**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 191P 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 121**  
**Introduction to Problem Solving with Computers (R2)**  
*Erik Learned-Miller*

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)**  
*Brian Levine*

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 191P**  
**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 197C**  
**Special Topics - Programming in C++**  
*Nicolas Scarcci*

A brief introduction to the C++ programming language for students with a good working knowledge of Java. This course is good preparation for CMPSCI 291SP and courses that use C++. Prerequisites: CMPSCI 121 and 187 or permission of instructor. Runs for 8 weeks. This course is for CMPSCI minors and majors only, but it does not count towards either degree. 1 credit.
This course offers a 4-week introduction to working with Unix, and it is intended to help students work with tools commonly used in CS courses. The fall semester offering of this class moves more quickly and covers more topics than the spring semester. 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 220 Programming Methodology**
*J. Eliot Moss*
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**
*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 to fulfill the requirement of CMPSCI 201 for students in the CMPSCI major. This course can be used as a "core" course for students in the CMPSCI minor. Prerequisites: CMPSCI 187 or ECE 242. 4 credits.

**CMPSCI 240 Reasoning Under Uncertainty**
*Hanna Wallich*
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 -- probability definitions, conditional probability, independence, Bayes' theorem, counting problems, binomial distribution, mean, variance, Markov and Chebyshev bounds. Probabilistic reasoning -- inference, parameter estimation, naive Bayes classifiers, Bayesian inference. Markov chains, 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**
*David Barrington*

**CMPSCI 305 Social Issues in Computing**
*Stephen Constantine*
Designed to satisfy the Junior Year writing requirement, CMPSCI 305 introduces the student to technical writing and editing, scientific journalism and the social essay. The course combines practical, scientific writing as found in industry and business with explorative essays that focus attention upon the technological and humanistic concerns inherent in society. Ten written assignments-two longer papers, eight shorter ones and one oral presentation. Prerequisite: ENGLWP 112 or CW. 3 credits.

**CMPSCI 311 Introduction to Algorithms**
*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.
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 and in-class presentations, two exams, and a major term project. Prerequisite: CMPSCI 220 or CMPSCI 287. 4 credits.

**CMPSCI 377** Operating Systems  
*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 187 (or ECE 242) and CMPSCI 201 (or CMPSCI 230 or ECE 232). Strongly recommend CMPSCI 220 with a ‘C’ or better prior to taking this course.  4 credits.

**CMPSCI 383** Artificial Intelligence  
*Sridhar Mahadevan*

The Course explores key concepts of artificial intelligence, including 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: (CMPSCI 287 or CMPSCI 220 or CMPSCI 230) and (CMPSCI 240 or CMPSCI 311).  3 credits.

**CMPSCI 390WP** Web Programming  
*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. Prerequisites: CMPSCI 187 or ECE 242. This course is for CMPSCI minors and majors and counts as a CS Elective toward the major.  3 credits.

**CMPSCI 401** Formal Language Theory  
*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 401.  3 credits.

**CMPSCI 445** Information Systems  
*Gerome Miklau*

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 287, or 230) and CMPSCI 311.  3 credits.
CMPSCI 446  Search Engines
David Smith
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). This course may be used in place of CMPSCI 445 as a semi-required course under the OLD rules for the BS. Prerequisite: CMPSCI 240 or CMPSCI 383, or equivalent. 3 credits.

CMPSCI 491DD  Seminar - Research Methods for Honors Students
David Jensen
This course introduces to commonwealth honors completing department capstones 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, all centered on the student's specific capstone project. Significant reading, reviewing, and writing will be required, and students will be expected to participate actively in class discussions. 3 credits.

CMPSCI 491G  Seminar - Computer Networking Lab
Parviz Kermani
In this course, students will learn how to put "principles into practice," in a hands-on-networking lab course. The course will cover router, switches and end-system labs in the areas of Single Segment IP Networks, Multiple Segment IP Networks and Static Routing, Dynamic Routing Protocols (RIP, OSPF and BGP), LAN switching, Transport Layer Protocols: UDP and TCP, NAT, DHCP, DNS, and SNMP. Students will also get engaged in evaluating power consumption of network components as an aid in the design of energy efficient (green) networks. This course counts as a CS Elective toward the CMPSCI major. Prerequisite: CMPSCI 453. 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, or E&C-ENG 373 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 575  Combinatorics and Graph Theory [Cross-List w/MATH 513]
Robert Moll
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. Prerequisites: mathematical maturity; calculus; linear algebra; strong performance in some discrete mathematics class, such as CMPSCI 250 or MATH 455. Modern Algebra - MATH 411 - is helpful but not required. 3 credits.
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: Undergraduate-level courses in discrete mathematics (CMPSCI 250) and algorithms (CMPSCI 311). Additional mathematical maturity (e.g., CMPSCI 611, A's in 250 or 311, or other mathematical background) quite desirable as is background in automata theory and formal languages (CMPSCI 401). Course requirements: biweekly problem sets, midterm and final. Also open to qualified undergraduates. 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.

CMPSCI 691AV Seminar - Advanced Computer Graphics
Rui Wang
This course covers selected topics in modern computer graphics research. Specific subjects include photorealistic image synthesis, machine learning in computer graphics, stochastic sampling and patterns, and geometry processing. This course will be a mix between lectures and student presentations. 3 credits.

CMPSCI 691DD Seminar - Research Methods in Empirical Computer Science
David Jensen
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.
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 691GM Seminar - Graphical Models
Benjamin Marlin

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

CMPSCI 691ST Seminar - Systems
Emery Berger

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 701 Advanced Topics in Computer Science
STAFF

An introduction to some more advanced algorithmic topics with a focus on randomization and probabilistic techniques. Topics will include the probabilistic method; tail inequalities; entropy and information; random walks; derandomization and limited independence. Applications to approximation and combinatorial optimization; online and stream computation; communication theory; and other areas will be discussed as time permits. Prerequisite is CMPSCI 611 or equivalent. 3 credits.

CMPSCI 711 Parallel Algorithms and Architectures
Andrew McGregor

A cognitive architecture is a type of agent architecture designed to provide a theory for understanding and simulating human cognition. The aim of the designers of cognitive architectures is to improve our understanding of a wide range of human cognitive functions. In this seminar we will study a variety of cognitive architectures in order to understand how they work, what they are able to do, and how they contribute to cognitive science. Prerequisites: basic knowledge of artificial intelligence and/or cognitive psychology. 3 credits.

CMPSCI 791CA Seminar - Cognitive Architectures
Andrew Barto

Descriptive Complexity measures the computational complexity of a property via how rich a logical language is needed to express the property. All the main complexity classes have natural descriptive characterizations. In this seminar we will bring all the participants up to speed concerning descriptive complexity as described in my 1999 book, Descriptive Complexity. We will also delve into more recent developments in descriptive complexity including the work of Ben Rossman and Martin Grohe. All students will carefully read the text and papers in advance of their class presentation, and later in the course, each student will present some of the material. Prerequisite: CMPSCI 601 or permission of the instructor. 3 credits.

CMPSCI 791SN Seminar - Computational Social Neuroscience
Hava Siegelmann

Our social interactions stem at who we are, neurologically, psychologically, and morally. The course will focus on the correlations of brain chemicals, substances, life experiences, cognition, memory and social behavior. We will focus on these from the computational point of view and design applications for most leading artificial intelligence systems and most advanced human-computer interfaces. Grades will be based on material presentation as well as on a semesterial projects (to be done in singles or pairs) with topics decided together between students and professor. 3 credits.
Optimization consists of maximizing some function of one or several variables. In this seminar we will read and discuss research in optimization of both continuous and discrete functions. We will emphasize optimization in graphical models, but not cover optimization over time (optimal control). The focus will be on gaining understanding of methods, not on applications. Students should have knowledge of graphical models. 1 credit.

Andrew McCallum

Seminar - Topics in Optimization

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.

The students in the course will read a number of works of fiction that explore the role technology plays in modern society. A final term paper will be required. There will be one hour-long class meeting per week, at a time to be arranged. The seminar will be one credit, with a grade independent of the CMPSCI 305 grade. Concurrent registration in 305 is required, but if space permits we may admit students who have _previously_ taken 305. 1 credit.

David Fisher

Honors Colloquium for CMPSCI 305

Honors Colloquium for CMPSCI 320

David Barrington

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.

David Barrington

Honors Colloquium for CMPSCI 401

Gerome Miklau

The colloquium will focus on advanced topics and recent research topics related to information management and databases. Student will participate in group discussions and carry out a group 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 reading, and group project work. 1 credit.