CMPSCI 105  Computer Literacy (R2)  
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  
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)  
Robert Moll  
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 191P 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  
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)  
Gerome Miklau, Timothy Richards  
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  
Robert Moll  
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.
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.

Special Topics - Programming in C
Nicolas Scarrci

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.

Special Topics - A Hands-on Introduction to UNIX
Jarred DeVaugh-Brown

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.

Computer Systems Principles
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. This course can be used as a "core" course for students in the CMPSCI minor. Prerequisites: CMPSCI 187 or ECE 242. 4 credits.

Reasoning Under Uncertainty
Andrew McGregor

Development of mathematical reasoning skills for problems that involve uncertainty. Each concept will be illustrated by real-world examples and demonstrated through in-class and homework exercises, 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.

Introduction to Computation
David Barrington


Social Issues in Computing
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. 3 credits.
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)**

David Fisher

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

**CMPSCI 326  Web Programming (IE)**

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 187 or ECE 242. 3 credits.

**CMPSCI 370  Introduction to Computer Vision**

Erik Learned-Miller

This introductory computer vision class will address fundamental questions about getting computers to "see" like humans. We investigate questions such as -What is the role of vision in intelligence? -How are images represented in a computer? -How can we write algorithms to recognize an object? -How can humans and computers "learn to see better" from experience? We will write a number of basic computer programs to do things like recognize handwritten characters, track objects in video, and understand the structure of images. Prerequisite: CMPSCI 240 or 383. 3 credits.

**CMPSCI 377  Operating Systems**

Sean Barker

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**

Philip Thomas, William Dabney

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.
This course introduces the fundamental concepts of two-dimensional (2D) and three-dimensional (3D) computer graphics. It covers the basic methods needed to model, render, and animate 3D objects. Topics include raster displays, line drawing, affine and perspective transformations, windows and viewports, clipping, visibility, illumination models, reflectance models, radiometry, curves and surfaces, shading, texture mapping, ray tracing, graphics toolkits, animation, and 3D printing. Throughout the class, students will learn algorithmic ways to model the visual world, and write Java programs to implement various computer graphics algorithms. 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.

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.

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.

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) and CMPSCI 240 (or STATS 515). 3 credits.

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
Leon Osterweil
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
David Fisher
The purpose of this course is to provide students with practical experience in the management of software development projects. Students in this course will gain this experience by serving as software development team technical managers for teams of software engineering students in 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 535  Computer Architecture
Charles Weems
Honors Colloq
The structure of digital computers is studied at several levels, from the basic logic level, to the component level, to the system level. Topics include: the design of basic components such as arithmetic units and registers from logic gates; the organization of basic subsystems such as the memory and I/O subsystems; the interplay between hardware and software in a computer system; the von Neumann architecture and its performance enhancements such as cache memory, instruction and data pipelines, coprocessors, and parallelism. Weekly assignments, semester project, 2 hours exams, final. Prerequisites: CMPSCI 391IB. 3 credits.

CMPSCI 603  Robotics
Roderic Grupen
This course is designed to be a advanced course in robotics that covers mechanisms (kinematics and dynamics), actuators, sensors, signal processing, 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 620  Advanced Software Engineering: Synthesis and Development
Leon Osterweil
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 635  Modern Computer Architecture
Charles Weems
This course examines the structure of modern computer systems. We explore hardware and technology trends that have led to current machine organizations, then consider specific features and their impact on software and performance. These may include superscalar issue, caches, pipelines, branch prediction, and parallelism. Midterm and final exams, individual projects, homework, in-class exercises. Prerequisites: CMPSCI 535 or equivalent. 3 credits.
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 677 Operating Systems**  
*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**  
*Benjamin Marlin*

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 691CL Seminar - Computational Linguistics: Syntax and Semantics**  
*Andrew McCallum*

This course is an introduction to computational linguistics syntax and semantics, including statistical processing of corpora. In syntax, this course will cover computational grammars such as Tree-Adjoining Grammars, Combinatory Categorial Grammars, Head-driven Phrase Structured Grammars, etc. In semantics, it will cover computational lexicons such as PropBank, FrameNet, VerbNet. There will be weekly lectures and paper readings, in-class exercises, small writing assignments and a research-relevant final project. This course counts as a CS Elective toward the CMPSCI major (BA/BS). 3 credits

**CMPSCI 691CO Seminar - Content-Oriented Networking**  
*Donald Towsley, James Kurose*

While the initial design of the Internet was focused on “delivering data between computers or between computers and terminals”, today’s Internet is more concerned with connecting people with content and information. This more recent use has simulated the development of content distribution networks (CDNs) such as Akamai, pub-sub systems, and peer-to-peer content distribution software such as BitTorrent. Most recently, there has been considerable research on extending the Internet and clean-slate architectures to support content as “the” first class citizen. In this seminar we will explore through a sequence of readings: 1) different approaches to supporting content ranging from CDNs and P2P systems that operate “on top of” the current Internet to revolutionary clean-slate Content Centric Networking designs to more evolutionary designs; 2) approaches and challenges raised by Internet use that now has more mobile wireless endpoints than fixed, wired endpoints; 3) quantitative methods for establishing performance tradeoffs and quantitatively characterizing performance of different configurations/designs; 4) algorithms for locating content in a distributed, decentralized manner. Students can register for this course for either one credit or three credits. All students will be expected to make at least one presentation for a minimum of one credit. In addition, students taking this course for three credits will propose and complete a course project in consultation with the instructors. Lect 01=3 credits; Lect 02=1 credit.
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 691DD**  
Seminar - Research Methods in Empirical Computer Science  
*Emery Berger, David Jensen*

Programming languages are fundamental to computer science and everyday programming. This course is an in-depth introduction to the formal, logical foundations of programming languages. The class will teach you how to precisely formulate questions about programs and programming languages, such as “Does this program behave as expected?”, “What errors does this type-checker catch?”, “Is this compiler optimization correct?”, “Is this program secure?”, “Is this language more expressive than some other language?”, and many others. To answer these questions, the course will introduce topics such as dynamic semantics, type systems, program logics, and abstract machines. The course will have regular homework assignments. Most assignments will require you to prove properties of simple programming languages and programs. There may be a few light programming assignments. The course will conclude with a project that has you explore some area of programming languages in greater depth. 3 credits.

**CMPSCI 691SU**  
Seminar - Computational Sustainability: Algorithms for Ecology and Conservation  
*Daniel Sheldon*

Ecosystems across the globe under threat. Defining good conservation policies is difficult because many aspects of ecosystems and our impacts on them are poorly understood. However, new data resources are emerging that can help us understand and manage ecosystems more effectively, if we can develop the algorithms to understand and use this data well. For example, some algorithmic problems in ecology include: where to place sensors, how to fit models of complex ecological such as bird migration from noisy and incomplete data, and where to place wildlife reserves or corridors to maximize the benefits to threatened species. In this seminar, students will read and discuss recent papers at the intersection of CS and ecology, with a focus on novel work in machine learning, AI, and discrete optimization. The instructor will also present open problems, data resources, and research ideas. Students taking the course for three credits will complete a project or extended survey. Students should have completed a Ph.D.-level course in machine learning or algorithms or receive permission from the instructor. Prerequisites: One of CMPSCI 611 (or 688 or 689 or permission of the instructor). Lect 01=3 credits; Lect 02=1 credit.

**CMPSCI 701**  
Advanced Topics in Computer Science  
*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  
*David Barrington*

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

**CMPSCI H305**  
Honors Colloquium for CMPSCI 305  
*Michelle Trim*

Weekly meetings will be spent deepening discussions of topics covered in class, occasionally with additional reading or extended versions of the reading covered in class, and developing research strategies related to an ongoing annotated bibliography project. Students completing the colloquium will have the opportunity to strengthen summary, analytical writing, and critical source evaluation skills. 1 credit.

**CMPSCI H320**  
Honors Colloquium for CMPSCI 320  
*David Fisher*

The purpose of this course is to provide students with supplementary material and insights about the software development enterprise. Students meet once a week for a one-hour discussion of software engineering topics whose exploration is intended to provide depth and perspective on the regular material of 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.
The colloquium will focus on advanced topics and recent research topics related to information management and databases. Students will participate in group discussions and carry out a group or individual project which will be an extension to the project work in CMPSCI 445. Students will be graded based on their active participation during meetings, written summaries of assigned readings, and project work. 1 credit.

The honors section of CMPSCI 535 provides an opportunity for University Honors students enrolled in the class to take a deeper look at some aspect of computer architecture or its underlying technology. The specific choice of topics is agreed upon by the instructor and student on an individual basis. Students may choose to explore the history of some aspect of architecture or technology, look at market influences on the science and engineering of computer hardware, experiment with a novel computer design through simulation, conduct a series of in-depth readings leading to a semester thesis, or other suitable work done under regular consultation with the instructor. Recommended for Juniors, Seniors; Majors. 1 credit.