SCHOOL OF COMPUTER SCIENCE - Course Descriptions Spring 2015

CMPSCI 105 **Computer Literacy (R2)** **CREDITS 3**

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. 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 CMPSCI 119 or CMPSCI 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 CMPSCI majors. 3 credits.

CMPSCI 119 **Introduction to Programming** **CREDITS 3**

William Verts

The Internet has transformed computers from machines that calculate to machines that communicate. 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 CMPSCI majors. 3 credits.

CMPSCI 121 **Introduction to Problem Solving with Computers (R2)** **CREDITS 4**

Gordon Anderson

CMPSCI 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, CMPSCI 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.

CMPSCI 145 **Representing, Storing, and Retrieving Information** **CREDITS 3**

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.

CMPSCI 187 **Programming with Data Structures (R2)** **CREDITS 4**

Marc Liberatore

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. There will be weekly assignments and assignments in discussion sections consisting of programming and written exercises. There will also be several exams. Prerequisites: CMPSCI 121 (or equivalent Java experience) and Basic Math Skills (R1). Basic Java language concepts are introduced quickly; if unsure of background, contact instructor. 4 credits.

CMPSCI 197B **Special Topics - Advanced Assignments for CMPSCI 121** **CREDITS 1**

Gordon Anderson

This add on to CMPSCI 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 CMPSCI 121 concurrently. 1 credit.

CMPSCI 197C **Special Topics - Programming in C** **CREDITS 1**

Nicolas 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 CMPSCI 230 and courses that use C and C++. Prerequisites: CMPSCI 121 and 187. Runs for 6 weeks. This course is for CMPSCI minors and majors only, but it does not count towards either degree. 1 credit.

CMPSCI 197U **Special Topics - A Hands-on Introduction to UNIX** **CREDITS 1**

Jarred DeVaughn-Brown

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.

CMPSCI 197WP **Special Topics - Introduction to HTML/CSS** **CREDITS 1**

Nicolas 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: CMPSCI 121. Mandatory P/F. Students who are enrolled in or have taken CMPSCI 326 are not eligible to take this course. 1 credit.

CMPSCI 220 **Programming Methodology** **CREDITS 4**

Timothy Richards

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: CMPSCI 187 or ECE 242. 4 credits.

CMPSCI 230 **Computer Systems Principles** **CREDITS 4**

J. Eliot Moss, Timothy 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 the relevant features of modern architectures, and moves up the stack from there to operating system services such as programming language runtime systems, concurrency and synchronization, with a focus on key operating system features, I/O and networking, and distributed services. Prerequisites: CMPSCI 187 or ECE 242. 4 credits.

CMPSCI 240 **Reasoning Under Uncertainty** **CREDITS 4**

Andrew McGregor

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 Java 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. This is a core course for the new CMPSCI curriculum and may be used as a math elective for the old curriculum. Prerequisites: CMPSCI 187 (or ECE 242) and MATH 132 or consent of instructor. 4 credits.

CMPSCI 250 **Introduction to Computation** **CREDITS 4**

David Barrington

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 CMPSCI 187 (or ECE 242). MATH 132 may be used as a co-requisite with permission of instructor. 4 credits.

CMPSCI 305 **Social Issues in Computing** **CREDITS 3**

Michelle 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 technical communication genres, public writing, and academic writing. Students will produce approximately 10 informal writing assignments and 4-6 larger written projects. Writing experiences will also include writing for electronic environments, collaborative writing, and public writing; there will be one individual and one team presentation assignment. Prerequisite: ENGLWRIT 112. 3 credits.

CMPSCI 311 **Introduction to Algorithms** **CREDITS 4**

Ramesh Sitaraman

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 CMPSCI 250) is required. You should also be able to program in Java, C, or some other closely related language. Prerequisite: CMPSCI 250 or MATH 455. 4 credits.

CMPSCI 320 **Introduction to Software Engineering (IE)** **CREDITS 4**

Yuriy Brun *Honors Colloq*

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: CMPSCI 220. 4 credits.

CMPSCI 326 **Web Programming (IE)** **CREDITS 3**

Timothy 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: CMPSCI 220 or CMPSCI 230. 3 credits.

CMPSCI 348 **Introduction to Knowledge Discovery** **CREDITS 3**

David Jensen

Knowledge discovery is the process of discovering useful regularities in large and complex data sets. The field encompasses techniques from artificial intelligence (representation and search), statistics (inference), and databases (data storage and access). When integrated in to useful systems, these techniques can help human analysts make sense of vast stores of digital information. This course presents the fundamental principles of the field, 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: CMPSCI 187 (or ECE 242), CMPSCI 240, and CMPSCI 250. 3 credits.

CMPSCI 365 **Digital Forensics** **CREDITS 4**

Brian Levine

The goal of forensics is to gather artifacts for refinement into evidence that supports or refutes a hypothesis about an alleged crime or policy violation. Done correctly, forensics represents the application of science to law. The techniques can also be abused to thwart privacy. This course is a broad introduction to forensic investigation of digital information and devices. We 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. A separate lab discussion ensures that students gain hands-on instruction and experience with these systems each week. 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: CMPSCI 220 or CMPSCI 230; CS majors only. 4 credits.

CMPSCI 377 **Operating Systems** **CREDITS 4**

Deepak Ganesan

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: CMPSCI 230 with a grade of 'C' or better. 4 credits.

CMPSCI 383 **Artificial Intelligence** **CREDITS 3**

Hava Siegelmann

The Course explores key concepts of artificial intelligence, including state-space and heuristic search techniques, game playing, knowledge representation, 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: CMPSCI 220 (or CMPSCI 230) and CMPSCI 240. 3 credits.

CMPSCI 390CG **Introduction to Computer Graphics** **CREDITS 3**

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, digital photography, 2D/3D modeling, 3D graphics pipeline, OpenGL, 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 Java. This course counts as a CS Elective toward the CMPSCI major (BA/BS). Students who have taken CMPSCI 473 are not eligible to take this course. Prerequisites: CMPSCI 187 (or ECE 242) and CMPSCI 190DM (or MATH 235 or CMPSCI 240 or equivalent courses from other departments). 3 credits.

CMPSCI 397G **Special Topics - Creative Game Design and Development** **CREDITS 3**

Joshua Newman

In this class, we will explore, through a series of projects, the fundamental questions of game design. What are the common features of hopscotch, Skyrim, boxing, Farmville, poker, and Tic-Tac-Toe? How do you create an engrossing, challenging, vivid, or surprising environment of play? How do you determine the value of skill, chance, cooperation, and competition in game play? What effect does the social, sexual, gender, political, and economic environment of the game's creation have on the play of the game? This course will introduce students to theoretical structures and demonstrate their implementation through the development of several small physical and digital games, ending with a group digital game project. Students will be evaluated based on class participation, process papers, and the creation of their own games. Not for CS Major/Minor Requirements. 3 credits.

CMPSCI 397J **Special Topics - Introductory Javascript Game Development** **CREDITS 1**

Colin Tincknell

Students will be taught the basics of game development using a Javascript/HTML5 game engine. This course will be project based, with students implementing features onto simple existing games to familiarize themselves with the engine. After becoming comfortable with the engine, students will propose a game project and then develop it from scratch. Prerequisite: CMPSCI 326. Not for CS Major/Minor Requirements. 1 credit.

CMPSCI 445 **Information Systems** **CREDITS 3**

Yanlei Diao

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

CMPSCI 446 **Search Engines** **CREDITS 3**

James Allan

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

CMPSCI 453 **Computer Networks** **CREDITS 3**

John Ridgway

This course provides an introduction to fundamental concepts in computer networks, including their design and implementation. Topics covered include the Web and other applications, transport protocols (providing reliability and congestion control), routing, and link access. Special attention is also paid to wireless networks and security. Homework assignments involve programming and written tasks. Prerequisites: Experience programming; CMPSCI 230 (or CMPSCI 377). 3 credits.

CMPSCI 491G **Seminar - Computer Networking Lab** **CREDITS 3**

Arunkumar Venkataramani

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 CMPSCI major (BA/BS). Prerequisite: CMPSCI 453. 3 credits.

CMPSCI 496S **Independent Study - Scalable Web Systems** **CREDITS 1**

Timothy Richards

Managing agile development of complex web systems is a difficult task. These applications require a broad range of conceptual understanding ranging from user interface design and client/server-side concepts and programming to caching, monitoring, scalability, and performance issues. These techniques need to work collectively to bring the end user a single working system. Unlike traditional waterfall methods of software development that demand a focus on design before development, agile development emphasizes techniques that require an iterative design, development, and deploy process. Test driven development is the focus and a working system is always available for review. This 1 credit course will require students to be managers that lead a software development team (students enrolled in 497S) to the successful completion and deployment of a real web application. Students enrolled in this course will be required to hold team meetings, manage work and design, and evaluate the individual work and contributions of students taking 497S and meet with the instructor to read and discuss agile software development and techniques from the literature. Prerequisites: CMPSCI 230 or 220 and 326. Co-requisites: CMPSCI 497S. 1 credit.

CMPSCI 497S **Special Topics - Scalable Web Systems** **CREDITS 3**

Timothy Richards

The web has become a large and complex area for application development. Access to an abundance of open source languages, libraries, and frameworks has led to the quick and easy construction of a variety of applications with several moving parts working in coordination to present to the user the illusion of a single program. In reality, web applications and services are extremely difficult to get right. They involve complicated user interfaces, multiple databases, security holes and performance issues, and a multitude of ever changing remote services, spread across several physical and virtual machines. These complications are further stressed by the large number of concurrent users that access these applications every second. This complexity leads to failed applications, performance bottlenecks, and security breaches. This course focuses on all areas of building scalable web systems. Topics include the study of cyber security, big data, system design, best practices, redundancy, scalability, performance and monitoring, and key principles in the agile development of complicated distributed web applications. Students will work as a team to design, develop, monitor, and analyze a large and scalable web application. Each student will focus on an area of interest and use what they have learned to contribute to the deployment of a real web application. Prerequisites: CMPSCI 230 or 220 and 326. 4 credits.

CMPSCI 501 **Formal Language Theory** **CREDITS 3**

David Barrington

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

CMPSCI 520 **Software Engineering: Synthesis and Development** **CREDITS 3**

Reda Bendraou

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: CMPSCI 320 with a grade of 'C' or better. 3 credits.

CMPSCI 529 **Software Engineering Project Management** **CREDITS 3**

Yuriy Brun

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

CMPSCI 589 **Machine Learning** **CREDITS 3**

Benjamin Marlin *Honors Colloq*

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: CMPSCI 383 and MATH 235. Restrictions: students that have already taken CMPSCI 689 may not take CMPSCI 589. 3 credits.

CMPSCI 597CR **Special Topics - Crypto Engineering [Cross-List w/ECE 597CR]** **CREDITS 3**

Christof Paar

Cryptographic algorithms are the "engine" that drives virtually every security solution, from email encryption to biometric passports. The course has three parts. (1) Review of the most imporant symmetric and public-key crypto algorithms, (2) the efficient implementation and (3) the secure implementation of cryptographic algorithms. Efficient and secure implementation are both theoretically interesting and important in practice. With respect to efficient implementations, we will study exponentiation algorithms and fast multiplication with long numbers, which are the performance bottleneck in many applications. Secure implementations is concerned with breaking crypto algorithms based on they way they are realized. We will study how to extract crypto keys based on faulty cipher outputs. A topic of particular interest in industry and academia are side-channel attacks. Here, we try to extract cryptographic keys from partial information obtained from power measurements. The methods are all algorithmic in nature. The course projects are concerned with research topics from the literature. The course grade will be based on homework assignments and projects. 3 credits.

CMPSCI 601 **Computation Theory** **CREDITS 3**

Neil Immerman

An in-depth introduction to the main models and concepts of the theory of computation, including: Computability: what problems can be solved in principle; Complexity: what problems can be solved in a given amount of time, space, parallel time; Logic: how do formal specification and proof mirror other forms of computation? Students will learn to go from a concrete problem to a mathematical model; and, after proving things about the mathematical model, to correctly interpret what they have learned about the concrete problem. Prerequisites: an undergraduate course in automata theory and formal languages such as CMPSCI 501 or permission of instructor. Course requirements: biweekly problem sets, midterm and final. Also open to qualified undergraduates. 3 credits.

CMPSCI 603 **Robotics** **CREDITS 3**

Roderic Grupen

This course is designed to be a advanced course in robotics that covers mechanisms (kinematics and dynamics), actuators, sensors, feedback control, and signal processing. The target is to provide an understanding of robot systems that interact with, interpret feedback from, and manipulate the world about them. We will relate the subject matter to biological systems whenever possible, including discussion about the relationships between learning and development in human beings and what it has to say about programming robots. Students will experiment with system identification and control, image processing, path planning, grasping, and machine learning to reinforce the material covered in class. 3 credits.

CMPSCI 617 **Computational Geometry** **CREDITS 3**

Ileana Streinu

Geometric algorithms lie at the heart of many applications, ranging from computer graphics in games and virtual reality engines to motion planning in robotics or even protein modeling in biology. This graduate course is an introduction to the main techniques from Computational Geometry, such as convex hulls, triangulations, Voronoi diagrams, visibility, art gallery problems, and motion planning. The class will cover theoretical as well as practical aspects of the field. The goal of the class it to enable students to exploit a broad range of algorithmic tools from computational geometry to solve problems in a variety of application areas. Prerequisite: Mathematical maturity; CMPSCI 611 or CMPSCI 601. Eligibility: Graduate CS students only. Others with permission of instructor. 3 credits.

CMPSCI 620 **Advanced Software Engineering: Synthesis and Development** **CREDITS 3**

Reda Bendraou

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.

CMPSCI 645 **Database Design and Implementation** **CREDITS 3**

Alexandra Meliou

This course covers the design and implementation of traditional relational database systems and advanced data management systems. The course will treat fundamental principles of databases: the relational model, conceptual design, query languages, and selected theoretical topics. We also cover core database implementation issues including storage and indexing, query processing and optimization, as well as transaction management, concurrency, and recovery. Additional topics will address the challenges of modern Internet-based data management. These include data mining, provenance, information integration, incomplete and probabilistic databases, and database security. 3 credits.

CMPSCI 660 **Advanced Information Assurance** **CREDITS 3**

Amir Houmansadr

This course provides an in-depth examination of the fundamental principles of information assurance. While the companion course for undergraduates is focused on practical issues, the syllabus of this course is influenced strictly by the latest research. We will cover a range of topics, including authentication, integrity, confidentiality of distributed systems, network security, malware, privacy, intrusion detection, intellectual property protection, and more. Prerequisites: CMPSCI 460 or 466, or equivalent. 3 credits.

CMPSCI 677 **Operating Systems** **CREDITS 3**

Prashant Shenoy

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., CMPSCI 377) will be helpful but not required. Lect 2 is on-line. 3 credits.

CMPSCI 688 **Probabilistic Graphical Models** **CREDITS 3**

Daniel Sheldon

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

CMPSCI 689 **Machine Learning** **CREDITS 3**

Subhransu Maji

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.

CMPSCI 690AA **Approximation Algorithms and Combinatorial Optimization** **CREDITS 3**

Barna Saha

Many important problems that arise in practical applications of discrete optimization are NP-hard. This implies no polynomial time algorithms exist for these problems unless P==NP. The field of approximation algorithms has developed to tackle this difficulty by designing polynomial time algorithms to solve otherwise intractable problems near-optimally. Approximation algorithms provide rigorous guarantees on approximation factors indicating how far the solution can be in the worst case. This paradigm has become a cornerstone in algorithm design, and this course aims to cover a comprehesive list of topics in this area at the graduate level. Towards the end of the course, we will also explore "hardness of approximation": study of the best approximation factor possible in polynomial time. A tentative list of topics include: Techniques: Greedy algorithms, local search, dynamic programming, randomized methods, LP techniques, primal-dual method, lagrangian methods, semi-definite programming, metric method, hardness of approximation. Problems: Set cover, Vertex cover, TSP and other planning problems, Scheduling and Generalized assignment problems, Facility Location, Steiner tree and other network design problems, Sparsest cut, multicut and other graph partitioning problems, MaxSat, Graph coloring, Approximate counting, Algorithms on sequences etc. 3 credits.

CMPSCI 691DA **Seminar - Big Data Algorithms and Applications** **CREDITS 1**

Barna 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 goal is two fold, (i) to study a subset of topics that build the mathematical foundation of big data processing, and (ii) to learn about applications of big data in important domains of interest. For the former, our plan is to concentrate on the following three topics: Stochastic Optimization, Metric Embedding, and High Dimensional Fourier Analysis. For the later, we will emphasize on the following two areas: Biomedical Applications and Social Network Analysis. This is a one credit seminar course. Depending on the number of students, each student will read 1 theory and/or 1 application paper and present them in the class. The presentation of theory paper needs to be in-depth covering the detailed proofs of main theorems and lemmas. Prerequisites: The students are expected to have strong mathematical foundations, must have basic knowledge of algorithms and probability, and should be able to read and understand papers appearing in top theoretical computer science conferences. Senior undergraduate students meeting these requirements are encouraged to take this course. 1 credit.

CMPSCI 691DD **Seminar - Research Methods in Empirical Computer Science** **CREDITS 3**

Emery Berger

This course introduces graduate students to basic ideas about conducting a personal research program. Students will learn basic methods for activities such as reading technical papers, selecting research topics, devising research questions, planning research, analyzing experimental results, modeling and simulating computational phenomena, and synthesizing broader theories. The course will be structured around three activities: lectures on basic concepts of research strategy and techniques, discussions of technical papers, and preparation and review of written assignments. Significant reading, reviewing, and writing will be required, and students will be expected to participate actively in class discussions. 3 credits.

CMPSCI 691MA **Seminar - Social Media Analysis and Computational Social Science** **CREDITS 1-3**

Brendan O'Connor

Can we use computation to study society? As computing appears everywhere in daily life, computational techniques could help us understand key social scientific questions. But also, since computing is becoming more social, insights from social science may help us design better systems for users. This seminar will consist of readings and presentations on (1) social media analysis, and (2) computational social science. Social media is one interesting and recent manifestation of computation in everyday life, and lends itself to studies on topics from mental health to the evolution of slang to the emergence of fads to the dynamics of social unrest. This data's richness and magnitude ("Big"-ness) requires non-trival computational methods for analysis, and comes with major questions about validity and representativeness. At the same time, we will investigate the social science literature and consider techniques and insights that may help achieve a deeper understanding of these phenomena, and better appreciate the important questions to ask. Furthermore, we will look at case studies of applying computational methods (like text analysis / NLP, network analysis, latent-variable statistical models, agent-based simulations, etc.) to understand social phenomena in other contexts. These have been developed in many other areas than just CS. For example, in the 1980's, political scientists developed what we would now call an unsupervised machine learning method to infer the ideological positions of legislators from their voting behavior, in order to answer questions about the structure of American politics. Work like this might inspire us to think of how related methods aid insight in other areas. Lect 01=3 credits; Lect 02=1 credit.

CMPSCI 701 **Advanced Topics in Computer Science** **CREDITS 6**

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.

CMPSCI 891M **Theory of Computation** **CREDITS 1**

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.

CMPSCI H320 **Honors Colloquium for CMPSCI 320** **CREDITS 1**

Yuriy Brun

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 CS 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.

CMPSCI H589 **Honors Colloquium for CMPSCI 589** **CREDITS 1**

Benjamin Marlin

This course is an honors colloquium for CMPSCI 589: Machine Learning. It will include an exploration of the mathematical foundations of the machine learning algorithms presented in class. It will also include the presentation of more advanced models and algorithms. Students will participate in and lead group discussions on these topics, as well as on their final course projects. 1 credits.