts of discrete mathematics useful to computer science: set theory, strings and formal languages, propositional and predicate calculus, relations and functions, basic number theory. Induction and recursion: interplay of inductive definition, inductive proof, and recursive algorithms. Graphs, trees, and search. Finite-state machines, regular languages, nondeterministic finite automata, Kleene's Theorem. Problem sets, 2 midterm exams, timed final. Prerequisite: COMPSCI 187 (or ECE 241) and MATH 132. 4 credits.

COMPSCI 311 Introduction to Algorithms

INSTRUCTOR(S): Md Abdul Aowal, Marius Minea

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 counts as an Elective toward the CS Major. Prerequisite: COMPSCI 187 and either COMPSCI 250 or MATH 455. 4 credits.

COMPSCI 320 Introduction to Software Engineering

INSTRUCTOR(S): Gordon Anderson, David Fisher

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

COMPSCI 325 Introduction to Human Computer Interaction

INSTRUCTOR(S): Cheryl Swanier

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

COMPSCI 326 Web Programming

INSTRUCTOR(S): Timothy Richards

The web is arguably today's most important application platform. Web browsers run on practically every device, and even many phone applications are in fact web applications under the covers. This course will cover a broad range of client-side web technologies, including HTTP itself, HTML5, CSS, and JavaScript; it will additionally cover key concepts for the server side of web applications, including key value stores and SQL servers. This course will also cover key concepts and technologies including AJAX, JavaScript libraries (e.g., jQuery), and web security. This course is hands-on and heavily project-based; students will construct a substantial dynamic web application based on the concepts, technologies, and techniques presented during lectures and in readings. This course satisfies the IE Requirement for CS and INFORM Majors. Prerequisites: COMPSCI 220 (OR COMPSCI 230), OR INFO 248 AND COMPSCI 186 (OR COMPSCI 187). Note: as the name 'web programming' denotes, programming is a key component of this class. Previous background in JavaScript is strongly recommended. 4 credits.

COMPSCI 345 Practice and Applications of Data Management

INSTRUCTOR(S): Jaime Davila

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

COMPSCI 348 Principles of Data Science

INSTRUCTOR(S): David Jensen

Data science uses various concepts, practices, algorithms, and systems to extract knowledge and insights from data. It encompasses techniques from machine learning, statistics, databases, visualization, and several other fields. When properly integrated, these techniques can help human analysts make sense of vast stores of digital information. This course presents the fundamental principles of data science, familiarizes students with the technical details of representative algorithms, and connects these concepts to applications in industry, science, and government, including fraud detection, marketing, scientific discovery, and web mining. The course assumes that students are familiar with basic concepts and algorithms from probability and statistics. This course counts as a CS Elective toward the CS Major. Prerequisites: COMPSCI 187, COMPSCI 240, and COMPSCI 250 (or MATH 455). 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 a CS Elective toward the CS and INFORM Majors. Prerequisites: COMPSCI 230. 3 credits.

COMPSCI 373 Introduction to Computer Graphics

INSTRUCTOR(S): Evangelos Kalogerakis

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

COMPSCI 377 Operating Systems

INSTRUCTOR(S): Mark Corner

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

COMPSCI 383 Artificial Intelligence

INSTRUCTOR(S): Matthew Rattigan

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

COMPSCI 390A Introduction to Machine Learning

INSTRUCTOR(S): Cooper Sigris, Philip Thomas

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

COMPSCI 390R Reverse Engineering and Vulnerability Analysis

INSTRUCTOR(S): Brian Levine

Many software developers aren't aware of how to properly write secure code. This course covers practical skills in reverse engineering and binary exploitation, and examines the techniques used by hackers in recent major security incidents. The course objective is to provide students with a strong understanding of attack patterns, and to ensure students implement more secure coding practices in their own code. This course begins with an introduction to Intel-based assembly, reverse engineering, vulnerability analysis, and various forms of Linux-focused binary exploitation. The course then covers stack, heap and Linux kernel-based exploitation, and dive into common defensive mitigations such as ASLR, NX and Stack Cookies alongside techniques to bypass each of them. This course is focused on low-level software written in C, which is why an understanding of the topics covered in CS230 is required. This course counts as an Elective for the CS Major. Prerequisites: COMPSCI 230 (or E&C-ENG 322 or E&C-ENG 373). 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. This course counts as an Elective toward the CS Major. Prerequisites: COMPSCI 230 and either COMPSCI 250 (or MATH 455). 3 credits.

COMPSCI 420 Software Entrepreneurship

INSTRUCTOR(S): Neena Thota

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

COMPSCI 445 Information Systems

INSTRUCTOR(S): Marco Serafini

This course is an introduction to the efficient management of large-scale data. The course includes principles for representing information as structured data, query languages for analyzing and manipulating structured data, and core systems principles that enable efficient computation on large data sets. Classical relational database topics will be covered (data modeling, SQL, query optimization, concurrency control), as well as semi-structured data (XML, JSON), and distributed data processing paradigms (e.g. MapReduce and Spark). Additional application topics may include web application development, data integration, processing data streams, database security and privacy. This course counts as an Elective toward the CS Major. Prerequisites: COMPSCI 220 (or 230) and COMPSCI 311 and COMPSCI 345. 3 credits.

COMPSCI 446 Search Engines

INSTRUCTOR(S): David Fisher

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

COMPSCI 453 Computer Networks

INSTRUCTOR(S): Arun Venkataramani

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

COMPSCI 466 Applied Cryptography

INSTRUCTOR(S): Adam O'Neill

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

COMPSCI 490U Introduction to User Experience Research

INSTRUCTOR(S): Ya Xiong

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

COMPSCI 491G Seminar - Computer Networking Lab

INSTRUCTOR(S): Parviz Kermani

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

COMPSCI 501 Formal Language Theory

INSTRUCTOR(S): David Barrington

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

COMPSCI 508 Ethical Considerations in Computing

INSTRUCTOR(S): Michelle Trim

This course considers an array of ethical issues in computing. Readings, class discussions, and guest speakers will cover topics related to avenues of development in artificial intelligence, privacy, identity, inclusiveness, environmental responsibility, internet censorship, network policy, plagiarism, intellectual property and others. All examples will be drawn from current and recent events with readings from a range of sources both journalistic and academic. Course assignments will have real world applications and offer students opportunities for developing their speaking and writing skills. Class discussions will be a vibrant component of the course. Open to Graduate students only. Undergraduate CS Majors with permission of instructor (counts as an Elective toward the CS Major). 3 credits.

COMPSCI 514 Algorithms for Data Science

INSTRUCTOR(S): Andrew McGregor

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

COMPSCI 520 Theory and Practice of Software Engineering

INSTRUCTOR(S): Heather Conboy

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

COMPSCI 528 Mobile and Ubiquitous Computing

INSTRUCTOR(S): Tauhidur Rahman

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

COMPSCI 529 Software Engineering Project Management

INSTRUCTOR(S): Gordon Anderson, David Fisher

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 529 will be responsible for: supervising and managing the work of teams of COMPSCI 320 students; interfacing with the other COMPSCI 529 students managing other teams in the course; interfacing with the course instructor, course TA, and course customer. COMPSCI 529 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 529 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 529 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 529 are also expected to be arranged by mutual agreement. An additional one hour weekly meeting of all of the students in COMPSCI 529 is required. Enrollment in this course is only by permission of the instructor, and is restricted to students who have previously taken COMPSCI 320, and received a grade of A or A-. 3 credits.

COMPSCI 532 Systems for Data Science

INSTRUCTOR(S): Peter Klemperer

In this course, students will learn the fundamentals behind large-scale systems in the context of data science. We will cover the issues involved in scaling up (to many processors) and out (to many nodes) parallelism in order to perform fast analyses on large datasets. These include locality and data representation, concurrency, distributed databases and systems, performance analysis and understanding. We will explore the details of existing and emerging data science platforms, including MapReduce-Hadoop, Spark, and more. This course counts as an Elective toward the CS Major. Undergraduate Prerequisites: COMPSCI 311, COMPSCI 345, and COMPSCI 377. 3 credits.

COMPSCI 546 Applied Information Retrieval

INSTRUCTOR(S): Hamed Zamani

COMPSCI 546 is a graduate level course intended to cover information retrieval and other information processing activities, from an applied perspective. There will be numerous programming projects and assignments. It provides a richer technical follow on to COMPSCI 446 (Search Engines) for undergraduates interested in a deeper understanding of the technologies. It also provides a strong basis for continuing on with COMPSCI 646 (Information Retrieval) for those graduate students who are interested in a more complete theoretical coverage of the area. Topics will include: search engine construction (document acquisition, processing, indexing, and querying); learning to rank; information retrieval system performance evaluation; classification and clustering; other machine learning information processing tasks (e.g. basic deep learning models for information retrieval); and many more. This course counts as an Elective toward the CS Major. Undergraduate Prerequisites: COMPSCI 320 and either COMPSCI 383, COMPSCI 446, or COMPSCI 585. 3 credits.

COMPSCI 550 Introduction to Simulation

INSTRUCTOR(S): Peter Haas

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

COMPSCI 561 System Defense and Test

INSTRUCTOR(S): Parviz Kermani

This class trains students to detect and analyze weaknesses and vulnerabilities in target systems as a method of assessing the security of a system. We focus on tools and techniques that an attacker would employ but from the perspective of an ethical system administrator. Topics include tools and techniques for penetration testing and attacks, information gathering, social engineering, and defenses. Specific topics include malware, denial of service attacks, SQL injection, buffer overflow, session hijacking, and system hacking, network sniffing and scans, wireless encryption weaknesses and other WiFi issues, IDS/firewall evasion, metasploit tools, physical security, and setting up honeypots. Was INFOSEC 690S. This course counts as an Elective toward the CS Major. Undergraduate Prerequisites: COMPSCI 460 (or COMPSCI 597N or COMPSCI 660) and COMPSCI 453. 3 credits.

COMPSCI 571 Data Visualization and Exploration

INSTRUCTOR(S): Ali Sarvghad Batn Moghaddam

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

COMPSCI 575 Combinatorics and Graph Theory

INSTRUCTOR(S): Alejandro Henry Morales Borrero

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. Undergraduate Prerequisites: mathematical maturity; calculus; linear algebra; strong performance in some discrete mathematics class, such as COMPSCI 250 or MATH 455. Modern Algebra - MATH 411 - is helpful but not required. 3 credits.

COMPSCI 589 Machine Learning

INSTRUCTOR(S): Bruno Castro da Silva

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

COMPSCI 590J Cyber Effects: Reverse Engineering, Exploit Analysis, and Capability Development

INSTRUCTOR(S): Seth Landsman, Nicholas Merlino, Daniel Walters, Edward Walters, Adam Woodbury

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

COMPSCI 590K Advanced Digital Forensic Systems

INSTRUCTOR(S): Marc Liberatore

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

COMPSCI 590L Making Predictions

INSTRUCTOR(S): Justin Domke

How can we make predictions? The traditional approach in computer science is machine learning. However, this question is addressed in many ways in different fields. One approach is to simply "guess", in which case cognitive biases are important. Another approach might be to identify people who are good at predicting. But do such people exist? And how can we combine their judgment. Economics suggests prediction markets, where people compete for financial reward. This course will cover many different methods for making predictions. The goal is to understand the strengths and weaknesses of each. 3 credits.

COMPSCI 590W Health Informatics and Data Science

INSTRUCTOR(S): Sunghoon Lee, Anthony Nunes

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 (BA or BS). Undergraduate Prerequisite: COMPSCI 240 or STATISTCS 515. 3 credits.

COMPSCI 591NR Seminar - Neural Networks: from Neuroscience to the Forefront of AI

INSTRUCTOR(S): Hava Siegelmann

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

COMPSCI 603 Robotics

INSTRUCTOR(S): Roderic Grupen

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

COMPSCI 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. Text: Engineering a Compiler, Cooper and Torczon. 3 credits.

COMPSCI 611 Advanced Algorithms

INSTRUCTOR(S): Ramesh Sitaraman

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 630 Systems

INSTRUCTOR(S): Emery Berger

This class is an in-depth introduction to systems, focusing on principles of system design that cross-cut numerous systems artifacts, including operating systems, databases, runtime systems, and architecture. We will cover all levels of the "system stack", from chips to distributed systems. This class may be used to satisfy systems core requirements. 3 credits.

COMPSCI 645 Database Design and Implementation

INSTRUCTOR(S): Yanlei Diao

This course covers the design and implementation of traditional relational database systems as well as advanced data management systems. The course will treat fundamental principles of databases such as the relational model, conceptual design, and schema refinement. We will also cover core database implementation issues including storage and indexing, query processing and optimization, and transaction management. Additionally, we will address challenges in modern networked information systems, including data mining, provenance, data stream management, and probabilistic databases. 3 credits.

COMPSCI 654 Advanced Wireless Networking and Sensing in IoT

INSTRUCTOR(S): Jie Xiong

Nowadays, wireless technologies (such as 802.11 Wi-Fi) do not only provide data service but also cater to diverse applications including indoor localization, user authentication, contactless activity sensing, vital sign monitoring, gesture recognition, sleep sensing, wireless charging, etc. This course introduces the students with fundamentals in wireless networking and also the state-of-the-art sensing applications in the Era of Internet-Of-Things. Wireless sensing is also a cutting edge research area which involves wireless communication, signal processing, human computer interaction, machine learning and also hardware prototyping. 3 credits.

COMPSCI 677 Distributed and Operating Systems

INSTRUCTOR(S): Prashant Shenoy

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

COMPSCI 682 Neural Networks: A Modern Introduction

INSTRUCTOR(S): Erik Learned-Miller

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 685 Advanced Natural Language Processing

INSTRUCTOR(S): Mohit Iyer

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

COMPSCI 688 Probabilistic Graphical Models

INSTRUCTOR(S): Daniel Sheldon

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

COMPSCI 690A Advanced Methods in HCI

INSTRUCTOR(S): Narges Mahyar

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

COMPSCI 690OP Optimization in Computer Science

INSTRUCTOR(S): Madalina Fiterau Brostean

Much recent work in computer science in a variety of areas, from game theory to machine learning and sensor networks, exploits sophisticated methods of optimization. This course is intended to give students an in-depth background in both the foundations as well as some recent trends in the theory and practice of optimization for computer science. There is currently no course in the department that covers these topics, and yet it is critical to a large number of research projects done within the department. 3 credits.

COMPSCI 690RA Randomized Algorithms and Probabilistic Data Analysis

INSTRUCTOR(S): Cameron Musco

Randomness has proven itself to be a useful resource for developing provably efficient algorithms and protocols for large scale data processing. As a result, the study of randomized algorithms has become a major research topic in recent years. This course will explore a collection of techniques for effectively using randomization and for analyzing randomized algorithms, as well as examples from a variety of settings and problem areas. The course is a natural follow on both COMPSCI 514: Algorithms for Data Science and COMPSCI 611: Advanced Algorithms. 3 credits.

COMPSCI 691NR Seminar - Neural Networks: from Neuroscience to the Forefront of AI

INSTRUCTOR(S): Hava Siegelmann

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

COMPSCI 691O Seminar - Tools for Explanatory and Tutoring Systems

INSTRUCTOR(S): Beverly Woolf

Artificial Intelligence will radically change education. Through machine learning, data mining, analytics, robotics, and user models, AI will replace false learning boundaries (e.g., learning places, time, level of study); personalize learning; make learning instantly available to everyone; connect learners with partners; provide multi-media; and augment human learning ability. This seminar examines recent work in explanatory and tutoring systems, presents theories about digital teaching and learning, and describes how to deliver personalized teaching in online systems. Such software supports people working alone or in collaborative inquiry to rapidly access and integrate global information. This course describes how to build tutors, stimulates awareness of research issues, and promotes sound analytic and design skills. Specific topics include systems that support collaboration, inquiry, natural language dialogue, authoring tools and user models. The course is appropriate for students from many disciplines (e.g., computer science, linguistics, education, and psychology), researchers, and practitioners from academia, industry, and government. No programming is required. Students will read and critique papers about AI tools (e.g., vision, natural language), methods, and will study the complexity of human learning through advances in cognitive science. Weekly assignments invite students to critique the literature and a final project requires a detailed specification (not a program) for a tutor on your chosen topic. Students present readings from the research literature and several working systems will be available for hands-on or virtual critique. 3 credits.

COMPSCI 692F Seminar - Conversational Artificial Intelligence

INSTRUCTOR(S): Hamed Zamani

Conversational AI is a graduate level seminar intended to cover recent research progress on chat bots, task-oriented dialogue systems, conversational search and recommender systems, conversational question answering, multi-modal conversational systems, and related topics. Students are expected to read, present, and discuss research papers related to conversational AI systems. They will also get familiar with state-of-the-art tools and techniques for developing conversational systems. Section 01=2 credits; Section 02=3 credits.

COMPSCI 692S Seminar - Systems for Machine Learning, Machine Learning for Systems

INSTRUCTOR(S): Marco Serafini

Over the last few years, a wave of excitement about machine learning (ML) and deep learning has proliferated from academia to industry, transforming prototypes in research labs to valid solutions to real-world problems. Using ML entails developing end-to-end pipelines to collect data, clean it, and run learning and inference algorithms in a scalable manner. This results in computationally intense workloads and complex software pipelines. Systems for ML help users organize their data and scale these computationally intense problems to larger and larger datasets. At the same time, ML is having an increasing impact on systems design. Fine-tuned analytical heuristics and cost models are being replaced by learned models, following trends observed in other fields. This seminar will review cutting-edge research on these topics and allow students to work on a hands-on project. It will focus on reading, presenting, and discussing recent papers in the domains of ML for systems and systems for ML (1 credit) and a final project focusing on a specific ML system topic (3 credits). 1-3 credits.

COMPSCI 697M Special Topics - Fixing Social Media

INSTRUCTOR(S): Ethan Zuckerman

Over the past decade, user-generated participatory media – social media – has emerged as the dominant model for content of the Internet. From Facebook to Twitter, YouTube to Wikipedia, content created by non-professionals and circulated for commercial and non-commercial motives underpins seven of the top 10 websites in the US, and has become an increasingly important component of the news ecosystem. While social media was initially hailed as a powerful tool for broadening civic participation, many problems have emerged with the rise of the medium, from questions of whether social media usage is bad for our individual mental health, to whether the fabric of our democracy is being damaged by disinformation, fragmentation and hyperpolarization. As legislators look to regulate these platforms and commentators propose shutting them down entirely, this course looks for an alternative: affirmative visions of social media that are good for individuals and society, which we could work towards building. This class examines possible problems with existing modes of social media, discusses ways in which social media could be a benefit to individuals and societies, develops case studies of successful and healthy online communities, and ultimately designs and builds tools to improve existing social media systems or replace them with novel models. Students will write reflectively about weekly readings and discussions and participate in multi-week projects, ultimately building teams to work on final projects. Meets with COMM 697M and SPP 697M. 3 credits.

COMPSCI 701 Advanced Topics in Computer Science

INSTRUCTOR(S):

Advanced Topics in Computer Science Master's Project: Advanced research project in Computer Science. The 3 credit option is for the second semester of a two semester sequence, 701Y followed by 701. The 6 credit option is for a project that will be begun and completed within the same semester. 6 credits.

COMPSCI 701Y Advanced Topics in Computer Science (1st Semester)

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 Teaching Assistants as Tomorrow's Faculty

INSTRUCTOR(S): Ivon Arroyo

Teaching Assistants as Tomorrow's Faculty prepares Teaching Assistants (TAs) at the College of Information and Computer Sciences to fulfill their duties in an effective and pedagogically sound manner. The one credit (not repeatable) course is semester long and taken by all TAs prior to assuming assistantship. 1 credit.

COMPSCI 891M Seminar - 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 is a one-credit seminar which may be taken repeatedly for credit up to six times. 1 credit.

COMPSCI H311 Honors Colloquium for Introduction to Algorithms

INSTRUCTOR(S): Marius Minea

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

COMPSCI H589 Honors Colloquium for Machine Learning

INSTRUCTOR(S): Bruno Castro da Silva

This colloquium will enrich the primary course by focusing on reading, presenting, and discussing foundational and recent research papers from the machine learning literature. Students will write weekly reading responses, and lead one to two group discussions over the course of the semester. Prerequisites: Students must be enrolled in or have completed COMPSCI 589. 1 credit.

INFO 101 Introduction to Informatics

INSTRUCTOR(S): Cheryl Swanier

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

INFO 150 A Mathematical Foundation 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 190S Introduction to Programming for Information Science

INSTRUCTOR(S): Ghazaleh Parvini

The world is rich in data and our ability to process, manipulate, analyze, and visualize that data is growing in power. This course discusses how to understand and solve real-world problems using modern programming techniques. Concepts and techniques covered include the representation of data, fundamentals of programming including data types, expressions, objects, methods, program design, program testing and debugging, state representation, accessing, processing, and manipulating data, generating visualizations of data, interpreting results, and introductory data analytics using data to make predictions that help solve important problems in the world around us. Use of a laptop computer is required. GenEd R2. Prerequisite: R1 (or a score of 20 or higher on the math placement test Part A), or one of the following courses: MATH 101&102 or MATH 104 or MATH 127 or MATH 128 or MATH 131 or MATH 132. 4 credits.

INFO 190T Introduction to Data Structures in Python

INSTRUCTOR(S): Matthew Rattigan

This course introduces foundational abstract data types and algorithms. The main focus is on the use of data structures in designing and developing programs to solve problems in a variety of domains. Specific topics include lists, sets, dictionaries, graphs, stacks and queues, searching, and sorting. There will be weekly programming assignments, programming and written exercises in discussion sections, regular quizzes, and a cumulative final exam. Use of a personal laptop in class is required. Prerequisites: INFO 190S. 4 credits.

INFO 203 A Networked World

INSTRUCTOR(S): James Kurose

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 majors-level treatment of this material should see COMPSCI 453. 3 credits.

INFO 248 Introduction 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. Course was previously INFO 397F. Prerequisites: COMPSCI 121 and PSYCH 240 (or OIM 240, or STAT 240, or STAT 515, or RES ECON 212, or SOCIOL 212). 4 credits.

INFO 490C Introduction to Social and Cultural Analytics

INSTRUCTOR(S): Laure Thompson

This course introduces the use of computational methods for studying culture and society. Through this course, students will learn how to transform textual and visual material into data; how to explore and analyze said data using machine learning and statistical models as both quantitative and qualitative tools; and how these computational methods are applied within the humanities and social sciences. This course counts as an Elective toward the INFORM major. Prerequisite: INFO 248. 3 credits.

INFO 690C Introduction to Social and Cultural Analytics

INSTRUCTOR(S): Laure Thompson

This course introduces the use of computational methods for studying culture and society. We will learn how to transform textual and visual material into data, and how we can explore and analyze this data using machine learning and statistical models as both quantitative and qualitative tools. We will also survey and discuss how these methods are applied within the humanities and social sciences, as well as reflect on the challenges, limitations, and ethical issues that arise in the computational study of culture and society. 3 credits.