Computer Science

Program of Study

The computer science program prepares students for careers in computer science by offering a BA and a BS degree, as well as combined BA/MS and BS/MS degrees. Students who earn the BA have sufficient depth and breadth for either graduate study in computer science or a career in industry. Beyond this, students who earn the BS have substantial depth and breadth in another field, in addition to computer science, by undertaking an approved course of study in a related area. The department also offers a minor. Beyond its direct application within the field, a computer science major or minor serves as an excellent supplementary foundation in other areas, including but not limited to mathematics, the natural sciences, social sciences, public administration, and artistic and creative work.

Program Requirements

Both the BA and BS in computer science require fulfillment of the mathematical sciences requirement in general education by completing an approved two-quarter calculus sequence. The physical sciences requirement in general education must be satisfied by completing an approved two-quarter sequence in either chemistry or physics. Both BA and BS students take at least fourteen computer science courses chosen from an approved program. BS students also take three courses in an approved related field outside computer science.

Advanced Placement

Computer science majors may use AP credit for chemistry or physics to meet their physical sciences requirement in general education or physical science components of the major. However, no credit designated simply as "physical science" (from either AP or the College’s physical sciences examinations) may be used to meet general education requirements or requirements in the computer science majors. No course credit is awarded for AP Computer Science.

Approved Programs

The computer science department counselor is responsible for approval of specific courses and sequences, and responds as needed to changing course offerings in our program and other programs. Students should consult the department counselor for details on specific courses they are considering taking to meet the requirements.

Approved Computer Science Program

For the authoritative version of the Department of Computer Science requirements and course descriptions, visit cs.uchicago.edu.

There is one approved general program comprising required courses in four topic areas, plus four elective computer science courses, used for either the BA or the BS degree. Upper-level or graduate courses in similar topics may be substituted for those on the list that follows with the approval of the department counselor.
Students considering a computer science major are strongly advised to register for an introductory sequence in their first year.

1. Introductory Sequence (three courses required):

   CMSC 15100  Introduction to Computer Science I  100
   or CMSC 16100  Honors Introduction to Computer Science I
   CMSC 15200  Introduction to Computer Science II  100
   or CMSC 16200  Honors Introduction to Computer Science II
   CMSC 15400  Introduction to Computer Systems  100

   Students may only receive credit for one introductory programming sequence: CMSC 10500-10600 Fundamentals of Computer Programming I-II, CMSC 12100-12200 Computer Science with Applications I-II, CMSC 15100-15200 Introduction to Computer Science I-II, or CMSC 16100-16200 Honors Introduction to Computer Science I-II. Exceptions must be approved by the department counselor prior to taking the second sequence.

2. Programming Languages and Systems Sequence (two courses required):

   Two of the following:
   - CMSC 22100  Programming Languages
   - CMSC 22200  Computer Architecture
   - CMSC 22300  Functional Programming
   - CMSC 22610  Implementation of Computer Languages I
   - CMSC 23000  Operating Systems
   - CMSC 23200  Introduction to Computer Security
   - CMSC 23300  Networks and Distributed Systems
   - CMSC 23400  Mobile Computing
   - CMSC 23500  Introduction to Database Systems
   - CMSC 23700  Introduction to Computer Graphics
   - CMSC 23710  Scientific Visualization
   - CMSC 23800  Game Construction

3. Algorithms and Theory Sequence (three courses required):

   Three of the following:
   - CMSC 27100  Discrete Mathematics
   - CMSC 27200  Theory of Algorithms
   - CMSC 28000  Introduction to Formal Languages
   - or CMSC 28100  Introduction to Complexity Theory

   We strongly encourage all majors to complete their Algorithms and Theory courses by the end of their third year.

4. Other Sequences (one two-course sequence required):

   Artificial Intelligence Sequence (two courses required):
   Two of the following:
CMSC 25010  Artificial Intelligence
CMSC 25020  Computational Linguistics
CMSC 25025  Machine Learning and Large-Scale Data Analysis
CMSC 25050  Computer Vision
CMSC 25400  Machine Learning
CMSC 27600  Computational Biology

Advanced Systems Sequence (two courses required):

Two of the following:

- CMSC 22001  Software Construction
- CMSC 22010  Digital Fabrication
- CMSC 22100  Programming Languages
- CMSC 22200  Computer Architecture
- CMSC 22300  Functional Programming
- CMSC 22610  Implementation of Computer Languages I
- CMSC 22620  Implementation of Computer Languages II
- CMSC 23000  Operating Systems
- CMSC 23010  Parallel Computing
- CMSC 23200  Introduction to Computer Security
- CMSC 23300  Networks and Distributed Systems
- CMSC 23310  Advanced Distributed Systems
- CMSC 23400  Mobile Computing
- CMSC 23500  Introduction to Database Systems
- CMSC 23700  Introduction to Computer Graphics
- CMSC 23710  Scientific Visualization
- CMSC 23800  Game Construction

* depending upon what courses the student has taken in the Programming Languages and Systems Sequence (courses may not be used to meet both requirements)

Scientific Computing Sequence (two courses required):

Two of the following:

- CMSC 23710  Scientific Visualization
- CMSC 27610  Digital Biology
- CMSC 28510  Introduction to Scientific Computing

5. Electives (four courses required):

Four additional elective Computer Science courses numbered 20000 or above. A BS student with a double major in a related area may petition to have some of the electives be courses in the other major.
## SUMMARY OF REQUIREMENTS

### GENERAL EDUCATION

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>One of the following sequences:</td>
<td>200</td>
</tr>
<tr>
<td>CHEM 10100 &amp; CHEM 10200</td>
<td></td>
</tr>
<tr>
<td>Introductory General Chemistry I</td>
<td></td>
</tr>
<tr>
<td>and Introductory General Chemistry II (or higher or equivalent)*</td>
<td></td>
</tr>
<tr>
<td>PHYS 13100-13200</td>
<td></td>
</tr>
<tr>
<td>Mechanics; Electricity and Magnetism (or higher) *</td>
<td></td>
</tr>
<tr>
<td>MATH 13100-13200</td>
<td>200</td>
</tr>
<tr>
<td>Elementary Functions and Calculus I-II (or higher) *</td>
<td></td>
</tr>
<tr>
<td>Total Units</td>
<td>400</td>
</tr>
</tbody>
</table>

* Credit may be granted by examination.

### MAJOR

#### Introductory Sequence:

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMSC 15100</td>
<td>Introduction to Computer Science I</td>
</tr>
<tr>
<td>or CMSC 16100</td>
<td>Honors Introduction to Computer Science I</td>
</tr>
<tr>
<td>CMSC 15200</td>
<td>Introduction to Computer Science II</td>
</tr>
<tr>
<td>or CMSC 16200</td>
<td>Honors Introduction to Computer Science II</td>
</tr>
<tr>
<td>CMSC 15400</td>
<td>Introduction to Computer Systems</td>
</tr>
</tbody>
</table>

#### Programming Languages and Systems Sequence (two of the following):

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMSC 22100</td>
<td>Programming Languages</td>
</tr>
<tr>
<td>CMSC 22200</td>
<td>Computer Architecture</td>
</tr>
<tr>
<td>CMSC 22300</td>
<td>Functional Programming</td>
</tr>
<tr>
<td>CMSC 22610</td>
<td>Implementation of Computer Languages I</td>
</tr>
<tr>
<td>CMSC 23000</td>
<td>Operating Systems</td>
</tr>
<tr>
<td>CMSC 23200</td>
<td>Introduction to Computer Security</td>
</tr>
<tr>
<td>CMSC 23300</td>
<td>Networks and Distributed Systems</td>
</tr>
<tr>
<td>CMSC 23400</td>
<td>Mobile Computing</td>
</tr>
<tr>
<td>CMSC 23500</td>
<td>Introduction to Database Systems</td>
</tr>
<tr>
<td>CMSC 23700</td>
<td>Introduction to Computer Graphics</td>
</tr>
<tr>
<td>CMSC 23710</td>
<td>Scientific Visualization</td>
</tr>
<tr>
<td>CMSC 23800</td>
<td>Game Construction</td>
</tr>
</tbody>
</table>

#### Algorithms and Theory Sequence:

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMSC 27100</td>
<td>Discrete Mathematics</td>
</tr>
<tr>
<td>CMSC 27200</td>
<td>Theory of Algorithms</td>
</tr>
<tr>
<td>CMSC 28000</td>
<td>Introduction to Formal Languages</td>
</tr>
<tr>
<td>or CMSC 28100</td>
<td>Introduction to Complexity Theory</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Two courses from an approved sequence</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Four electives numbered CMSC 20000 or above</td>
<td>400</td>
</tr>
<tr>
<td>Plus the following requirements:</td>
<td>0-300</td>
</tr>
<tr>
<td>Degree</td>
<td>Requirements</td>
</tr>
<tr>
<td>--------</td>
<td>--------------</td>
</tr>
<tr>
<td>BA (no other courses required)</td>
<td>BS (3 courses in an approved program in a related field)</td>
</tr>
</tbody>
</table>

**Total Units** 1400-1700

**GRADING**

Computer science majors must take courses in the major for quality grades. A grade of C- or higher must be received in each course in the major. Any 20000-level computer science course taken as an elective beyond requirements for the major may, with consent of instructor, be taken for P/F grading.

Non-majors may take courses either for quality grades or, subject to College regulations and with consent of instructor, for P/F grading. A Pass grade is given only for work of C- quality or higher. Courses fulfilling general education requirements must be taken for quality grades.

Incompletes are typically given in the Department of Computer Science only to students who have done at least 60 percent of the course’s work of a passing quality and who are unable to complete all course work by the end of the quarter. Other restrictions on Incompletes are the province of individual instructors, many of whom do not permit Incompletes. To receive an Incomplete, students must make arrangements in advance with the instructor; a consent form to be signed by the instructor is available from the College adviser.

**HONORS**

Students can earn a BA or BS degree with honors by attaining a grade of B or higher in all courses in the major and a grade of B or higher in three approved graduate computer science courses (30000-level and above). These courses may be courses taken for the major or as electives.

Students may also earn a BA or BS degree with honors by attaining the same minimum B grade in all courses in the major and by writing a successful bachelor’s thesis as part of CMSC 29900 Bachelor’s Thesis. This thesis must be based on an approved research project that is directed by a faculty member and approved by the department counselor.

**RECOMMENDED INTRODUCTORY SEQUENCES IN COMPUTER SCIENCE**

The Department of Computer Science offers different introductory pathways into the program. In consultation with their College adviser and the Computer Science Department advisers, students should choose their introductory courses carefully. Some guidelines follow.

- Students interested in a technical introduction to computer science, without assuming prior experience or unusually strong preparation in mathematics, are encouraged to take CMSC 15100-15200 Introduction to Computer Science I-II.
- Students with programming experience and strong preparation in mathematics should consider CMSC 16100-16200 Honors Introduction to Computer Science I-II.
• Students majoring in quantitative fields other than computer science, including other sciences, mathematics, and economics, should consider CMSC 12100-12200 Computer Science with Applications I-II, possibly followed by CMSC 12300 Computer Science with Applications III.

• Students in the humanities and social sciences may consider CMSC 11000 Multimedia Programming as an Interdisciplinary Art I, CMSC 10500 Fundamentals of Computer Programming I, or CMSC 10600 Fundamentals of Computer Programming II.

• Students interested in only one or two quarters of study should consider CMSC 10500-10600 Fundamentals of Computer Programming I-II or CMSC 12100-12200 Computer Science with Applications I-II. For students intending to pursue advanced study, we recommend CMSC 15100 Introduction to Computer Science I or CMSC 16100 Honors Introduction to Computer Science I as the first course.

• Students who are interested in web design should take CMSC 10100 Introduction to Programming for the World Wide Web I.

• Students may only receive credit for one introductory programming sequence: CMSC 10500-10600 Fundamentals of Computer Programming I-II, CMSC 12100-12200 Computer Science with Applications I-II, CMSC 15100-15200 Introduction to Computer Science I-II, or CMSC 16100-16200 Honors Introduction to Computer Science I-II. Exceptions must be approved by the department counselor prior to taking the second sequence.

• Students who have credit for any of the following courses (or equivalent) may not take CMSC 10200 Introduction to Programming for the World Wide Web II for credit: CMSC 10600 Fundamentals of Computer Programming II, CMSC 12100 Computer Science with Applications I, CMSC 15200 Introduction to Computer Science II, or CMSC 16200 Honors Introduction to Computer Science II.

**Preparation for Graduate Study in Computer Science**

Students interested in continuing their studies beyond the undergraduate level should major in computer science and take as many computer science courses as possible. The following courses are especially relevant:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credit Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMSC 15100</td>
<td>Introduction to Computer Science I</td>
<td>100</td>
</tr>
<tr>
<td>CMSC 15200</td>
<td>Introduction to Computer Science II</td>
<td>100</td>
</tr>
<tr>
<td>CMSC 15400</td>
<td>Introduction to Computer Systems</td>
<td>100</td>
</tr>
<tr>
<td>CMSC 22100</td>
<td>Programming Languages</td>
<td>100</td>
</tr>
<tr>
<td>CMSC 22200</td>
<td>Computer Architecture</td>
<td>100</td>
</tr>
<tr>
<td>CMSC 22610</td>
<td>Implementation of Computer Languages I</td>
<td>100</td>
</tr>
<tr>
<td>CMSC 23000</td>
<td>Operating Systems</td>
<td>100</td>
</tr>
<tr>
<td>CMSC 23300</td>
<td>Networks and Distributed Systems</td>
<td>100</td>
</tr>
<tr>
<td>CMSC 23500</td>
<td>Introduction to Database Systems</td>
<td>100</td>
</tr>
<tr>
<td>CMSC 23700</td>
<td>Introduction to Computer Graphics</td>
<td>100</td>
</tr>
<tr>
<td>CMSC 25400</td>
<td>Machine Learning</td>
<td>100</td>
</tr>
<tr>
<td>CMSC 27100</td>
<td>Discrete Mathematics</td>
<td>100</td>
</tr>
<tr>
<td>CMSC 27200</td>
<td>Theory of Algorithms</td>
<td>100</td>
</tr>
</tbody>
</table>
MINOR PROGRAM IN COMPUTER SCIENCE

The minor in computer science requires seven courses. The introductory sequence of three courses is followed by four approved upper-level courses. Courses in the minor must be taken for quality grades, with a grade of C- or higher in each course. Students may not use AP credit for computer science to meet requirements for the minor.

No courses in the minor can be double counted with the student’s major(s) or with other minors; nor can they be counted toward general education requirements. More than half of the requirements for the minor must be met by registering for courses bearing University of Chicago course numbers. The minor advisor must approve the student’s minor consent form and the student must submit that form to their College adviser by the end of Spring Quarter of their third year.

Introductory Courses

Students must choose three courses from the following (one course from Area A and one course from Area B):

Area A:

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMSC 12100</td>
<td>Computer Science with Applications I</td>
</tr>
<tr>
<td>CMSC 15100</td>
<td>Introduction to Computer Science I</td>
</tr>
<tr>
<td>CMSC 16100</td>
<td>Honors Introduction to Computer Science I</td>
</tr>
</tbody>
</table>

Area B:

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMSC 12200</td>
<td>Computer Science with Applications II</td>
</tr>
<tr>
<td>CMSC 15200</td>
<td>Introduction to Computer Science II</td>
</tr>
<tr>
<td>CMSC 16200</td>
<td>Honors Introduction to Computer Science II</td>
</tr>
<tr>
<td>CMSC 15400</td>
<td>Introduction to Computer Systems</td>
</tr>
</tbody>
</table>

Upper-Level Courses

Four 20000-level or above computer science courses must be approved by the minor advisor. A 20000-level course must replace each 10000-level course in the list above that was used to meet general education requirements.

JOINT BA/MS OR BS/MS PROGRAM

Outstanding computer science majors may apply to complete an MS in computer science along with a BA or BS during their four years at the College. Students must be admitted to the joint MS program.

Prior to applying to either joint program, interested students must meet with the department counselor and Pete Segall, the College BA/MS or BS/MS adviser. (For an appointment with Mr. Segall, call the Office of the Dean of Students at 773.702.8615.)
Students must submit applications for the joint program in the Winter Quarter of their third year.

Participants in the joint BA/MS or BS/MS program must meet the requirements for the BA or BS plus nine courses for the MS and a master’s project. Three of the nine courses for the MS may also be used to meet the requirements of the BA or BS, resulting in a total of 20 courses in computer science. For details visit cs.uchicago.edu/info/BxMS.

**Graduate Courses**

Graduate courses and seminars offered by the Department of Computer Science are open to College students with consent of instructor and department counselor. For more information, consult the department counselor.

**Computer Science Courses**

**CMSC 10100. Introduction to Programming for the World Wide Web I. 100 Units.** This course teaches the basics of building and maintaining a site on the World Wide Web. We discuss Internet terminology and how the Internet and its associated technologies work. Topics include programming websites, hypertext markup language (HTML), Cascading Style Sheets (CSS), and Common Gateway Interface (CGI) scripts (using PERL). Students also learn how to use JavaScript to add client-side functionality.

Instructor(s): W. Sterner
Terms Offered: Winter
Note(s): This course does not meet the general education requirement in the mathematical sciences.

**CMSC 10200. Introduction to Programming for the World Wide Web II. 100 Units.** This course introduces computer programming in Java with a focus on designing and implementing software for the World Wide Web. We first introduce the fundamentals of programming, giving particular attention to basic object-oriented techniques. We employ Java Server Pages to develop programs that interact with users through web browsers. Finally, we study relational databases and, integrating that study with general-purpose Java programming, build database-backed web applications.

Terms Offered: Not offered in 2015-16
Prerequisite(s): Placement into Math 13100 or equivalent; and knowledge of HTML
Note(s): This course meets the general education requirement in the mathematical sciences. May not be taken for credit by students who have credit for CMSC 12100, 15200, or 16200.

**CMSC 10500-10600. Fundamentals of Computer Programming I-II.**

This sequence meets the general education requirement in the mathematical sciences.
CMSC 10500. Fundamentals of Computer Programming I. 100 Units.
This course introduces computer programming using the functional programming language Scheme. We emphasize design, algorithm construction, and procedural/functional/data abstraction.
Terms Offered: Not offered in 2015-16
Prerequisite(s): Placement into Math 13100 or equivalent; or consent of departmental counselor required; previous computer experience and advanced mathematical knowledge not required
Note(s): CMSC 10500 and 10600 may be taken in sequence or individually. This course meets the general education requirement in the mathematical sciences.

CMSC 10600. Fundamentals of Computer Programming II. 100 Units.
This course is an introduction to computer programming using the object-oriented programming language C++. We emphasize algorithm design and construction. Topics include complex types, iteration, recursion, procedural/functional/data abstraction, classes, methods, inheritance, and polymorphism.
Instructor(s): Staff
Terms Offered: Autumn
Prerequisite(s): Placement into Math 13100 or equivalent; or consent of departmental counselor
Note(s): CMSC 10500 and 10600 may be taken in sequence or individually. This course meets the general education requirement in the mathematical sciences.

CMSC 10600. Fundamentals of Computer Programming II. 100 Units.
This course is an introduction to computer programming using the object-oriented programming language C++. We emphasize algorithm design and construction. Topics include complex types, iteration, recursion, procedural/functional/data abstraction, classes, methods, inheritance, and polymorphism.
Instructor(s): Staff
Terms Offered: Autumn
Prerequisite(s): Placement into Math 13100 or equivalent; or consent of departmental counselor
Note(s): CMSC 10500 and 10600 may be taken in sequence or individually. This course meets the general education requirement in the mathematical sciences.

CMSC 11000-11100. Multimedia Programming as an Interdisciplinary Art I-II.
Either course in this sequence meets the general education requirement in the mathematical sciences. Like other classic Chicago general education courses, this sequence provides students with both practical programming skills and core ideas in computer science in interdisciplinary applications. Students learn how to perform in a multi-platform (Mac/Linux/Windows) environment using a high-level prototyping language (revTalk) that allows for the quick creation of useful multimedia applications. As a classic Core course in the Chicago tradition, the course presents introductory techniques of problem solving, algorithm construction, program coding, and debugging as interdisciplinary arts adaptable to a wide range of disciplines with their specialized problems. The first course moves through a sequence from step-by-step introductory labs, to labs that require independent analysis and solution, to a student-designed final project. The second course consists of several scientific and humanistic projects such as Turing Machines, biological modeling, and language manipulation with another final project.
CMSC 11000. Multimedia Programming as an Interdisciplinary Art I. 100 Units.
Either course in this sequence meets the general education requirement in the mathematical sciences. Like other classic Chicago general education courses, this sequence provides students with both practical programming skills and core ideas in computer science in interdisciplinary applications. Students learn how to perform in a multi-platform (Mac/Linux/Windows) environment using a high-level prototyping language (revTalk) that allows for the quick creation of useful multimedia applications. As a classic Core course in the Chicago tradition, the course presents introductory techniques of problem solving, algorithm construction, program coding, and debugging as interdisciplinary arts adaptable to a wide range of disciplines with their specialized problems. The first course moves through a sequence from step-by-step introductory labs, to labs that require independent analysis and solution, to a student-designed final project. The second course consists of several scientific and humanistic projects such as Turing Machines, biological modeling, and language manipulation with another final project.
Instructor(s): W. Sterner Terms Offered: Spring
Prerequisite(s): Placement into Math 13100 or equivalent; or consent of instructor
Note(s): This course meets the general education requirement in the mathematical sciences.

CMSC 11100. Multimedia Programming as an Interdisciplinary Art II. 100 Units.
No description available.
Terms Offered: Not offered in 2015-16
Prerequisite(s): Placement into Math 13100 or equivalent; or consent of instructor
Note(s): This course meets the general education requirement in the mathematical sciences.

CMSC 11710. Networks. 100 Units.
Networks help explain phenomena in such technological, social, and biological domains as the spread of opinions, knowledge, and infectious diseases. Networks also help us understand properties of financial markets, food webs, and web technologies. At the same time, the structure and evolution of networks is determined by the set of interactions in the domain. Our study of networks will employ formalisms such as graph theory, game theory, information networks, and network dynamics, with the goal of building formal models and translating their observed properties into qualitative explanations.
Instructor(s): J. Simon Terms Offered: Spring
Prerequisite(s): Completion of the general education requirement in the mathematical sciences, and familiarity with basic concepts of probability at the high school level.
Note(s): Necessary mathematical concepts will be presented in class.
CMSC 12100-12200-12300. Computer Science with Applications I-II-III.
This three-quarter sequence teaches computational thinking and skills to students who are majoring in the sciences, mathematics, and economics. Lectures cover topics in (1) programming, such as recursion, abstract data types, and processing data; (2) computer science, such as clustering methods, event-driven simulation, and theory of computation; and to a lesser extent (3) numerical computation, such as approximating functions and their derivatives and integrals, solving systems of linear equations, and simple Monte Carlo techniques. Applications from a wide variety of fields serve both as examples in lectures and as the basis for programming assignments. In recent offerings, students have written programs to evaluate betting strategies, determine the number of machines needed at a polling place, and predict the size of extinct marsupials. Students learn Java, Python, R and C++.

CMSC 12100. Computer Science with Applications I. 100 Units.
This three-quarter sequence teaches computational thinking and skills to students who are majoring in the sciences, mathematics, and economics. Lectures cover topics in (1) programming, such as recursion, abstract data types, and processing data; (2) computer science, such as clustering methods, event-driven simulation, and theory of computation; and to a lesser extent (3) numerical computation, such as approximating functions and their derivatives and integrals, solving systems of linear equations, and simple Monte Carlo techniques. Applications from a wide variety of fields serve both as examples in lectures and as the basis for programming assignments. In recent offerings, students have written programs to evaluate betting strategies, determine the number of machines needed at a polling place, and predict the size of extinct marsupials. Students learn Java, Python, R and C++.
Instructor(s): A. Rogers, B. Sotomayor Terms Offered: Autumn
Prerequisite(s): Placement into MATH 15200 or higher, or consent of instructor
Note(s): This course meets the general education requirement in the mathematical sciences.

CMSC 12200. Computer Science with Applications II. 100 Units.
No description available.
Instructor(s): A. Rogers, M. Wachs Terms Offered: Winter
Prerequisite(s): CMSC 12100
Note(s): This course meets the general education requirement in the mathematical sciences.
CMSC 12300. Computer Science with Applications III. 100 Units.
The course revolves around core ideas behind the management and
computation of large volumes of data ("Big Data"). Topics include (1) Statistical
methods for large data analysis, (2) Parallelism and concurrency, including
models of parallelism and synchronization primitives, and (3) Distributed
computing, including distributed architectures and the algorithms and
techniques that enable these architectures to be fault-tolerant, reliable,
and scalable. Students will continue to use R, and will also learn C++ and
distributed computing tools and platforms, including Amazon AWS and
Hadoop. This course includes a project where students will have to formulate
hypotheses about a large dataset, develop statistical models to test those
hypothesis, implement a prototype that performs an initial exploration of the
data, and a final system to process the entire dataset.
Terms Offered: Spring
Prerequisite(s): CMSC 12200

CMSC 12200. Computer Science with Applications II. 100 Units.
No description available.
Instructor(s): A. Rogers, M. Wachs Terms Offered: Winter
Prerequisite(s): CMSC 12100
Note(s): This course meets the general education requirement in the mathematical
sciences.

CMSC 12300. Computer Science with Applications III. 100 Units.
The course revolves around core ideas behind the management and computation of
large volumes of data ("Big Data"). Topics include (1) Statistical methods for large
data analysis, (2) Parallelism and concurrency, including models of parallelism and
synchronization primitives, and (3) Distributed computing, including distributed
architectures and the algorithms and techniques that enable these architectures to be fault-tolerant, reliable, and scalable. Students will continue to use R, and will also
learn C++ and distributed computing tools and platforms, including Amazon AWS and
Hadoop. This course includes a project where students will have to formulate
hypotheses about a large dataset, develop statistical models to test those hypothesis,
implement a prototype that performs an initial exploration of the data, and a final
system to process the entire dataset.
Terms Offered: Spring
Prerequisite(s): CMSC 12200
CMSC 15100-15200. Introduction to Computer Science I-II.
This sequence, which is recommended for all students planning to take more advanced courses in computer science, introduces computer science mostly through the study of programming in functional (Scheme) and imperative (C) programming languages. Topics include program design, control and data abstraction, recursion and induction, higher-order programming, types and polymorphism, time and space analysis, memory management, and data structures including lists, trees, and graphs. NOTE: Non-majors may use either course in this sequence to meet the general education requirement in the mathematical sciences; students who are majoring in Computer Science must use either CMSC 15100-15200 or 16100-16200 to meet requirements for the major.

CMSC 15100. Introduction to Computer Science I. 100 Units.
This sequence, which is recommended for all students planning to take more advanced courses in computer science, introduces computer science mostly through the study of programming in functional (Scheme) and imperative (C) programming languages. Topics include program design, control and data abstraction, recursion and induction, higher-order programming, types and polymorphism, time and space analysis, memory management, and data structures including lists, trees, and graphs. NOTE: Non-majors may use either course in this sequence to meet the general education requirement in the mathematical sciences; students who are majoring in Computer Science must use either CMSC 15100-15200 or 16100-16200 to meet requirements for the major.
Instructor(s): A. Shaw (Aut), M. Wachs (Aut), J. Reppy (Win) Terms Offered: Autumn, Winter, Summer
Prerequisite(s): Placement into MATH 15100 or equivalent, or consent of departmental counselor
Note(s): This course meets the general education requirement in the mathematical sciences.

CMSC 15200. Introduction to Computer Science II. 100 Units.
No description available.
Instructor(s): A. Shaw (Winter), M. Wachs (Winter) Terms Offered: Winter, Spring, Summer
Prerequisite(s): CMSC 15100
Note(s): This course meets the general education requirement in the mathematical sciences.

CMSC 15200. Introduction to Computer Science II. 100 Units.
No description available.
Instructor(s): A. Shaw (Winter), M. Wachs (Winter) Terms Offered: Winter, Spring, Summer
Prerequisite(s): CMSC 15100
Note(s): This course meets the general education requirement in the mathematical sciences.
CMSC 15400. Introduction to Computer Systems. 100 Units.
This course covers the basics of computer systems from a programmer’s perspective. Topics include data representation, machine language programming, exceptions, code optimization, performance measurement, memory systems, and system-level I/O. Extensive programming required.
Instructor(s): H. Gunawi, H. Hoffmann, M. Wachs Terms Offered: Spring
Prerequisite(s): CMSC 15200, 16200 or 12200
Note(s): Required of students who are majoring in Computer Science.

CMSC 16100-16200. Honors Introduction to Computer Science I-II.
Both courses in this sequence meet the general education requirement in the mathematical sciences; students who are majoring in Computer Science must use either CMSC 15200 or 16200 to meet requirements for the major.

CMSC 16100. Honors Introduction to Computer Science I. 100 Units.
Programming in a functional language (currently Haskell), including higher-order functions, type definition, algebraic data types, modules, parsing, I/O, and monads. Basic data structures, including lists, binary search trees, and tree balancing. Basic mathematics for reasoning about programs, including induction, inductive definition, propositional logic, and proofs. Search in graphs, including depth-first and breadth-first search. Search in metric graphs, including greedy and A* search, with applications.
Instructor(s): R. Chugh, S. Kurtz Terms Offered: Autumn
Prerequisite(s): Placement into MATH 16100 or equivalent and programming experience, or consent of department counselor
Note(s): This course meets the general education requirement in the mathematical sciences.
CMSC 16200. Honors Introduction to Computer Science II. 100 Units.
This course emphasizes the C Programming Language, but not in isolation. Instead, C is developed as a part of a larger programming toolkit that includes the shell (specifically ksh), shell programming, and standard Unix utilities (including awk). Nonshell scripting languages, in particular perl and python, are introduced, as well as interpreter (#!) files that use the command-line version of DrScheme. We cover various standard data structures, both abstractly, and in terms of concrete implementations—primarily in C, but also from time to time in other contexts like scheme and ksh. The course uses a team programming approach. There is a mixture of individual programming assignments that focus on current lecture material, together with team programming assignments that can be tackled using any Unix technology. Team projects are assessed based on correctness, elegance, and quality of documentation. We teach the "Unix way" of breaking a complex computational problem into smaller pieces, most or all of which can be solved using pre-existing, well-debugged, and documented components, and then composed in a variety of ways.
Instructor(s): A. Feldman, S. Kurtz Terms Offered: Winter
Prerequisite(s): CMSC 16100, or consent of department counselor
Note(s): Students who have taken CMSC 15100 may take 16200 with consent of instructor. This course meets the general education requirement in the mathematical sciences.

CMSC 16200. Honors Introduction to Computer Science II. 100 Units.
This course emphasizes the C Programming Language, but not in isolation. Instead, C is developed as a part of a larger programming toolkit that includes the shell (specifically ksh), shell programming, and standard Unix utilities (including awk). Nonshell scripting languages, in particular perl and python, are introduced, as well as interpreter (#!) files that use the command-line version of DrScheme. We cover various standard data structures, both abstractly, and in terms of concrete implementations—primarily in C, but also from time to time in other contexts like scheme and ksh. The course uses a team programming approach. There is a mixture of individual programming assignments that focus on current lecture material, together with team programming assignments that can be tackled using any Unix technology. Team projects are assessed based on correctness, elegance, and quality of documentation. We teach the "Unix way" of breaking a complex computational problem into smaller pieces, most or all of which can be solved using pre-existing, well-debugged, and documented components, and then composed in a variety of ways.
Instructor(s): A. Feldman, S. Kurtz Terms Offered: Winter
Prerequisite(s): CMSC 16100, or consent of department counselor
Note(s): Students who have taken CMSC 15100 may take 16200 with consent of instructor. This course meets the general education requirement in the mathematical sciences.
CMSC 22001. Software Construction. 100 Units.
Large software systems are difficult to build. The course discusses both the empirical aspects of software engineering and the underlying theory. Topics will include, among others, software specifications, software design, software architecture, software testing, software reliability, and software maintenance. Students will be expected to actively participate in team projects in this course.
Instructor(s): S. Lu Terms Offered: Spring
Prerequisite(s): CMSC 15400

CMSC 22010. Digital Fabrication. 100 Units.
Digital fabrication involves translation of a digital design into a physical object. While digital fabrication has been around for decades, only now has it become possible for individuals to take advantage of this technology through low cost 3D printers and open source tools for 3D design and modeling. In this course we will cover the foundations of 3D object design including computational geometry, the type of models that can and can't be fabricated, the uses and applications of digital fabrication, the algorithms, methods and tools for conversion of 3D models to representations that can be directly manufactured using computer controlled machines, the concepts and technology used in additive manufacturing (aka 3D printing) and the research and practical challenges of developing self-replicating machines. We will have several 3D printers available for use during the class and students will design and fabricate several parts during the course.
Instructor(s): R. Stevens Terms Offered: Spring
Prerequisite(s): CMSC 15400 and some experience with 3D modeling concepts.

CMSC 22100. Programming Languages. 100 Units.
Programming language design aims at the closest possible correspondence between the structures of a program and the task it performs. This course studies some of the structural concepts affecting programming languages: iterative and recursive control flow, data types and type checking, procedural versus functional programming, modularity and encapsulation, fundamentals of interpreting and compiling, and formal descriptions of syntax and semantics. Students write short programs in different languages to illuminate the variety of possible designs.
Instructor(s): A. Shaw Terms Offered: Spring
Prerequisite(s): CMSC 15400

CMSC 22200. Computer Architecture. 100 Units.
This course is a survey of contemporary computer organization covering CPU design, instruction sets, control, processors, busses, ALU, memory, pipelined computers, multiprocessors, networking, and case studies. We focus on the techniques of quantitative analysis and evaluation of modern computing systems, such as the selection of appropriate benchmarks to reveal and compare the performance of alternative design choices in system design. We emphasize major component subsystems of high-performance computers: pipelining, instruction-level parallelism, memory hierarchies, input/output, and network-oriented interconnections.
Instructor(s): M. Wachs Terms Offered: Autumn
Prerequisite(s): CMSC 15400
CMSC 22300. Functional Programming. 100 Units.
We will explore various aspects of advanced functional programming in this course. Topics will vary from quarter to quarter and may include: untyped and typed programming; pure and impure programming; eager and lazy semantics; "object-functional programming"; functional reactive programming; and concurrent functional programming.
Instructor(s): R. Chugh Terms Offered: Winter
Prerequisite(s): CMSC 15400

CMSC 22311. Functional Systems in Haskell. 100 Units.
Advanced and systems programming in Haskell, including testing, meta-programming, exceptions, concurrency, web, IO, and network programming.
Instructor(s): S. Kurtz Terms Offered: Not offered in 2015-16
Prerequisite(s): CMSC 16100 and CMSC 15400, or consent of instructor

CMSC 22610. Implementation of Computer Languages I. 100 Units.
This course covers principles and techniques for implementing computer languages (e.g., programming languages, query languages, specification languages, domain-specific languages). Topics include lexical analysis, parsing, tree representations of programs (both parse trees and abstract syntax trees), types and type checking, interpreters, abstract machines, and run-time systems. This is a project-based course involving the implementation of a small language using Standard ML.
Instructor(s): J. Reppy Terms Offered: Not offered in 2015-16. Generally offered alternate years.
Prerequisite(s): CMSC 15400 required; CMSC 22100 recommended
Note(s): Prior experience with ML programming not required. This course is offered in alternate years.

CMSC 22620. Implementation of Computer Languages II. 100 Units.
This course is a continuation of CMSC 22610, covering compilers for general-purpose languages. Topics include compiler-immediate representations, continuation-passing style, runtime representations, code generation, code optimization, register allocation, instruction scheduling, and garbage collection. This is a project-based course in which students construct a complete, working compiler for a small language using Standard ML.
Instructor(s): J. Reppy Terms Offered: Not offered in 2015-16. Generally offered alternate years.
Prerequisite(s): CMSC 22610 required; CMSC 22100 strongly recommended
Note(s): Generally offered alternate years.
Equivalent Course(s): CMSC 32620
CMSC 22630. Advanced Implementation of Computer Languages. 100 Units.
This course explores advanced topics in the implementation of high-level programming languages that vary each year (e.g., control-flow analysis algorithms, abstract interpretation, partial evaluation, advanced optimizations, runtime system representations, garbage collection algorithms, foreign-function interfaces). Students are expected to develop both a foundational and applied understanding of these topics.
Instructor(s): J. Reppy Terms Offered: Not offered in 2015-16
Prerequisite(s): CMSC 22100 and 22620, or equivalent

CMSC 23000. Operating Systems. 100 Units.
This course provides an introduction to the basic concepts and techniques used to implement operating systems. Topics include processes and threads, interprocess communication and synchronization, memory management, segmentation, paging, linking and loading, scheduling, file systems, and input/output. The course will revolve around the implementation of an x86 operating system kernel.
Instructor(s): H. Gunawi Terms Offered: Autumn
Prerequisite(s): CMSC 15400, and one of the following: CMSC 22200, CMSC 22610, CMSC 23300, CMSC 23400, CMSC 23500, CMSC 23700, CMSC 23710, or CMSC 23800.

CMSC 23010. Parallel Computing. 100 Units.
This course provides an introduction to the concepts of parallel programming, with an emphasis on programming multicore processors. Topics include: Processes and threads, shared memory, message passing, direct-memory access (DMA), hardware mechanisms for parallel computing, synchronization and communication, patterns of parallel programming. The course will involve a substantial programming project implementing a parallel computations.
Instructor(s): H. Hoffmann Terms Offered: Winter
Prerequisite(s): CMSC 15400 and one of the following: CMSC 22000 (Architecture), CMSC 23000 (Operating Systems), or CMSC 23300 (Networks and Distributed Systems), or consent of the instructor.

CMSC 23200. Introduction to Computer Security. 100 Units.
This course introduces the principles and practice of computer security. It aims to teach how to model threats to computer systems and how to think like a potential attacker. It presents standard cryptographic functions and protocols and gives an overview of threats and defenses for software, host systems, networks, and the Web. It also touches on some of the legal, policy, and ethical issues surrounding computer security in areas such as privacy, surveillance, and the disclosure of security vulnerabilities. The goal of this course is to provide a foundation for further study in computer security and to help better understand how to design, build, and use computer systems more securely.
Instructor(s): A. Feldman Terms Offered: Autumn
Prerequisite(s): CMSC 15400
CMSC 23300. Networks and Distributed Systems. 100 Units.
This course focuses on the principles and techniques used in the development of networked and distributed software. Topics include programming with sockets; concurrent programming; data link layer (Ethernet, packet switching, etc.); internet and routing protocols (UDP, TCP); and other commonly used network protocols and techniques. This is a project-oriented course in which students are required to develop software in C on a UNIX environment.
Instructor(s): B. Sotomayor Terms Offered: Winter
Prerequisite(s): CMSC 15400
Equivalent Course(s): CMSC 33300

CMSC 23310. Advanced Distributed Systems. 100 Units.
This course explores advanced topics in distributed systems. Topics include supercomputing (architectures, applications, programming models, etc.); grid computing with an emphasis on Globus technologies; Infrastructure-as-a-Service clouds (virtual infrastructure management, Amazon EC2, etc.), Platform-as-a-Service clouds (Google App Engine, etc.), and the Software-as-a-Service model; and other current topics related to using and building distributed systems. The course includes a substantial practical component but also requires students to read papers and articles on current advances in the field.
Instructor(s): B. Sotomayor Terms Offered: Spring
Prerequisite(s): CMSC 23300 or consent of instructor
Equivalent Course(s): CMSC 33310

CMSC 23340. Grid Computing. 100 Units.
The new Open Grid Services Architecture (OGSA) defines interfaces and protocols that promise to make it far easier to construct decentralized, dynamic, large-scale systems. We explore and evaluate this technology by using it to develop a range of scalable distributed services. We use the Globus Toolkit, an open source implementation of key OGSA standards, to design and build services. We then evaluate our implementations from the perspectives of performance and programmability.
Instructor(s): I. Foster Terms Offered: Not offered in 2015-16
Prerequisite(s): Substantial programming experience
CMSC 23400. Mobile Computing. 100 Units.
Mobile computing is pervasive and changing nearly every aspect of society. Sensing, actuation, and mediation capabilities of mobile devices are transforming all aspects of computing: uses, networking, interface, form, etc. This course explores new technologies driving mobile computing and their implications for systems and society. Current focus areas include expanded visual experience with computational photography, video and interactive augmented reality, and synchronicity and proximity-detection to enable shared social experiences. Labs expose students to software and hardware capabilities of mobile computing systems, and develop the capability to envision radical new applications for a large-scale course project.
Instructor(s): A. Chien Terms Offered: Winter
Prerequisite(s): CMSC 15200 and 15400 are required and CMSC 23000 or 23300 are recommended. Knowledge of Java is required.
Equivalent Course(s): CMSC 33400

CMSC 23500. Introduction to Database Systems. 100 Units.
This course is an introduction to database design and programming using the relational model. Topics include DBMS architecture, entity-relationship and relational models, relational algebra, relational calculus, functional dependencies and normal forms, web DBs and PHP, query optimization, and physical data organization. The lab section guides students through the collaborative implementation of a relational database management system, allowing students to see topics such as physical data organization and DBMS architecture in practice, and exercise general skills such as collaborative software development.
Instructor(s): A. Elmore Terms Offered: Winter
Prerequisite(s): CMSC 15400

CMSC 23700. Introduction to Computer Graphics. 100 Units.
This course introduces the basic concepts and techniques used in three-dimensional computer graphics. The focus is on real-time rendering techniques, such as those found in computer games. These include coordinate systems and transformations, the graphics pipeline, basic geometric algorithms, texture mapping, level-of detail optimizations, and shadows. Students are required to complete both written assignments and programming projects using OpenGL.
Instructor(s): J. Reppy Terms Offered: Autumn. Generally offered alternate years.
Prerequisite(s): CMSC 15400
Note(s): This course is offered in alternate years.
CMSC 23710. Scientific Visualization. 100 Units.
Scientific visualization combines computer graphics, numerical methods, and mathematical models of the physical world to create a visual framework for understanding and solving scientific problems. The mathematical and algorithmic foundations of scientific visualization (for scalar, vector, and tensor fields) will be explained in the context of real-world data from scientific and biomedical domains. The course is also intended for students outside computer science who are experienced with programming and scientific computing on scientific data. Programming projects will be in C.
Instructor(s): G. Kindlmann Terms Offered: Not offered in 2015-16. Generally offered alternate years.
Prerequisite(s): CMSC 15400 or equivalent linear algebra and programming experience and consent of the instructor.

CMSC 23800. Game Construction. 100 Units.
Computer games are one of the most exciting applications of computer technology. They also are large software systems that embody cutting-edge graphics, as well as techniques from AI, scientific simulation, networking, and databases. This course introduces the student to the basic algorithms and techniques used in computer-game construction. Students work in teams to design and create games using existing libraries for graphics, physics simulation, and so forth.
Instructor(s): J. Reppy Terms Offered: Not offered in 2015-16
Prerequisite(s): CMSC 15400, and at least two of the following courses: CMSC 23700, CMSC 23000, CMSC 23300, CMSC 23500. Strong background in programming and expertise in at least two technical areas underlying computer games (e.g., AI, graphics, scientific computing, networking).
Equivalent Course(s): CSPP 53800

CMSC 23900. Data Visualization. 100 Units.
Data visualizations provide a visual setting in which to explore, understand, and explain data sets. This class describes mathematical and perceptual principles, methods, and applications of "data visualization" (as it is popularly understood to refer to primarily to tabulated data). A range of data types and visual encodings will be presented and evaluated. Visualizations will be primarily web-based, using D3.js, and possibly other higher-level languages and libraries.
Terms Offered: Spring
Prerequisite(s): CMSC 12200, CMSC 15200 or CMSC 16200
CMSC 25020. Computational Linguistics. 100 Units.
This is a course in the Computer Science department, intended for upper-level undergraduates, or graduate students, who have good programming skills. There will be weekly programming assignments in Python. We will look at several current topics in natural language processing, and discuss both the theoretical basis for the work and engaging in hands-on practical experiments with linguistic corpora. In line with most current work, our emphasis will be on systems that draw conclusions from training data rather than relying on the encoding of generalizations obtained by humans studying the data. As a consequence of that, in part, we will make an effort not to focus on English, but to look at a range of human languages in our treatments.
Instructor(s): J. Goldsmith Terms Offered: Spring
Prerequisite(s): CMSC 12200, 15200 or 16200, or by consent
Equivalent Course(s): CMSC 35050, LING 28600, LING 38600

CMSC 25025. Machine Learning and Large-Scale Data Analysis. 100 Units.
This course is an introduction to machine learning and the analysis of large data sets using distributed computation and storage infrastructure. Basic machine learning methodology and relevant statistical theory will be presented in lectures. Homework exercises will give students hands-on experience with the methods on different types of data. Methods include algorithms for clustering, binary classification, and hierarchical Bayesian modeling. Data types include images, archives of scientific articles, online ad clickthrough logs, and public records of the City of Chicago. Programming will be based on Python and R, but previous exposure to these languages is not assumed.
Instructor(s): J. Lafferty Terms Offered: Spring
Prerequisite(s): (STAT 22000 or STAT 23400) and (CMSC 15400 or CMSC 12200), or consent of the instructor
Equivalent Course(s): STAT 37601

CMSC 25050. Computer Vision. 100 Units.
This course covers deformable models for detecting objects in images. Topics include one-dimensional models to identify object contours and boundaries; two-dimensional models for image matching; and sparse models for efficient detection of objects in complex scenes. Mathematical tools needed to define the models and associated algorithms are developed. Applications include detecting contours in medical images, matching brains, and detecting faces in images. Neural network implementations of some of the algorithms are presented, and connections to the functions of the biological visual system are discussed.
Instructor(s): Y. Amit Terms Offered: Spring
Equivalent Course(s): STAT 37900, CMSC 35500
CMSC 25400. Machine Learning. 100 Units.
This course offers a practical, problem-centered introduction to machine learning. Topics covered include the Perceptron and other online algorithms; boosting; graphical models and message passing; dimensionality reduction and manifold learning; SVMs and other kernel methods; and a short introduction to statistical learning theory. Weekly programming assignments give students the opportunity to try out each learning algorithm on real world datasets.
Instructor(s): R. Kondor Terms Offered: Autumn
Prerequisite(s): CMSC 15400. STAT 22000 or STAT 23400 strongly recommended. Equivalent Course(s): STAT 27725

CMSC 27100. Discrete Mathematics. 100 Units.
This course emphasizes mathematical discovery and rigorous proof, which are illustrated on a refreshing variety of accessible and useful topics. Basic counting is a recurring theme and provides the most important source for sequences, which is another recurring theme. Further topics include proof by induction; recurrences and Fibonacci numbers; graph theory and trees; number theory, congruences, and Fermat’s little theorem; counting, factorials, and binomial coefficients; combinatorial probability; random variables, expected value, and variance; and limits of sequences, asymptotic equality, and rates of growth.
Instructor(s): A. Razborov, J. Simon Terms Offered: Autumn
Prerequisite(s): CMSC 15400, or MATH 16300 or higher.
Note(s): This is a directed course in mathematical topics and techniques that is a prerequisite for courses such as CMSC 27200 and 27400.

CMSC 27200. Theory of Algorithms. 100 Units.
This course covers design and analysis of efficient algorithms, with emphasis on ideas rather than on implementation. Algorithmic questions include sorting and searching, discrete optimization, algorithmic graph theory, algorithmic number theory, and cryptography. Design techniques include “divide-and-conquer” methods, dynamic programming, greedy algorithms, and graph search, as well as the design of efficient data structures. Methods of algorithm analysis include asymptotic notation, evaluation of recurrent inequalities, the concepts of polynomial-time algorithms, and NP-completeness.
Instructor(s): A. Drucker Terms Offered: Winter
Prerequisite(s): CMSC 27100 or consent of instructor
CMSC 27410. Honors Combinatorics. 100 Units.
Methods of enumeration, construction, and proof of existence of discrete structures are discussed in conjunction with the basic concepts of probability theory over a finite sample space. Enumeration techniques are applied to the calculation of probabilities, and, conversely, probabilistic arguments are used in the analysis of combinatorial structures. Other topics include basic counting, linear recurrences, generating functions, Latin squares, finite projective planes, graph theory, Ramsey theory, coloring graphs and set systems, random variables, independence, expected value, standard deviation, and Chebyshev's and Chernoff's inequalities.
Instructor(s): L. Babai Terms Offered: Spring
Prerequisite(s): MATH 19900 or 25400, or CMSC 27100, or consent of instructor.
Experience with mathematical proofs.
Note(s): This course is offered in alternate years.

CMSC 27500. Graph Theory. 100 Units.
This course covers the basics of the theory of finite graphs. Topics include shortest paths, spanning trees, counting techniques, matchings, Hamiltonian cycles, chromatic number, extremal graph theory, Turan's theorem, planarity, Menger's theorem, the max-flow/min-cut theorem, Ramsey theory, directed graphs, strongly connected components, directed acyclic graphs, and tournaments. Techniques studied include the probabilistic method.
Instructor(s): K. Mulmuley Terms Offered: Spring
Prerequisite(s): CMSC 15300 or MATH 20400

CMSC 27610. Digital Biology. 100 Units.
Explores the digital nature of biology at the molecular scale. Focuses on the role of hydrophobic effect in protein/ligand associations. Utilizes data-mining as a tool both to understand basic biophysics and to explain protein-ligand associations. Shows how such analog interactions can lead to digital devices (e.g., switches). No biochemistry background will be assumed.
Instructor(s): L. R. Scott Terms Offered: Spring
Prerequisite(s): MATH 15100-15200 and ability to program. All prerequisites will be provided in class.
Note(s): High school chemistry is helpful.

CMSC 27700-27800. Mathematical Logic I-II.
Mathematical Logic I-II

CMSC 27700. Mathematical Logic I. 100 Units.
This course introduces mathematical logic. Topics include propositional and predicate logic and the syntactic notion of proof versus the semantic notion of truth (e.g., soundness, completeness). We also discuss the Gödel completeness theorem, the compactness theorem, and applications of compactness to algebraic problems.
Terms Offered: Autumn
Prerequisite(s): MATH 25400 or 25700; open to students who are majoring in computer science who have taken CMSC 15400 along with MATH 16300 or MATH 15900 or MATH 19900
Equivalent Course(s): MATH 27700
CMSC 27800. Mathematical Logic II. 100 Units.
Topics include number theory, Peano arithmetic, Turing compatibility, unsolvable problems, Gödel’s incompleteness theorem, undecidable theories (e.g., the theory of groups), quantifier elimination, and decidable theories (e.g., the theory of algebraically closed fields).
Terms Offered: Winter
Prerequisite(s): MATH 27700 or equivalent
Equivalent Course(s): MATH 27800

CMSC 27800. Mathematical Logic II. 100 Units.
Topics include number theory, Peano arithmetic, Turing compatibility, unsolvable problems, Gödel’s incompleteness theorem, undecidable theories (e.g., the theory of groups), quantifier elimination, and decidable theories (e.g., the theory of algebraically closed fields).
Terms Offered: Winter
Prerequisite(s): MATH 27700 or equivalent
Equivalent Course(s): MATH 27800

CMSC 28000. Introduction to Formal Languages. 100 Units.
This course is a basic introduction to computability theory and formal languages. Topics include automata theory, regular languages, context-free languages, and Turing machines.
Instructor(s): S. Kurtz Terms Offered: Autumn
Prerequisite(s): CMSC 15400, or MATH 19900 or MATH 25500
Equivalent Course(s): MATH 28000

CMSC 28100. Introduction to Complexity Theory. 100 Units.
Computability topics are discussed (e.g., the s-m-n theorem and the recursion theorem, resource-bounded computation). This course introduces complexity theory. Relationships between space and time, determinism and non-determinism, NP-completeness, and the P versus NP question are investigated.
Instructor(s): K. Mulmuley Terms Offered: Spring
Prerequisite(s): CMSC 27100, or MATH 19900 or 25500; and experience with mathematical proofs
Equivalent Course(s): MATH 28100

CMSC 28501. Topics in Scientific Computing. 100 Units.
This course covers current topics in scientific computing.
Terms Offered: Autumn, Winter, Spring
Prerequisite(s): Consent of instructor

CMSC 29700. Reading and Research in Computer Science. 100 Units.
Students do reading and research in an area of computer science under the guidance of a faculty member. A written report is typically required.
Terms Offered: Summer, Autumn, Winter, Spring
Prerequisite(s): Consent of instructor and approval of department counselor
Note(s): Open both to students who are majoring in Computer Science and to nonmajors. Students are required to submit the College Reading and Research Course Form.
CMSC 29900. Bachelor's Thesis. 100 Units.
No description available.
Terms Offered: Summer, Autumn, Winter, Spring
Prerequisite(s): Consent of instructor and department counselor. Students are required to submit the College Reading and Research Course Form.
Note(s): Open to fourth-year students who are candidates for honors in Computer Science