Computer Science

The major and minor in computer science are administered by the Department of Computer Science: Associate Professors Ballesteros, Lerner, St. John; Assistant Professors Pon-Barry, Sheldon. Visiting Lecturers DeFlumere, Klemperer.

Overview

Computer science is an exciting field with applications across many other disciplines, including biology, chemistry, physics, environmental science, mathematics, economics, sociology, psychology, and art. The main role of a computer scientist is that of a problem solver. A degree in the field signifies formal training in computational and analytical approaches to problem solving as well as the skills necessary to develop software to tackle new challenges. These computational approaches can be applied to a wide spectrum of problems, including protein folding and flexibility, modeling and forecasting bird migration, improving on the capabilities of search engines to retrieve the most relevant documents, understanding how the connectedness provided by social networks impact the lives we lead, supporting scientists in the management and analysis of the data they collect, developing video games and computer animations, and more. In truth, it is difficult to think of a scenario in which the tools acquired in computer science do not provide a powerful advantage.

Getting Started

The recommended way to begin a study of computer science is with Computer Science 101. This course is an introduction to the use of computers as a problem-solving tool. Students with programming experience may consider beginning with 201. Any member of the computer science faculty can advise students who have questions about their course of study. COMSC-101 and 201 are offered both semesters.

Contact Info

Wendy Queiros, senior administrative assistant
Barbara Lerner, chair

Requirements for the Major

Credits

- A minimum of 40 credits

Courses

- Computer science (36 credits):
  - 101, Problem Solving and Object-Oriented Programming
  - 201, Advanced Object-Oriented Programming
  - 211, Data Structures
  - 221, Computer System & Assembly Language
  - 312, Algorithms
  - 322, Operating Systems
  - Three additional computer science courses (12 credits). Two of these must be at the 300 level (8 credits). The third may be at either the 200 level or 300 level (4 credits). Independent study courses do not count as electives.
- Mathematics (4 credits):
  - 232, Discrete Mathematics

The skills and abstract reasoning of mathematics are especially important in computer science. It is strongly recommended that students take additional mathematics courses (at least through Mathematics 101 and 102). Mathematics 211, Linear Algebra, is very useful for some fields, like machine learning and computer graphics. Computer science majors who elect a mathematics or statistics minor may not count Mathematics 232 for credit in both mathematics or statistics and computer science.

Students may also count Philosophy 225, Symbolic Logic as an elective because of its strong connection to the foundations of computer science.

Students planning to pursue an advanced degree in computer science should take additional computer science courses and include independent research leading to a thesis in their plans.

Requirements for the Minor

Credits

- A minimum of 20 credits

Courses

- Computer science:
  - 101, Problem Solving and Object-Oriented Programming
  - 201, Advanced Object-Oriented Programming
  - 211, Data Structures
  - Two additional computer science courses (8 credits), including one at the 300 level. The second can be at either the 200 level or 300 level.

Other

Honors

To graduate with honors in computer science, a student must complete a project and write an accompanying thesis. This is often a full year commitment, during which the student works closely with a faculty member to explore a topic in depth by reading research papers, writing programs, and experimenting with ideas. Preliminary research usually begins in the summer following her junior year, with the student submitting and defending a thesis proposal early in the fall of her senior year. Upon department approval of this proposal, she will complete the research during her senior year, writing and defending her thesis in the spring. Some honors students attend conferences and/or coauthor papers with their mentors.

Programming-Intensive Courses

Courses designed to offer students significant software design and programming experience are labeled as "Programming intensive."

Course Offerings

COMSC-100 An Introduction to Computer Science
Fall and Spring
An introduction to basic computer science concepts with a focus on Python programming to manipulate images and sounds. Laboratory assignments will provide the main programming opportunities with Python, images, and sounds, while lectures will cover topics such as the origins of computing, computer architecture, artificial intelligence, and robotics.
Applies to requirement(s): Math & Sciences
B. Lerner, The department
Coreq: COMSC-100L
Notes: Students should NOT take this course after Computer Science 101
Credits: 4
COMSC-101 Problem Solving and Object-Oriented Programming  
Fall and Spring  
Computers are used every day for an enormous variety of tasks, from playing games and chatting with friends to transferring billions of dollars, delivering radiation treatments, and controlling the electrical grid. Computer programs are an essential ingredient in allowing for this great diversity of applications. In this course, you will learn to create your own programs, based on core programming concepts and analytical problem solving approaches. You will develop dynamic programs by first using Adobe Flash AS3 (ActionScript 3), the technology behind many Web applications. The last portion of the course will teach you Java, a very popular modern programming language. We assume no prior study of computer science. Programming intensive.  
Applies to requirement(s): Math & Sciences  
A. DeFlumere, A. St. John, The department  
Coreq: COMSC-101L.  
Credits: 4  

COMSC-103 Networks  
Fall  
How do opinions, fads, and political movements spread through society? What makes food webs and financial markets robust? What are the technological, political, and economic forces at play in online communities? This course examines connections between the social, technological, and natural worlds through the lens of networks. Students will learn basics of graph theory and game theory and apply them to build mathematical models of processes that take place in networks.  
Applies to requirement(s): Math & Sciences  
D. Sheldon  
Credits: 4  

COMSC-106 Introduction to Scientific Computing  
Spring  
The matrix-based programming language Matlab will be used to introduce students to programming and fundamental computing methodologies such as top-down design and objects. Course work will involve vectors, matrices, numerical integration and differentiation, curve-fitting and graphics. The course is programming intensive.  
Applies to requirement(s): Meets No Distribution Requirement  
A. St. John  
Prereq: Mathematics 101.  
Credits: 4  

COMSC-109 iDesign Studio  
Fall  
Designers are continually innovating ways of incorporating technology into today’s world, from projections of butterflies on Grammy performance dresses to “smart” purses that sense when your wallet is missing. The recent emergence of low-cost, user-friendly components is making this new world of design accessible to a broad community. In this course, students will think critically about products already in the marketplace and will be given the tools to create their own designs. A sequence of hands-on workshops on electronics basics and microcontroller programming will provide the surprisingly minimal level of comfort and background in technology required to produce prototypes of these designs.  
Applies to requirement(s): Math & Sciences  
P. Klemperer  
Credits: 4  

COMSC-201 Advanced Object-Oriented Programming  
Fall and Spring  
This course builds on the basic programming concepts learned in Computer Science 101. Emphasis is on developing the skills needed to write more sophisticated programs. This includes strategies to aid in assuring the correctness of programs through the use of assertions and unit testing as well as advanced Java features such as inheritance, polymorphism, and network programming. We will also introduce some widely used data structures such as vectors and linked lists. This course is programming-intensive.  
Applies to requirement(s): Math & Sciences  
H. Pon-Barry, The department  
Prereq: Computer Science 101 with a grade of C or better.; Coreq: COMSC-201L.  
Credits: 4  

COMSC-211 Data Structures  
Fall and Spring  
Using Java. Solving problems with computers is accomplished by writing programs that operate on data to produce a desired result. The way data is organized and presented to the program can significantly affect its efficiency and simplicity and can sometimes determine whether or not a program can be written to solve the problem at all. This course presents ways of organizing data into ‘data structures’ and analyzes how structuring the data can improve program performance.  
Applies to requirement(s): Math & Sciences  
P. Klemperer, The department  
Prereq: Computer Science 201.  
Credits: 4  

COMSC-215 Software Design  
Fall  
Building large software systems introduces new challenges to software development. Appropriate design decisions early in the development of large software can make a major difference in developing software that is correct and maintainable. In this course, students will learn techniques and tools to help them address these problems and develop larger software projects, improving their skills in designing, writing, debugging, and testing software. Topics include design patterns, UML, designing for maintainability, software architecture, and designing concurrent and fault tolerant systems. Programming intensive.  
Applies to requirement(s): Math & Sciences  
B. Lerner  
Prereq: Computer Science 201.  
Credits: 4  

COMSC-221 Computer System & Assembly Language  
Fall and Spring  
This course looks at the inner workings of a computer and computer systems. It is an introduction to computer architecture. Specific topics include assembly language programming, memory, and parallelism.  
Applies to requirement(s): Math & Sciences  
L. Ballesteros, The department  
Prereq: Computer Science 201; Computer Science 211 also recommended.  
Credits: 4  

COMSC-243 Topic:  
COMSC-243SP Topic: ‘Scientific and Parallel Computing’  
Fall  
How powerful is your smartphone, tablet, or laptop? Can a computer predict earthquakes or simulate climate change? How much processing power does it take to render a 3D movie? Parallelism is ubiquitous in computing, especially scientific computing, as it increases system performance in the face of ever more challenging applications. Parallelism complicates everything from hardware design to operating systems to algorithms and programming. This course examines issues and techniques for parallel programming, the scientific
COMSC-295 Independent Study
Fall and Spring
The department
Instructor permission required.
Credits: 1-4
Course can be repeated for credit.

COMSC-311 Theory of Computation
Fall
Are there any limits to what computers can do? Does the answer to this question depend on whether you use a PC or a Mac? Is C more powerful than PASCAL? This seminar explores these questions by investigating several models of computation, illustrating the power and limitations of each of these models, and relating them to computational problems and applications. Topics include finite state automata, pushdown automata, grammars, Turing machines, the Universal Turing Machine, and computability.
Applies to requirement(s): Math & Sciences
A. St. John
Prereq: Computer Science 201 and Mathematics 232.
Credits: 4

COMSC-312 Algorithms
Spring
How does Mapquest find the best route between two locations? How do computers help to decode the human genome? At the heart of these and other complex computer applications are nontrivial algorithms. While algorithms must be specialized to an application, there are some standard ways of approaching algorithmic problems that tend to be useful in many applications. Among other topics, we will explore graph algorithms, greedy algorithms, divide-and-conquer, dynamic programming, and network flow. We will learn to recognize when to apply each of these strategies as well as to evaluate the expected runtime costs of the algorithms we design.
Applies to requirement(s): Math & Sciences
D. Sheldon
Prereq: Computer Science 201 and Mathematics 232.
Credits: 4

COMSC-316 Software Practicum
Not Scheduled for This Year
Tired of writing programs that nobody ever uses? Then, this is the course for you. Software enables enterprises to carry out previously tedious or impossible tasks, but many organizations lack the resources to develop needed software. You will apply your programming skills to develop and deliver software to meet the requirements of a client from the community. You will learn critical communication skills required to work with a client, work as a team with classmates, and experience the software lifecycle from requirements elicitation through delivery. You will synthesize many topics learned in courses as well as new technologies required to complete the project. Programming intensive.
Applies to requirement(s): Math & Sciences
The department
Instructor permission required.
Prereq: Computer Science 215.
Credits: 4

COMSC-322 Operating Systems
Fall
An introduction to the issues involved in orchestrating the use of computer resources. Topics include operating system evolution, file-handling systems, memory management, virtual memory, resource scheduling, multiprogramming, deadlocks, concurrent processes, protection, and design principles. Course emphasis: understanding the effects of operating system design on computer system performance. This course is programming intensive.
Applies to requirement(s): Math & Sciences
A. DeFlumere
Prereq: Computer Science 201 and Mathematics 232.
Credits: 4

COMSC-331 Computer Graphics
Not Scheduled for This Year
The creation of pictorial images using a computer. Topics include drawing of two- and three-dimensional scenes using OpenGL and other graphical environments; transformations of objects (translations, scalings, rotations, shearings) using homogeneous coordinates; creating perspective in three-dimensional drawing; algorithms for enhancing realism and visual effect; and the mathematical underpinnings of graphic design. Students will complete a number of graphics projects based on readings and class discussion. This course is programming intensive.
Applies to requirement(s): Math & Sciences
The department
Prereq: Computer Science 201 and Mathematics 232, and at least one of the following: Mathematics 203, 211, or 232.
Advisory: Computer Science 211 may be taken concurrently
Credits: 4

COMSC-334 Artificial Intelligence
Not Scheduled for This Year
An introduction to artificial intelligence (AI) research issues in fields such as Natural Language Processing and Machine Learning. This course is programming intensive.
Applies to requirement(s): Math & Sciences
H. Pon-Barry
Prereq: Computer Science 211 and Mathematics 232.
Credits: 4

COMSC-335 Machine Learning
Not Scheduled for This Year
How does Netflix learn what movies a person likes? How do computers read handwritten addresses on packages, or detect faces in images? Machine learning is the practice of programming computers to learn and improve through experience, and it is becoming pervasive in technology and science. This course will cover the mathematical underpinnings, algorithms, and practices that enable a computer to learn. Topics will include supervised learning, unsupervised learning, evaluation methodology, and Bayesian probabilistic modeling. Students will learn to program in MATLAB or Python and apply course skills to solve real world prediction and pattern recognition problems. Programming Intensive.
Applies to requirement(s): Math & Sciences
D. Sheldon
Credits: 4

COMSC-336 Intelligent Information Retrieval
Not Scheduled for This Year
Introduces the basic concepts, methodologies, and research findings in information retrieval. Special topics include Web searching, cross-language retrieval, data mining, and data extraction. Completion of this course will

Computer Science 3
provide the necessary foundation to work in today’s business environment where competitive advantage is obtained by retrieving needed information.

Applies to requirement(s): Math & Sciences

The department

Prereq: Computer Science 211.

Credits: 4

COMSC-341 Topics

COMSC-341PL Topics: 'Programming Language Design and Implementation'

Not Scheduled for This Year

Ever wonder why there are so many semicolons in Java programs, or what it would mean for a language to not be object-oriented? In this course, we will explore issues related to the design and implementation of programming languages. Along the way, we will discover answers to these questions and more. Topics will include syntax, semantics, runtime support for languages as well as an introduction to functional programming.

Applies to requirement(s): Math & Sciences

The department

Prereq: Computer Science 211.

Credits: 4

COMSC-341SN Topics: 'Search Technologies for Data Science'

Fall

The explosion of new unstructured and structured data on the web increases the need for approaches for processing very large data sets. In this course, we will explore tools and techniques such a Map-Reduce and Hadoop in the context of search technologies such as web search (e.g. Google) and recommender systems.

Applies to requirement(s): Math & Sciences

L. Ballesteros

Prereq: Computer Science 211.

Credits: 4

COMSC-395 Independent Study

Fall and Spring

The department

 Instructor permission required.

Credits: 1-8

Course can be repeated for credit.

See Also

- Engineering