# College of Information and Computer Sciences
# UMassAmherst

**Course Descriptions**
**Fall 2016**

**COMPSCI** (unless otherwise noted)

### 105 Computer Literacy (R2) 3 credits

INSTRUCTOR(S): 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. There are optional lab times set up for students who do not have the proper equipment or software available to them. 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.

### 119 Introduction to Programming 3 credits

INSTRUCTOR(S): Verts

This introduction to computer programming with Python emphasizes multimedia (graphics and sound) applications that are relevant for Web designers, graphic artists, and anyone who just wants to have more fun with their computer. Students will explore basic concepts in computer science and computer programming by manipulating digital images and sound files. No prior programming experience is needed. Not for CS majors. 3 credits.

### 120 Introduction to Problem Solving with the Internet (R2) 3 credits

INSTRUCTOR(S): Verts

The Internet is a goldmine of information and software resources for those who know how to plug in and navigate it. Originally designed by computer scientists for computer scientists, the net is now a driving force behind life in the information age and a new global economy. This course will provide non-CS majors with timely skills needed to tap the net as well as an introduction to basic networking, client-side web programming in HTML, CSS, and Javascript, and server-side programming in Python. In addition to static and dynamic web page and web site design and implementation, we will cover strategies for finding information, managing e-mail, and ensuring privacy. We will survey current social, technical, and political topics that are relevant to the Internet such as spam and malware, net neutrality, censorship, copyright laws, and public key cryptography. Prerequisites: some hands-on experience with PCs or MACs or UNIX (programming experience is NOT required). Not for CS majors. 3 credits.

### 121 Introduction to Problem Solving with Computers (R2) 4 credits

INSTRUCTOR(S): Anderson; Thota

COMPSCI 121 provides an introduction to problem solving and computer programming using the programming language Java; it also provides an integrated introduction to some of the wonderful innovations to modern science and indeed modern life that can be attributed to computer science. 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, arrays, graphics, and GUIs. No previous programming experience required. A companion introduction to programming class, COMPSCI 119 is also offered. If you are fairly sure you only want to do just one programming class, take that course; if you think it likely that you will do more than one programming course, take 121. Use of computer is required. Prerequisite: R1. 4 credits.

### 187 Programming with Data Structures (R2) 4 credits

INSTRUCTOR(S): Corner; Thota

This 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. There will be weekly assignments and assignments in discussion sections consisting of programming and written exercises. There will also be several exams. Prerequisites: COMPSCI 121 (or equivalent Java experience). Basic Math Skills (R1). Advising information about the prerequisite: for Fall 2016, students with grades of B-, C+, or C in 121 are strongly encouraged to take 190D, Using Data Structures, in preparation for COMPSCI 187. Beginning with the Spring 2017 offering, the prerequisite for 187 will change to a grade of B or better in 121 (or C or better in 190D). 4 credits.

## COMPSCI (unless otherwise noted)

### 187HH Programming with Data Structures with Honors Colloquium (R2) 5 credits

INSTRUCTOR(S): Corner; Thota

This 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. There will be weekly assignments and assignments in discussion sections consisting of programming and written exercises. There will also be several exams. Prerequisites: COMPSCI 121 (or equivalent Java experience). Basic Math Skills (R1). Advising information about the prerequisite: for Fall 2016, students with grades of B-, C+, or C in 121 are strongly encouraged to take 190D, Using Data Structures, in preparation for COMPSCI 187. Beginning with the Spring 2017 offering, the prerequisite for 187 will change to a grade of B or better in 121 (or C or better in 190D). Honors Enrichment: Honors students will be learning about advanced data structures that are not covered in the non-honors course. Students will develop new projects for the course and complete the projects developed by other students. 5 credits.

### 190D Using Data Structures 4 credits

INSTRUCTOR(S): Liberatore

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, maps, graphs, stacks, queues, searching, and sorting. There will be weekly programming assignments, assignments in discussion sections consisting of programming and written exercises, and several exams. 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). 4 credits.

### 197B Special Topics - Advanced Assignments for CMPSCI 121 1 credit

INSTRUCTOR(S): Anderson

This add on to COMPSCI 121 consists entirely of five additional more advanced programs that students write to extend the coding experience available in the base 121 class. The course is intended primarily for students in 121 who have some previous programming experience. Must be enrolled in COMPSCI 121 concurrently. 1 credit.

### 197C Special Topics - Programming in C 1 credit

INSTRUCTOR(S): Scarrci

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 121 and 187. Runs for 6 weeks. This course is for CS minors and majors only, but it does not count towards either degree. 1 credit.

### 197U Special Topics - A Hands-on Introduction to UNIX 1 credit

INSTRUCTOR(S): Foley

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.

### 197WP Special Topics - Introduction to HTML/CSS 1 credit

INSTRUCTOR(S): Scarrci

An academic approach to building websites with html and css. This course is aimed at beginning programmers and self-taught web developers who would like to cement their skills. Topics include structural html5, css, and introductory javascript/jquery (if time permits). Prerequisite: COMPSCI 121. Mandatory P/F. Students who are enrolled in or have taken COMPSCI 326 are not eligible to take this course. 1 credit.

### 220 Programming Methodology 4 credits

INSTRUCTOR(S): Guha

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 or ECE 242. 4 credits.

### 230 Computer Systems Principles 4 credits

INSTRUCTOR(S): Richards

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 or ECE 242. 4 credits.

### 240 Reasoning Under Uncertainty 4 credits

INSTRUCTOR(S): Mazumdar

Development of mathematical reasoning skills for problems that involve uncertainty. Each concept will be illustrated by real-world examples and demonstrated though in-class and homework exercises, some of which will involve programming. Counting and probability -- basic counting problems, probability definitions, mean, variance, binomial distribution, Markov and Chebyshev bounds. Probabilistic reasoning -- conditional probability and odds, Bayes' Law, Naive Bayes classifiers, Monte Carlo simulation. Markov chains, Markov decision processes, classical game theory, introduction to information theory. Prerequisites: COMPSCI 187 (or ECE 242) and MATH 132 or consent of instructor. 4 credits.

### 250 Introduction to Computation 4 credits

INSTRUCTOR(S): Immerman

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, 2-3 midterm exams, timed final. Prerequisite: MATH 132 and COMPSCI 187 (or ECE 242). MATH 132 may be used as a co-requisite with permission of instructor. 4 credits.

### 305 Social Issues in Computing 3 credits

INSTRUCTOR(S): Napoleone; Trim

Using a range of different disciplinary perspectives we will explore various impacts of computers on modern society. This exploration will focus primarily on the social impacts of computers, with an emphasis on ethical concerns. Students will gain practice in several writing genres, including public writing and academic writing. Students will produce approximately 20-25 polished pages via 4-6 larger written projects. Writing experiences will also include writing for electronic environments, extensive practice with peer review, and discussion of job-seeking documents; there will be two presentations. Prerequisite: ENGLWRIT 112. 3 credits

### 311 Introduction to Algorithms 4 credits

INSTRUCTOR(S): McGregor; Krishnamurthy

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 250 or MATH 455. 4 credits.

### 320 Introduction to Software Engineering (IE) 4 credits

INSTRUCTOR(S): Anderson; 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, exams, and a term project. This course satisfies the IE Requirement. Prerequisite: COMPSCI 220. 4 credits.

### 325 Introduction to Human Computer Interaction 3 credits

INSTRUCTOR(S): Hudlicka

In this course we examine the important problems in Usability, Human Computer Interaction, User Interfaces and Human Centered Computing. We will examine elements of HCI history, human information processing capabilities, HCI design, user interface prototyping methods and new applications and directions in HCI, including affective HCI. This is not a course on how to make better dialog boxes, but rather a much more thorough exploration of how humans interact with computers and how to evaluate the effectiveness of HCI designs. Some elementary programming, or the use of UI prototyping tools, will be required, but students without prior programming experience should feel right at home in this class. The course is available to any undergraduate student and is not limited to computer scientists. IT-minor students are especially encouraged to participate. The course emphasizes group projects and students will be required to work in teams. There is one midterm exam, numerous class presentations of on-going project work, and a final demo and presentation of the major class project. No prerequisites. 3 credits.

### 326 Web Programming (IE) 3 credits

INSTRUCTOR(S): Richards

The World Wide Web was proposed originally as a collection of static documents inter-connected by hyperlinks. Today, the web has grown into a rich platform, built on a variety of protocols, standards, and programming languages, that aims to replace many of the services traditionally provided by a desktop operating system. Topics will include: producing dynamic content using a server-based language, content serving databases and XML documents, session state management, multi-tier web-based architectures, web security, and core technologies including HTTP, HTML5, CSS, JavaScript, and SQL will be emphasized. This course will also study concepts and technologies including AJAX, social networking, mashups, JavaScript libraries (e.g., jQuery), and web security. This course is hands-on and project-based; students will construct a substantial dynamic web application based on the concepts, technologies, and techniques presented during lecture. This course satisfies the IE Requirement. Prerequisites: COMPSCI 220 or COMPSCI 230. 3 credits.

### 335 Inside the Box: How Computers Work 3 credits

INSTRUCTOR(S): Weems

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

### 335HH Inside the Box: How Computers Work with Honors Colloquium 4 credits

INSTRUCTOR(S): Weems ***Honors Colloq***

How does the computer actually work? In this course we peel away the layers of abstraction and look at how switches become logic circuits, how logic circuits do math, and how programs really execute. We will wire up some simple examples of logic, then move on to programming an embedded ARM processor in a mix of assembly language and C, interfacing with various I/O devices and sensors, to experience what happens when machine code executes. We will also see the impact of hidden acceleration mechanisms like caches, pipelines, and branch predictors. In addition to the regular requirements of COMPSCI 335, honors section students are expected to meet weekly with the instructor. These meetings can be a combination of lecture, student research presentations, project progress reports, discussions, demonstrations of work, and problem solving. Students define their own program of enrichment, which will typically be either a research project or a project to develop an application of embedded systems. Students may work individually or in teams. Grading is based upon participation in the weekly meetings and the quality of the finished project. Prerequisite: COMPSCI 220 (or COMPSCI 230). 4 credits.

### 345 Practice and Applications of Data Management 3 credits

INSTRUCTOR(S): Miklau

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 alternate between lecture and practice, and the students will experience the covered topics through a semester-long collaborative mini-project. This course counts as a CS Elective toward the COMPSCI major (BA/BS). Students who have taken COMPSCI 445 are not eligible to take this course. Prerequisite: COMPSCI 187 (or ECE 242). 3 credits.

### 377 Operating Systems 4 credits

INSTRUCTOR(S): Shenoy

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.

### 383 Artificial Intelligence 3 credits

INSTRUCTOR(S): Zilberstein

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 how these concepts are applied in the context of several applications. Prerequisites: COMPSCI 220 (or COMPSCI 230) and COMPSCI 240. 3 credits.

### 390MB Mobile Health Sensing and Monitoring 3 credits

INSTRUCTOR(S): Ganesan

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

### 391H Software Engineering with Honors 1 credit

INSTRUCTOR(S): Osterweil ***Honors Colloq***

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. Prerequisite of COMPSCI 220 with a grade of B or better. 1 credit.

### 391L Seminar - Computer Crime Law and the Technologies of Investigation and Privacy 3 credits

INSTRUCTOR(S): Liberatore

A study, analysis, and discussion of the legal issues related to crimes involving computers and networks, including topical actions by dissidents and governments. We will also study the technologies of forensic investigation, intelligence gathering, privacy enhancement, and censorship resistance. Our main legal topics will include recent and important case law, statutes, and constitutional clauses concerning authorization, access, search and seizure, wiretaps, the right to privacy, and FISA. Our technology topics will include methods of investigation and resistance in the context of the Internet and Cellular networks. Students are assumed to have no background in legal concepts. Students will be required to complete substantial legal readings, complete significant written analysis of rulings, learn about technologies in detail, and participate in lively class discussion. Prerequisite: COMPSCI 230 and ENGLWRIT 112. This course counts as a CS Elective toward the CS major. 3 credits.

### 403 Introduction to Robotics: Perception, Mechanics, Dynamics, and Control 3 credits

INSTRUCTOR(S): Biswas

This course covers basic methods and concepts in order to explain how robots work. We will study how they sense things in the world, how you make a robot move, and how robots can make their own decisions. We will study mechanisms (kinematics and dynamics), actuators, sensors, (with a focus on machine vision), signal processing, feedback control theory, machine learning, and path planning. Students will build software systems for simulated robots and logged real robot sensor data, and do experiments with system identification and control, image processing, and path planning to reinforce the material presented in class. Prerequisites: Math 235 (or Math 236) and COMPSCI 220 (or COMPSCI 230). 3 credits.

### 410 Compiler Techniques 3 credits

INSTRUCTOR(S): Moss

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. Prerequisites: COMPSCI 250 (or MATH 455) and COMPSCI 377. 3 credits.

### 453 Computer Networks 3 credits

INSTRUCTOR(S): 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.

### 460 Introduction to Computer and Network Security 3 credits

INSTRUCTOR(S): Houmansadr

This course provides an introduction to the principles and practice of computer and network security with a focus on both fundamentals and practical information. The three key topics of this course are cryptography, privacy, and network security. Subtopics include ciphers, key exchange, security services (integrity, availability, confidentiality, etc.), network and web based security attacks, anonymous communications, vulnerabilities, and countermeasures. Students will complete several lab assignments. Grades will be determined by class participation, lab work, homework, quizzes, and exams. Prerequisites include COMPSCI 377 and a familiarity with Unix. 3 credits.

### 474 Image Synthesis 3 credits

INSTRUCTOR(S): Wang

This course provides a broad overview of the theory and practice of image synthesis. Topics include photorealistic rendering, global illumination techniques, Monte Carlo ray tracing, virtual reality, texture synthesis, and image editing. Previous experience with introductory computer graphics (such as COMPSCI 473) is preferred. Prerequisites: COMPSCI 187 or equivalent, MATH 235. 3 credits.

### 491IP Seminar - Programming the iPhone and iPad 3 credits

INSTRUCTOR(S): Weems

The iPad, iPhone, and iPod Touch present examples of a mobile processor with an interesting set of peripheral devices and limitations. They are programmed using Swift, Objective C and a large set of APIs. This seminar will be an introduction to the Swift and Objective C languages and iOS development environment, and will include student programming assignments and development projects targeting the Apple iPhone/iPad simulator. We will spend much of the class time actually working through examples, and discovering how some of the features really work. Students will need to have access to an Intel machine running OSX, because the development environment and simulator work only on that platform. Students also must join the Apple Developer Program, and accept their license agreements in order to obtain and use the necessary software. This course counts as a CS Elective toward the CS major (BA/BS). Prerequisites: COMPSCI 320 or COMPSCI 326. 3 credits.

### 520 Software Engineering: Synthesis and Development 3 credits

INSTRUCTOR(S): Just

Introduces students to the principal activities involved in developing high-quality software systems in a variety of application domains. Topics include: requirements analysis, formal and informal specification methods, process definition, software design, testing, and risk management. The course will pay particular attention to differences in software development approaches in different contexts. Prerequisites: COMPSCI 320 with a grade of 'C' or better. 3 credits.

### 529 Software Engineering Project Management 3 credits

INSTRUCTOR(S): Anderson; 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.

### 575 Combinatorics and Graph Theory [Cross-list with: MATH 513] 3 credits

INSTRUCTOR(S): Barrington

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. 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.

### 585 Introduction to Natural Language Processing 3 credits

INSTRUCTOR(S): O'Connor

Natural Language Processing (NLP) is the engineering art and science of how to teach computers to understand human language. NLP is a type of artificial intelligence technology, and it's now ubiquitous -- NLP lets us talk to our phones, use the web to answer questions, map out discussions in books and social media, and even translate between human languages. Since language is rich, subtle, ambiguous, and very difficult for computers to understand, these systems can sometimes seem like magic -- but these are engineering problems we can tackle with data, math, machine learning, and insights from linguistics. This course will introduce NLP methods and applications including probabilistic language models, machine translation, and parsing algorithms for syntax and the deeper meaning of text. During the course, students will (1) learn and derive mathematical models and algorithms for NLP; (2) become familiar with basic facts about human language that motivate them, and help practitioners know what problems are possible to solve; and (3) complete a series of hands-on projects to implement, experiment with, and improve NLP models, gaining practical skills for natural language systems engineering. Undergraduate Prerequisites: COMPSCI 220 (or COMPSCI 230) and COMPSCI 240. An alternate prerequisite of LINGUIST 492B is acceptable for Linguistics majors. 3 credits.

### 585HH Introduction to Natural Language Processing with Honors Colloquium 4 credits

INSTRUCTOR(S): O'Connor ***Honors Colloq***

Natural Language Processing (NLP) is the engineering art and science of how to teach computers to understand human language. NLP is a type of artificial intelligence technology, and it's now ubiquitous -- NLP lets us talk to our phones, use the web to answer questions, map out discussions in books and social media, and even translate between human languages. Since language is rich, subtle, ambiguous, and very difficult for computers to understand, these systems can sometimes seem like magic -- but these are engineering problems we can tackle with data, math, machine learning, and insights from linguistics. This course will introduce NLP methods and applications including probabilistic language models, machine translation, and parsing algorithms for syntax and the deeper meaning of text. During the course, students will (1) learn and derive mathematical models and algorithms for NLP; (2) become familiar with basic facts about human language that motivate them, and help practitioners know what problems are possible to solve; and (3) complete a series of hands-on projects to implement, experiment with, and improve NLP models, gaining practical skills for natural language systems engineering. Honors Enrichment: Extra readings from research papers and discussion. Students will be responsible for paper presentations. Undergraduate Prerequisites: COMPSCI 220 (or COMPSCI 230) and COMPSCI 240. An alternate prerequisite of LINGUIST 492B is acceptable for Linguistics majors. 4 credits.

### 589 Machine Learning 3 credits

INSTRUCTOR(S): 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 including desktop, cluster, and cloud computing. The course will include programming assignments, a midterm exam, and a final project. Python is the required programming language for the course. Prerequisites: COMPSCI 383 and MATH 235. 3 credits.

### 590B Detecting Interference in Networks 3 credits

INSTRUCTOR(S): Gill

This course will cover topics related to the broad issue of detecting and measuring network interference. This will include topics such as traffic differentiation, network neutrality, surveillance, and blocking/censorship of content. The course will focus on network measurement tools and techniques in the context of detecting and observing manipulations and interference with network traffic. When relevant, historical and ethical/legal context will be discussed in relation to these topics. This course counts as a CS Elective toward the CS major (BS/BA) as well as an Any 2 Menu for BS subplans: SECURPRV, NETWORK, SOFTSYS. Undergraduate prerequisite: COMPSCI 453. 3 credits.

### 590D Algorithms for Data Science 3 credits

INSTRUCTOR(S): Saha

Big Data brings us to interesting times and promises to revolutionize our society from business to government, from healthcare to academia. As we walk through this digitized age of exploded data, there is an increasing demand to develop unified toolkits for data processing and analysis. In this course our main goal is to rigorously study the mathematical foundation of big data processing, develop algorithms and learn how to analyze them. Specific Topics to be covered include: 1) Clustering 2) Estimating Statistical Properties of Data 3) Near Neighbor Search 4) Algorithms over Massive Graphs and Social Networks 5) Learning Algorithms 6) Randomized Algorithms. This course counts as a CS Elective toward the CS major. Undergraduate Prerequisites: COMPSCI 240 or COMPSCI 311. 3 credits.

**COMPSCI** (unless otherwise noted)

### 590N Introduction to Numerical Computing with Python 1 credit

INSTRUCTOR(S): Adams

This course is an introduction to computer programming for numerical computing. The course is based on the computer programming language Python and is suitable for students with no programming or numerical computing background who are interested in taking courses in machine learning, natural language processing, or data science. The course will cover fundamental programming, numerical computing, and numerical linear algebra topics, along with the Python libraries that implement the corresponding data structures and algorithms. The course will include hands-on programming assignments and quizzes. No prior programming experience is required. Familiarity with undergraduate-level probability, statistics and linear algebra is assumed. 1 credit.

### 590S Systems for Data Science 3 credits

INSTRUCTOR(S): Berger

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 map-reduce and graph analytics systems like Hadoop and Apache Spark. Undergraduate Prerequisites: COMPSCI 311, COMPSCI 345, and COMPSCI 377. 3 credits.

### 591CF Seminar - Cybersecurity Lecture Series [Cross-list with: ECE/MATH 591CF/SCH-MGMT 697CB] 1 credit

INSTRUCTOR(S): Levine; Houmansadr

This course is a one-credit seminar on security research across departments at UMass. Each presentation will cover an active research topic at UMass in a way that assumes only a basic background in security. External speakers may also be invited. Note that this course is not intended to be an introduction to cybersecurity, and will not teach the fundamentals of security in a way that would be useful as a foundation for future security coursework. The intended audience is graduate and advanced undergraduate students, as well as faculty. Meets with ECE/MATH 591CF. May be taken for credit up to 2 times. This course does not count toward any requirements for the CS major or minor.

### 591L Seminar - Computer Crime Law and the Technologies of Investigation and Privacy 3 credits

INSTRUCTOR(S): Liberatore

A study, analysis, and discussion of the legal issues related to crimes involving computers and networks, including topical actions by dissidents and governments. We will also study the technologies of forensic investigation, intelligence gathering, privacy enhancement, and censorship resistance. Our main legal topics will include recent and important case law, statutes, and constitutional clauses concerning authorization, access, search and seizure, wiretaps, the right to privacy, and FISA. Our technology topics will include methods of investigation and resistance in the context of the Internet and Cellular networks. Students are assumed to have no background in legal concepts. Students will be required to complete substantial legal readings, complete significant written analysis of rulings, learn about technologies in detail, and participate in lively class discussion. Students who have completed COMPSCI 391L are not eligible for this course. Graduate students will be expected to read selected current law and technical articles and to analyze state-of-the-art systems in the context of law and policy. 3 credits.

### 597A Special Topics - Agile Software Project Management 3 credits

INSTRUCTOR(S): Richards

Students will lead a software development team, in the context of a web development startup seeking venture capital funding, to the successful completion and deployment of a real web application. This course emphasizes agile software engineering principles and practice such as scrum, sprints, iterative design and development, test driven development, and shippable increments. Students enrolled in this course will be assigned a team of approximately 5-7 students enrolled in COMPSCI 326 and will be required to hold team meetings, manage work and design of a unique product, evaluate the individual work and contributions of their team, handle team conflicts and communication, and report weekly on the progress of their startup and its product development. Students enrolled in this course will meet with the instructor to report on their team and discuss agile software development techniques and practice from the literature. Students will also be responsible for organizing and attending weekly team meetings. Counts as a CS Elective for the CS Major (BS/BA). 3 credits.

### 597N Special Topics - Introduction to Computer and Network Security 3 credits

INSTRUCTOR(S): Houmansadr

This course provides an introduction to the principles and practice of computer and network security with a focus on both fundamentals and practical information. The three key topics of this course are cryptography, privacy, and network security. Subtopics include ciphers, key exchange, security services (integrity, availability, confidentiality, etc.), network and web based security attacks, anonymous communications, vulnerabilities, and countermeasures. Students will complete several lab assignments. Grades will be determined by class participation, lab work, homework, quizzes, and exams. Prerequisites include COMPSCI 377 and a familiarity with Unix. 3 credits.

### 597R Special Topics - Applied Information Retrieval 3 credits

INSTRUCTOR(S): Fisher

This course will provide a “flipped classroom” experience, with both online lectures, programming exercises, and other activities as well as a weekly lecture/exploration/discussion section. This 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, as well as short answer homeworks. It provides both 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; and many more. Prerequisites: UNDERGRADUATES: COMPSCI 320 AND COMPSCI 383 (OR COMPSCI 446 OR COMPSCI 585). This course counts as a CS Elective toward the BS/BA. 3 credits.

### 610 Compiler Techniques 3 credits

INSTRUCTOR(S): Moss

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.

### 611 Advanced Algorithms 3 credits

INSTRUCTOR(S): 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.

### 620 Advanced Software Engineering: Synthesis and Development 3 credits

INSTRUCTOR(S): Just

This course examines the varied approaches to the development of computer software. We examine various ideas about how software products should be structured and function. We then examine how software processes serve as vehicles for manufacturing such products. This approach facilitates the direct study of different software development approaches, and a more direct study of their effects on the products they produce. This approach will be used by students, who will examine representative current software development product and process approaches in in-class presentations and written project papers as part of their coursework. 3 credits.

### 631 Programming Languages 3 credits

INSTRUCTOR(S): Guha

This course surveys the principles of modern programming languages by implementation. Topics include building interpreters for several kinds of programming languages, type-inference algorithms, program analyses, and program verification. We will conclude with topics that bridge to compilers and runtime systems, including program transformations and garbage collection. 3 credits.

### 646 Information Retrieval 3 credits

INSTRUCTOR(S): Jiang

The course will cover basic and advanced techniques for text-based information systems. Topics covered include retrieval models, indexing and text representation, browsing and query reformulation, data-intensive computing approaches, evaluation, and issues surrounding implementation. The course will include a substantial project such as the implementation of major elements of search engines and applications. Prerequisite: CMPSCI 446, CMPSCI 445, or equivalent. 3 credits.

### 670 Computer Vision 3 credits

INSTRUCTOR(S): Maji

This course will explore current techniques for the analysis of visual data (primarily color images). In the first part of the course we will examine the physics and geometry of image formation, including the design of cameras and the study of color sensing in the human eye. In each case we will look at the underlying mathematical models for these phenomena. In the second part of the course we will focus on algorithms to extract useful information from images. This includes detection of reliable interest points for applications such as image alignment, stereo and instance recognition; robust representations of images for recognition; and principles for grouping and segmentation. Time permitting we will look at some additional topics at the end of the course. Course assignments will highlight several computer vision tasks and methods. For each task you will construct a basic system, then improve it through a cycle of error analysis and model redesign. There will also be a final project, which will investigate a single topic or application in greater depth. This course assumes a good background in basic probability, linear algebra, and ability to program in MATLAB. Prior experience in signal/image processing is useful but not required. 3 credits.

### 689 Machine Learning 3 credits

INSTRUCTOR(S): Mahadevan

Machine learning is the computational study of artificial systems that can adapt to novel situations, discover patterns from data, and improve performance with practice. This course will cover the popular frameworks for learning, including supervised learning, reinforcement learning, and unsupervised learning. The course will provide a state-of-the-art overview of the field, emphasizing the core statistical foundations. Detailed course topics: overview of different learning frameworks such as supervised learning, reinforcement learning, and unsupervised learning; mathematical foundations of statistical estimation; maximum likelihood and maximum a posteriori (MAP) estimation; missing data and expectation maximization (EM); graphical models including mixture models, hidden-Markov models; logistic regression and generalized linear models; maximum entropy and undirected graphical models; nonparametric models including nearest neighbor methods and kernel-based methods; dimensionality reduction methods (PCA and LDA); computational learning theory and VC-dimension; reinforcement learning; state-of-the-art applications including bioinformatics, information retrieval, robotics, sensor networks and vision. Prerequisites: undergraduate level probability and statistics, linear algebra, calculus, AI; computer programming in some high level language. 3 credits.

### 690B Detecting Interference in Networks 3 credits

INSTRUCTOR(S): Gill

This course will cover topics related to the broad issue of detecting and measuring network interference. This will include topics such as traffic differentiation, network neutrality, surveillance, and blocking/censorship of content. The course will focus on network measurement tools and techniques in the context of detecting and observing manipulations and interference with network traffic. When relevant, historical and ethical/legal context will be discussed in relation to these topics. 3 credits.

### 690IV Intelligent Visual Computing 3 credits

INSTRUCTOR(S): Kalogerakis

The course will teach students algorithms that intelligently process, analyze and generate visual data. The course will start by covering the most commonly used image and shape descriptors. It will proceed with statistical models for representing 2D images, textures, 3D shapes and scenes. The course will then provide an in-depth background on topics of shape and image analysis and co-analysis. Particular emphasis will be given on topics of automatically inferring function from shapes, as well as their contextual relationships with other shapes in scenes and human poses. Finally, the course will cover topics on automating the design and synthesis of 3D shapes with machine learning algorithms and advanced human-computer interfaces. Students will read, present and critique state-of-the-art research papers on the above topics. This course counts as a CS Elective toward the CS major (BA/BS). 3 credits.

### 690PE Performance Evaluation 3 credits

INSTRUCTOR(S): Nain

This course will provide an introduction to the tools and techniques needed to construct and analyze performance models of computer systems, distributed systems, and communication networks. The course covers three topics: i) analytical methods including discrete and continuous time Markov chain models, queues in isolation, queueing networks, and fluid queues; ii) computer/communication system measurement methodology including statistical inference and estimation of pertinent performance metrics, optimal measurement design, and bias removal; iii) applications to solving real world problems including model validation against measurements and/or simulation, case studies will be drawn from the areas of parallel and distributed systems, and networks. The goal is to teach fundamentals with a long half-life. Students are expected to have taken probability theory at at least the undergraduate level. 3 credits.

### 691E Seminar - Interactive Machine Learning: Algorithms and Theory 1-3 credits

INSTRUCTOR(S): Krishnamurthy

Interactive machine learning involves an algorithm or an agent making decisions about data collection, contrasting starkly with traditional learning paradigms. Interactive data collection often enables learning with significantly less data, and it is critical in a number of applications including personalized recommendation, medical diagnosis, and dialogue systems. This seminar will focus on the design and analysis of interactive learning algorithms for settings including active learning, bandits, reinforcement learning, and adaptive sensing. We will cover foundational and contemporary papers, with an emphasis on algorithmic design principles as well as understanding and proving performance guarantees. Students enrolled in the 3 credit version of the course will present one paper in detail to the class as well as prepare notes for one additional lecture. Students enrolled in the 1 credit version of the course will prepare notes for one lecture. Lect 01=3 credits; Lect 02=1 credit.

### 691O Seminar - Tools for Explanatory and Tutoring Systems 3 credits

INSTRUCTOR(S): Woolf

This seminar examines recent work in explanatory and tutoring systems. Participants study *artificial intelligence in education*, a young field that explores theories about learning, and explores how to build software that delivers differential teaching as it adapt its response to student needs and domain knowledge. Such software supports people who work alone or in collaborative inquiry, students who question their own knowledge, and students who rapidly access and integrate global information. This course describes how to build these tutors and how to produce the best possible learning environment, whether for classroom instruction or lifelong learning. The objective of the course is to stimulate awareness of research issues and to promote sound analytic and design skills as they pertain to building knowledge representations and control strategies. Specific topics include collaboration, inquiry, dialogue systems, machine learning, simulators, authoring tools and user models. The course is appropriate for students from many disciplines (computer science, linguistics, education, and psychology), researchers, and practitioners from academia, industry, and government. No programming is required. Students read and critique tools, methods, and ideas, learn how artificial intelligence is applied (e.g., vision, natural language), and 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 about a chosen topic. Students present book chapter material weekly in parallel with readings from the research literature. Several working systems will be available for hands-on critique. 3 credits.

### 691PL Seminar - Advanced Programming Languages 3 credits

INSTRUCTOR(S): Guha

A project-based seminar, where students implement and evaluate ideas from recent programming languages research papers. 3 credits.

### 691SE Seminar - Software Engineering Societal Impact and Ethics 3 credits

INSTRUCTOR(S): Osterweil

The goal of this course is to provide a forum in which students can consider the wide-ranging impact that the work of software engineers has on society. This will be done by having students make presentations about their ongoing case studies of example software engineering systems (both successes and failures), software practices (both good and bad), and applications to domains (both effective and problematic). Thus, for example, it is expected that one student case study will focus on the disastrous healthcare.gov web site rollout in Fall 2014; another will examine the development and deployment of software that enabled Volkswagen to defraud pollution inspections, as was exposed in 2015; and still another will address the ways in which software systems are transforming healthcare. Each case study will focus on the responsibilities of the software developers in creating these systems, the impacts that their work will have (whether they know it or not), and the ethical considerations that they should always have in mind. The course will meet once a week, for 2.5 hours, which will be spent largely on discussions of student case studies, with only minimal lecturing by the Professor. Instead, it is expected that each class meeting will consist of discussions, focused by student presentations and examination of newsworthy events concerned with software systems. Each student will be required to do an in-depth case study over the course of the semester, will make at least two presentations about the case study and will write a thoroughly-researched, scholarly paper summarizing what the student has found and has learned. The final student presentation is expected to summarize what the student has learned through the study. Each student is expected to be an active participant in class discussions, and the instructor's evaluation of the student's participation in discussions will be weighted as an important component of the student's final course grade. 3 credits.

### 697L Seminar - Deep Learning 3 credits

INSTRUCTOR(S): Learned-Miller

This course will focus on modern, practical methods for deep learning. 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, and some elements of recurrent neural networks, such as long short-term memory networks (LSTMs). The emphasis will be on understanding the basics and on practical application more than on theory. Most applications will be in computer vision, but we will make an effort to cover some natural language processing (NLP) applications as well, contingent upon TA support. The current plan is to use Python and associated packages such as Numpy and TensorFlow. Prerequisites include Linear Algebra, Probability and Statistics, and Multivariate Calculus. Some assignments will be in Python and some in C++. 3 credits.

### 701 Advanced Topics in Computer Science 6 credits

INSTRUCTOR(S): STAFF

This is a 6 credit reading course corresponding to the master's project. The official instructor is the GPD although the student does the work with and is evaluated by the readers of his or her master's project.

### 891M Theory of Computation 1 credit

INSTRUCTOR(S): Mazumdar

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.

### 899 PhD Dissertation 1-9 credits

INSTRUCTOR(S): STAFF

### CICS 191CICS1 FYS - CICS First Year Seminar 1 credit

INSTRUCTOR(S): Hu; McCartin-Lim; Schein; Wemhoener

An exploration of computer science for first-year computer science majors and exploratory track students, focusing on a single topic. Sections 01, 03, 05, 07 will address “Big Data”: the ways in which the collection and analysis of data on a large scale affects our lives. Sections 02, 04, 06, 08 will address “Statistical Learning”: using and understanding a set of tools for modeling and understanding complex data sets. Sections 09, 10 are entitled “Dream Big” and will feature discussion of the ideas, decisions, and pioneers that historically shaped computing, with perspectives on how we might shape the future. Section 11 will explore the philosophy of artificial intelligence and the theory of consciousness, through readings, discussion, and other activities. Additional sections on other topics may be added during the summer. 1 credit.

Sections 01, 03, 05, 07 will address “Big Data”: Advances in large-scale data storage and processing, along with the availability of data for use by companies and governments from sources such as social media have increased the demand for new approaches to analyzing this information as well as individuals skilled in the application of these approaches. Students will learn about developments in data science and the ways in which the collection and analysis of data on a large scale affects their own lives as well as others. We will examine the impact of "Big Data" on corporations, government policy, research, and healthcare. Along with the technical issues involved with analyzing large data collections, we will examine the ethical and legal implications of these developments.

Sections 02, 04, 06, 08 will address “Statistical Learning”: Statistical learning provides a set of tools for modeling and understanding complex data sets, which are often encountered in daily research. In the past decade, with the explosion of data, statistical learning has become a very hot field in many scientific areas as well as marketing, finance, and other business disciplines. This course is intended to provide an introductory on statistical learning. Course topics include: Classical linear methods for regression and classification (linear regression, logistic regression); Model selection methods (cross-validation and the bootstrap); Advanced linear method and non-linear additive models (ridge regression and lasso); Tree-based methods and support vector machines; Dimension reduction techniques and clustering methods (principal components analysis, K-means clustering and hierarchical clustering).

### CICS 191CICS2 Computing and Informatics Exploratory RAP First Year Seminar 1 credit

INSTRUCTOR(S): Boisseau

A weekly seminar for students in the Computing and Informatics Exploratory Track RAP residential program. Exploration of strategies for success at UMass and computer science topics through presentations and problem-solving games. Open only to first-year students in the Computing and Informatics Exploratory Track RAP. This course does not count toward any requirements for the CS major or minor. 1 credit.

### CICS 191CMPS2 CS Major RAP First Year Seminar 1 credit

INSTRUCTOR(S): Barrington; Wileden

A weekly seminar for students in the CS RAP residential program. Exploration of computer science topics through presentations and problem-solving games. Mandatory pass/fail. Open only to first-year students in the CS RAP. This course does not count toward any requirements for the CS major or minor. 1 credit.

### CICS 596T Independent Study - Curricular Practical Training 1 credit

INSTRUCTOR(S): STAFF

### INFO 150 A Mathematical Foundation for Informatics 3 credits

INSTRUCTOR(S): Barrington

Mathematical techniques useful in the study of computing and information processing. The mathematical method of definition and proof. Sets, functions, and relations. Combinatorics, probability and probabilistic reasoning. Graphs and trees as models of data and of computational processes. Prerequisite: R1 math skills recommended. 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 190IN Introduction to Informatics 3 credits

INSTRUCTOR(S): Trim

An introduction to the main concepts of Informatics. There are several "Big Ideas" in computing, including but not limited to abstraction, data and information, algorithms, programming, and analysis of both computational problems and computational artifacts. 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 190INISH Introduction to Informatics (Honors Independent Study) 4-5 credits

INSTRUCTOR(S): Trim

An introduction to the main concepts of Informatics. There are several "Big Ideas" in computing, including but not limited to abstraction, data and information, algorithms, programming, and analysis of both computational problems and computational artifacts. 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. Includes an honors independent study component. 4-5 credits.