### 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 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. 1 credit.

### CICS 305 Social Issues in Computing

INSTRUCTOR(S): Michelle Trim, Justin Obara, Siobhan Mei

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

### COMPSCI 105 Computer Literacy

INSTRUCTOR(S): William Verts

Microcomputers are used widely in all areas of modern life. For this reason it is important for all students to understand how computers work and how computers can be used as a problem-solving tool. The focus of this course is on computer applications. The course stresses the ways in which computers can help you solve problems efficiently and effectively. The course provides a broad introduction to hardware, software, and mathematical aspects of computers. Four application areas are discussed: Internet tools (including Web page design), word processing, spreadsheets, and databases. Weekly lab assignments are an integral part of the course, and it is expected that students have access to their own computing equipment. This course is a "Foundations" course for the Information Technology minor. Students who are more interested in computer programming should take a course such as COMPSCI 119 or COMPSCI 121. Prerequisites: reasonable high school math skills. Typing ability is also an important asset for the course. Some previous computer experience, while not absolutely required, will prove helpful. Not for CS majors. 3 credits.

### COMPSCI 119 Introduction to Programming

INSTRUCTOR(S): William Verts

This introduction to computer programming emphasizes problem solving using the Python language. Students will explore basic concepts in computer science and computer programming by writing Python 3 programs to analyze text, create files of graphics commands viewable in a Web browser, and many other tasks. No prior programming experience is needed. Not for CS majors. 3 credits.

### COMPSCI 121 Introduction to Problem Solving with Computers

INSTRUCTOR(S): Gordon Anderson, Neena Thota

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 object-oriented metaphor that underlies Java. Concepts and techniques covered include data types, expressions, objects, methods, top-down program design, program testing and debugging, state representation, interactive programs, data abstraction, conditionals, iteration, interfaces, inheritance, polymorphism, arrays, graphics, and GUIs. 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. 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), exploring the frequency domain and holography. Basic information theory, use and limitations of file compression and encryption. Structured databases and how to use them. Information retrieval in heterogenous environments such as the Web. XML as a language for defining new formats for representing data. 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): Jaime Dávila

In this course we will both study commonly used data structures, and develop programming skils that go further than what we might see in an introductory programming course. Our programming language of choice will be Java, but the concepts and techniques we will see apply equally to programming in any object oriented or imperative language. As an application domain, we will be writing programs to solve real life problems. In terms of data structures, we will see lists, stacks, queues, trees, and graphs, among others, as well as techniques for sorting and searching. Students should expect to have programming assignments close to weekly, as well as two midterm exams and a final exam. Please note that this course is not a substitute for COMPSCI 187. If unsure of whether this course or COMPSCI 187 is more appropriate, please contact me at jaimedavila@cs.umass.edu. 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): Matthew Rattigan, Meng-Chieh Chiu

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 197C Special Topics - Programming in C

INSTRUCTOR(S): Run Zhu

A brief introduction to the C programming language for students with a good working knowledge of Java and data structures. This course is good preparation for COMPSCI 230 and courses that use C and C++. Prerequisites: COMPSCI 186 or 187. Runs for 6 weeks. This course is for CS minors and majors only, but it does not count towards either degree. 1 credit.

### COMPSCI 197U Special Topics - A Hands-on Introduction to UNIX

INSTRUCTOR(S): Arun Dunna

This course offers a 6-week introduction to working with Unix, and it is intended to help students work with tools commonly used in CS courses. The class is comprised of both discussion and hands-on exercises in the EdLab. Topics covered include working with the command line, installing and maintaining the OS and software packages, version control systems, compiling programs, and more. No previous experience with Unix is required. This course is for CS minors and majors only, but it does not count towards either degree. 1 credit.

### COMPSCI 198J Practicum - High School Teaching Practicum

INSTRUCTOR(S): Ramesh Sitaraman

Assisting high school students in learning computer programming in the context of a high school course. The language of instruction will be Java or Python. Students taking this independent study will be expected to work under the direct supervision of the high school computer science teacher at a local high school. In some terms students are expected to attend high school classes weekly for approximately one hour. In others they will work on developing projects for future courses, meeting with the high school instructor once a week. Course evaluation will be performed by the high school teacher in cooperation with the supervising faculty member, and a final written report will be required. 1 credit.

### COMPSCI 220 Programming Methodology

INSTRUCTOR(S): Marius Minea, Jaime Dávila

Development of individual skills necessary for designing, implementing, testing and modifying larger programs, including: use of integrated design environments, design strategies and patterns, testing, working with large code bases and libraries, code refactoring, and use of debuggers and tools for version control. There will be significant programming and a mid-term and final examination. 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 language, and moves up the stack from there to the features of modern architectures, assembly languages, and operating system services such as I/O and synchronization. Prerequisites: 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 or consent of instructor. 4 credits.

### COMPSCI 250 Introduction to Computation

INSTRUCTOR(S): David Barrington, Hia Ghosh

Lecture, discussion. 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: MATH 132 and COMPSCI 187. 4 credits.

### COMPSCI 311 Introduction to Algorithms

INSTRUCTOR(S): David Barrington, Marius Minea

This course will introduce you to algorithms in a variety of areas of interest, such as sorting, searching, string-processing, and graph algorithms. 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. There will be one or more programming assignments as well to help you relate the empirical performance of an algorithm to theoretical predictions. Mathematical experience (as provided by COMPSCI 250) is required. You should also be able to program in Java, C, or some other closely related language. Prerequisite: COMPSCI 187 and either COMPSCI 250 or MATH 455. 4 credits.

### COMPSCI 320 Introduction to Software Engineering

INSTRUCTOR(S): David Fisher, Neena Thota

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. Prerequisite: COMPSCI 220. 4 credits.

### COMPSCI 326 Web Programming

INSTRUCTOR(S): Emery Berger

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. Prerequisites: COMPSCI 220 (OR COMPSCI 230), OR INFO 248 AND COMPSCI 186 (OR COMPSCI 187). Previous background in JavaScript is strongly recommended. 4 credits.

### COMPSCI 328 Mobile Health Sensing and 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. Prerequisite: COMPSCI 187 (OR INFO 248 AND COMPSCI 186). 3 credits.

### COMPSCI 333 Internet of Things

INSTRUCTOR(S): Sunghoon Lee

We are living in a world where everyday objects, such as smartphones, cars, TVs, and even refrigerators, are becoming smarter and constantly connected to each other to build, operate, and manage the physical world. This emerging paradigm, namely the Internet of Things (IoT), has great potential to impact how individuals live and work by providing a source of innovative decision making. The design of the IoT, which is defined as "a internetwork of physical items – each embedded with sensors – that are connected to the Internet", requires the understanding of embedded electronics, software, sensors, network, and data analytics. To prepare our students as forerunners of this future, this course will introduce a wide range of topics in the broad areas of IoT, and provide hands-on experiences via a series of exciting projects. Prerequisite: COMPSCI 187. This course counts as a CS Elective for the CS Major (BA/BS). 3 credits.

### COMPSCI 345 Practice and Applications of Data Management

INSTRUCTOR(S): Alexandra Meliou

Computing has become data-driven, and databases are now at the heart of commercial applications. The purpose of this course is to provide a comprehensive introduction to the use of data management systems within the context of various applications. Some of the covered topics include application-driven database design, schema refinement, implementation of basic transactions, data on the web, and data visualization. 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 COMPSCI major (BA/BS). 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. Prerequisites: COMPSCI 187, COMPSCI 240, and COMPSCI 250 (or MATH 455). 3 credits.

### COMPSCI 365 Digital Forensics

INSTRUCTOR(S): Arun Dunna

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 will cover the acquisition, analysis, and courtroom presentation of information from file systems, operating systems, networks, cell phones, and the like. Students do not need experience with these systems. We will review the use of some professional tools that automate data harvesting, however, the primary goal of the class is to understand why and from where artifacts are recoverable in these systems. Several assignments involve coding forensic tools from scratch. For a small portion of the class, we will 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. Prerequisites: COMPSCI 230. 3 credits.

### COMPSCI 373 Introduction to Computer Graphics

INSTRUCTOR(S): Rui Wang

This course introduces the fundamental concepts of 2D and 3D computer graphics. It covers the basic methods for modeling, rendering, and imaging. Topics include: image processing, 2D/3D modeling, 3D graphics pipeline, WebGL, shading, texture mapping, ray tracing, 3D printing. Throughout the class, we will teach students to learn modern graphics techniques, to model the visual world algorithmically, and to implement algorithms using JavaScript. Students who have taken COMPSCI 473 are not eligible to take this course. Students cannot take COMPSCI 497C after taking this course. 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. 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. Prerequisites: COMPSCI 220 (or COMPSCI 230) and COMPSCI 240 (or STAT 515). 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). 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, applications, network, programming interfaces, transport, congestion, routing, data link protocols, local area networks, emerging high-speed 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. Prerequisites: 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. This course counts as a CS Elective toward the CS major (BA/BS), as well an Any 2 menu choice for the Security & Privacy track. Prerequisites: COMPSCI 326, COMPSCI 345, COMPSCI 377, COMPSCI 453, OR COMPSCI 497P. 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. Prerequisites: COMPSCI 311. 3 credits.

### COMPSCI 490S 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 toward the CS major (BA or BS). Prerequisites: COMPSCi 320 (or COMPSCI 326). 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 a CS Elective toward the CS major (BA/BS). Prerequisite: COMPSCI 453. 3 credits.

### COMPSCI 501 Formal Language Theory

INSTRUCTOR(S): Neil Immerman

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. 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 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. Course was previously COMPSCI 590D. Undergraduate Prerequisites: COMPSCI 240 and COMPSCI 311. 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 analysis, formal specification methods, software design, software testing and debugging, program analysis, and automated software engineering. Undergraduate Prerequisites: COMPSCI 320. 3 credits.

### COMPSCI 529 Software Engineering Project Management

INSTRUCTOR(S): David Fisher, Neena Thota

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 535 Computer Architecture

INSTRUCTOR(S): Charles Weems

The structure of digital computers is studied at several levels, from the basic logic level, to the component level, to the system level. Topics include: the design of basic components such as arithmetic units and registers from logic gates; the organization of basic subsystems such as the memory and I/O subsystems; the interplay between hardware and software in a computer system; the von Neumann architecture and its performance enhancements such as cache memory, instruction and data pipelines, coprocessors, and parallelism. Weekly assignments, semester project, 2 hours exams, final. Undergraduate Prerequisites: COMPSCI 335. 3 credits.

### COMPSCI 577 Operating Systems Design and Implementation

INSTRUCTOR(S): Prashant Shenoy

This course will expose students to the internals of an operating system kernel. The course will be based on an Unix-like operating system and will consist of lectures and programming exercises involving implementation of kernel modules, system calls, CPU schedulers, memory management and file systems. Advanced topics such as OS virtualization via containers and distributed scheduling will also be covered. Since operating systems are written in C, proficiency in the C programming language is a must. An undergraduate course on operating systems (equivalent to COMPSCI 377) is also a prerequisite for this course. 3 credits

### COMPSCI 589 Machine Learning

INSTRUCTOR(S): Justin Domke

This course will introduce core machine learning models and algorithms for classification, regression, clustering, and dimensionality reduction. On the theory side, the course will focus on understanding models and the relationships between them. On the applied side, the course will focus on effectively using machine learning methods to solve real-world problems with an emphasis on model selection, regularization, design of experiments, and presentation and interpretation of results. The course will also explore the use of machine learning methods across different computing contexts. Students will complete programming assignments and exams. Python is the required programming language for the course. Undergraduate Prerequisites: COMPSCI 383 and MATH 235. 3 credits.

### COMPSCI 590A 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 a CS Elective toward the CS major (BA or BS). Undergraduate Prerequisites: COMPSCI 460 (or COMPSCI 597N or COMPSCI 660). 3 credits.

### COMPSCI 590G Game Programming

INSTRUCTOR(S): Evangelos Kalogerakis

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

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

INSTRUCTOR(S): Seth Landsman, 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 a CS Elective toward the CS Major (BA or BS). 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 a CS Elective toward the CS Major (BA or BS). 3 credits.

### COMPSCI 590M 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 a CS Elective toward the CS major (BA/BS). Undergraduate Prerequisite: COMPSCI 187 and STAT 515. 3 credits.

### COMPSCI 590U 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 a CS Elective toward the CS major (BA or BS). Undergraduate Prerequisites: COMPSCI 230 and COMPSCI 240. 3 credits.

### COMPSCI 590V 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 a CS Elective toward the CS major (BA or BS). Undergraduate Prerequisites: COMPSCI 220 or 230. No prior knowledge of data visualization or exploration is assumed. 3 credits.

### COMPSCI 591NR Seminar - Neural Networks and Neurodynamics

INSTRUCTOR(S): Robert Kozma

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. This course counts as a CS Elective toward the CS major (BA or BS). 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 a 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 real robot data, as well as simulations. 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 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 661 Secure Distributed Systems

INSTRUCTOR(S): Brian Levine

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". Course was previously COMPSCI 690P. 3 credits.

### COMPSCI 677 Distributed and Operating Systems

INSTRUCTOR(S): Marco Serafini

This course provides an in-depth examination of the principles of distributed systems in general, and distributed operating systems in particular. Covered topics include processes and threads, concurrent programming, distributed interprocess communication, distributed process scheduling, virtualization, distributed file systems, security in distributed systems, distributed middleware and applications such as the web and peer-to-peer systems. Some coverage of operating system principles for multiprocessors will also be included. A brief overview of advanced topics such as multimedia operating systems and mobile computing will be provided, time permitting. Prerequisites: Students should be able to easily program in a high-level language such as C, have had a course on data structures, be familiar with elements of computer architecture and have had previous exposure to the operating system concepts of processes, virtual memory, and scheduling. A previous course on uniprocessor operating systems (e.g., COMPSCI 377) will be helpful but not required. 3 credits.

### COMPSCI 683 Artificial Intelligence

INSTRUCTOR(S): Shlomo Zilberstein

In-depth introduction to Artificial Intelligence focusing on techniques that allow intelligent systems to reason effectively with uncertain information and cope limited computational resources. Topics include: problem-solving using search, heuristic search techniques, constraint satisfaction, local search, abstraction and hierarchical search, resource-bounded search techniques, principles of knowledge representation and reasoning, logical inference, reasoning under uncertainty, belief networks, decision theoretic reasoning, representing and reasoning about preferences, planning under uncertainty using Markov decision processes, multi-agent systems, and computational models of bounded rationality. 3 credits.

### COMPSCI 685 Advanced Natural Language Processing

INSTRUCTOR(S): Brendan O'Connor

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): Justin Domke

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 690J Advanced Cryptography

INSTRUCTOR(S): Adam O'Neill

This is a graduate-level course in advanced cryptographic primitives and protocols, with an eye toward their far-reaching applications in secure messaging, surveillance prevention, cloud computing, and privacy-preserving machine learning, among others. Topics may include: functional encryption, homomorphic encryption, multiparty computation, identification protocols, zero-knowledge proofs, authenticated key exchange, key ratcheting. Specific topics and schedule to be determined by student preferences. This course is a natural 'second course' in cryptography, but we will review core concepts in the beginning to make it accessible to motivated students who have not studied cryptography before. 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): Andrew McGregor

Randomness has proven itself to be a useful resource for developing provably efficient algorithms and protocols. 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. 3 credits.

### COMPSCI 690W 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 691NR Seminar - Neural Networks and Neurodynamics

INSTRUCTOR(S): Robert Kozma

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 692M Seminar - Computing for the Common Good

INSTRUCTOR(S): Narges Mahyar

Computing for the common good is a seminar course that explores new ways of utilizing technology for connecting people and empowering communities to address complex social problems related to health, and democracy. In this course, students will read and discuss the sate of the art papers in digital civics and social technology. Students will discuss and present papers, participate in group discussions, and carry out research projects. Topics include human-computer interaction, digital civics, social computing, collective intelligence, and community sourcing. 1 credit.

### COMPSCI 692Q Seminar - Distributed Quantum Information Systems

INSTRUCTOR(S): Donald Towsley

This course will cover topics in quantum information systems including quantum communications, quantum networking, distributed quantum key distribution, and distributed quantum computing. Students will be expected to present papers on these topics and to participate in classroom discussion. Although no project is required, one of the objectives of this seminar is to identify MS and PhD level research problems in the area of quantum information systems. 1 credit.

### COMPSCI 692R Seminar - Machine Learning in the Real World: Case Studies

INSTRUCTOR(S): Edward Rietman

The course will explore several areas of applications including: aircraft maintenance, semiconductor chip manufacturing, hospice data, spectroscopy data, genomic signal processing, among others. The requirements for the class are skill in Python scripting, Python coding, machine learning, hyper-paramater tuning. The course will have at least three projects; and some guest speakers. 3 credits.

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

INSTRUCTOR(S): Marco Serafini

Machine learning is employed in an increasingly wide range of applications. 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 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. This course will primarily involve reading, presenting, and discussing papers (for 1 credit), and a final project building a machine learning pipeline (for 3 credits). 1-3 credits.

### COMPSCI 692T Seminar - Distributed Machine Learning and Data Mining

INSTRUCTOR(S): Peter Haas, Donald Towsley

In this seminar, students will explore state-of-the-art techniques for finding patterns and functional relationships in data that is split across multiple locations. Topics include factorized learning for vertically partitioned data, model fitting via distributed stochastic gradient descent, and more. Students will read and present papers from a list provided by the instructors. 1-3 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 791U Seminar - Advanced Topics in Information Retrieval

INSTRUCTOR(S): Razieh Rahimi

A seminar in which students will read, present, and discuss research papers on recent and advanced topics in Information Retrieval. Students are expected to lead one or more discussions throughout the semester. This seminar covers recent research in Information Retrieval including neural IR, explainability and interpretability, conversational search, product and multimedia recommendation, and other research that is published within the past few years. Participants will be expected to read up to two papers per week. One or more attendees will be responsible for presenting a 10-15 minute summary of each paper and help lead discussion of the papers. When possible, researchers will be asked to relate their work to the papers. A prerequisite for this seminar is COMPSCI 646, the Information Retrieval course, or its equivalent. Contact Razieh Rahimi (rahimi@cs.umass.edu) if you are uncertain whether you have the necessary background. A project option is available for 3 credits with permission of instructor. 1 or 3 credits.

### COMPSCI 891M Seminar - Theory of Computation

INSTRUCTOR(S): Andrew McGregor

The theory seminar is a weekly meeting in which topics of interest in the theory of computation - broadly construed - are presented. This is sometimes new research by visitors or local people. It is sometimes work in progress, and it is sometimes recent material of others that some of us present in order to learn and share. This is a one-credit seminar which may be taken repeatedly for credit up to six times. 1 credit.

### COMPSCI H250 Honors Colloquium for Introduction to Computation

INSTRUCTOR(S): David Barrington

We will have weekly readings from Godel, Esher, Bach: An Eternal Golden Braid by Douglas Hofstadter.This book contains mathematical problems related to the main course material, and presents some of the same topics as well as others. Students will report on their reading in the seminar, and we will discuss connections between the book and the CS 250 material. Each student will make an oral presentation on a topic of their choice at the end of the term. 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. 1 credit.

### COMPSCI H320 Honors Colloquium for Software Engineering

INSTRUCTOR(S): David Fisher

The purpose of this course is to provide students with supplementary material and insights about the software development enterprise. Students meet once a week for a one-hour discussion of software engineering topics whose exploration is intended to provide depth and perspective on the regular material of COMPSCI 320. Topics may be suggested by current events or by problems that may arise in the course of the 320 semester. Students will be required to write a term paper as part of the requirements for this course. 1 credit.

### COMPSCI H446 Honors Colloquium for Search Engines

INSTRUCTOR(S): David Fisher

This course is an honors colloquium for COMPSCI 446. Students will explore and discuss topics from the 446 curriculum in greater detail. Students will implement an expanded programming project, such as implementing the Inference Network retrieval model, and produce a final report on that project. 1 credit.

### COMPSCI H589 Honors Colloquium for Machine Learning

INSTRUCTOR(S): Benjamin Marlin

This colloquium will enrich the primary course by focusing on reading, presenting, and discussing foundational and recent research papers from the machine learning literature. Students will write weekly reading responses, and lead one to two group discussions over the course of the semester. 1 credit.

### INFO 197P Special Topics - Programming in Python for Data Science

INSTRUCTOR(S): Emma Anderson

A brief introduction to the Python programming language for students with a working knowledge of basic programming concepts. Class is comprised of discussion and hands-on activities, as well as two projects completed outside of class. Topics include Python data structures, data analysis tools, and an overview of machine learning libraries. Intended for Informatics majors. 1 credit.

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

### INFO 248 Introduction to Data Science

INSTRUCTOR(S): Gordon Anderson

The terms "data science" and "big data" appear in the news media and in everyday conversations. Moreover, we are told that we live in the "age of information", where almost every business venture and scientific research initiative collect a massive amount of data which may contain valuable information. 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 discussed. Students will work with data from the financial, epidemiological, educational, and other domains. The course provides many case studies and examples of real-world data that students work with using various tools including the R programming language as well as the structured query language (SQL). This course consists of two meetings per week. Each meeting includes a lecture, where conceptual material will be presented, followed by lab time where students receive instruction on the use of software tools and apply the concepts by working on data sets. Readings will be assigned as preparation for each class meeting. A project will be assigned during the course. The project provides students with an opportunity to explore the topics in more depth in a specialized domain. Two midterm and final exams will be given. Grades are determined by a combination of class participation including the in-class lab activities, projects, and exam scores. Software: The R software for statistical analysis (www.r-project.org). 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) both with a C or better. 4 credits.