

Computer Science

Department Website: <https://www.cs.uchicago.edu>

Program of Study

The computer science program prepares students for careers in computer science by offering BA and BS degrees, as well as combined BA/MS and BS/MS degrees. Students who earn the BA are prepared either for graduate study in computer science or a career in industry. Students who earn the BS degree build strength in an additional field by following an approved course of study in a related area. The department also offers a minor. Furthermore, a computer science major or minor serves as an excellent foundation for work in other areas, including but not limited to mathematics, the natural sciences, social sciences, public administration, and the arts.

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.

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 with questions about specific courses they are considering taking to meet the requirements. The department counselor maintains a website with up-to-date program details at <http://major-advising.cs.uchicago.edu>.

Approved Computer Science Program

There is one approved general program for both the BA and BS degrees, comprised of introductory courses, a sequence in Theory, and a sequence in Programming Languages and Systems, followed by advanced electives. Students may substitute upper-level or graduate courses in similar topics for those on the list that follows with the approval of the department counselor. Students who matriculated prior to Autumn 2016 may choose to follow these requirements; otherwise they should consult the archived catalog from their year of matriculation for the degree requirements in computer science. All students who matriculated in Autumn 2016 or later should follow this program.

The course information in this catalog, with respect to who is teaching which course and in which quarter(s), is subject to change during the academic year. For up-to-date information on our course offerings, please consult <http://course-info.cs.uchicago.edu>.

Students considering a computer science major are strongly advised to register for an introductory sequence, starting either with CMSC 15100 or CMSC 16100, in their first year. Incoming students should note that while CMSC 12100 can be used as the first course in the major, it is not open to first-year students, and it is not intended as an entry point for students who already know they want to major in computer science. Students who decide to pursue a computer science major or minor after completing CMSC 12100 may continue with either CMSC 15200-15400 or CMSC 12200-12300-15400. Note that CMSC 12200 does not meet the prerequisites for CMSC 15400.

1. Introductory Sequence (three courses required):

CMSC 15100	Intro To Computer Science-1	100
or CMSC 16100	Honors Introduction to Computer Science I	
or CMSC 12100	Computer Science with Applications I	
CMSC 15200	Intro To Computer Science-2	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.

Students may count only one of CMSC 12100 Computer Science with Applications I, CMSC 15100 Intro To Computer Science-1, and CMSC 16100 Honors Introduction to Computer Science I towards the 4200 units of credit required for graduation. The same is true for CMSC 12200 Computer Science with Applications II, CMSC 15200 Intro To Computer Science-2, and CMSC 16200 Honors Introduction to Computer Science II. Additionally, students who have taken CMSC 15200 or 16200 may not register for CMSC 12100.

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

Three of the following:

CMSC 22001	Software Construction
------------	-----------------------

CMSC 22100	Programming Languages
CMSC 22200	Computer Architecture
CMSC 22300	Functional Programming
CMSC 22600	Compilers for Computer Languages
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

3. Theory Sequence (three courses required):

Two of the following:

CMSC 27100	Discrete Mathematics
or CMSC 27130	Honors Discrete Mathematics
CMSC 27200	Theory of Algorithms
or CMSC 27230	Honors Theory of Algorithms

One of the following:

CMSC 27410	Honors Combinatorics
CMSC 27500	Graph Theory
CMSC 27502	Advanced Algorithms
CMSC 27700	Mathematical Logic-I
CMSC 28000	Introduction to Formal Languages
CMSC 28100	Introduction to Complexity Theory
CMSC 28400	Introduction to Cryptography

The graduate versions of Discrete Mathematics and/or Theory of Algorithms can be substituted for their undergraduate counterparts. We strongly encourage all majors to complete their theory courses by the end of their third year.

4. Electives (five courses required):

The major requires five additional elective Computer Science courses numbered 20000 or above. Students may enroll in CMSC 29700 Reading and Research in Computer Science and CMSC 29900 Bachelor's Thesis for multiple quarters, but only one of each may be counted as a major elective.

Summary of Requirements**

GENERAL EDUCATION		
MATH 13100-13200	Elementary Functions and Calculus I-II (or higher) *	200
One of the following sequences: 200		
CHEM 10100 & CHEM 10200	Introductory General Chemistry I and Introductory General Chemistry II *	
CHEM 11100-11200	Comprehensive General Chemistry I-II *	
CHEM 12100 & CHEM 12200	Honors General Chemistry I and Honors General Chemistry II *	
PHYS 13100-13200	Mechanics; Electricity and Magnetism (or higher) §	
Total Units		400

* Credit may be granted by examination.

§ Students with AP credit for PHYS 12100-12200 may use that to satisfy the general education requirement for the Physical Sciences. However, if a student is fulfilling this requirement through course enrollments, only PHYS 13100-13200 (or higher) will be accepted.

MAJOR

Introductory Sequence (see above)	300
Programming Languages and Systems Sequence (three courses from the list above)	300

Theory Sequence (three courses from the list above)	300
Five electives numbered CMSC 20000 or above §	500
Plus the following requirements:	0-300
BA (no other courses required)	
BS (three courses in an approved program in a related field)	
Total Units	1400-1700

** Students who matriculated prior to Autumn 2016 may choose to follow these requirements; otherwise they should consult the archived catalog from their year of matriculation for the degree requirements in computer science. All students who matriculated in Autumn 2016 or later should follow this program.

§ While a student may enroll in CMSC 29700 or CMSC 29900 for multiple quarters, only one instance of each may be counted toward the major.

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.
- Students interested in only one or two quarters of study should consider CMSC 12100-12200 Computer Science with Applications I-II. For students intending to pursue advanced study, we recommend CMSC 15100 Intro To Computer Science-I or CMSC 16100 Honors Introduction to Computer Science I as the first course.
- Students who are interested in a short pathway to writing simple web software should consider CMSC 10100 Introduction to Programming for the World Wide Web I.
- Students may only receive credit for one introductory programming sequence: 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 Programming-2, CMSC 12100 Computer Science with Applications I, CMSC 15200 Intro To Computer Science-2, or CMSC 16200 Honors Introduction to Computer Science II.

Please be aware that course information is volatile, and the catalog does not necessarily reflect the latest changes. Students should consult <http://course-info.cs.uchicago.edu> for up-to-date information.

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 each from Areas A, B, and C). Please note that not all possible pathways through these courses are valid: for example, CMSC 15200 is not a prerequisite for CMSC 12300. Please consult the prerequisite information below and/or talk to the minor advisor to discuss viable plans.

Area A:		100
CMSC 12100	Computer Science with Applications I	
CMSC 15100	Intro To Computer Science-1	
CMSC 16100	Honors Introduction to Computer Science I	
Area B:		100
CMSC 12200	Computer Science with Applications II	
CMSC 15200	Intro To Computer Science-2	
CMSC 16200	Honors Introduction to Computer Science II	
Area C:		100
CMSC 12300	Computer Science with Applications III	
CMSC 15400	Introduction to Computer Systems	

Upper-Level Courses

The computer science minor must include four courses chosen from among all 20000-level CMSC courses and above. CMSC 12300 may be used as an elective if a student has used CMSC 15400 as the Area C introductory course. 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 undergraduates may apply to complete an MS in computer science along with a BA or BS (generalized to "Bx") during their four years at the College. Students must be admitted to the joint MS program. There are three different paths to a Bx/MS: a research-oriented program for computer science majors, a professionally oriented program for computer science majors, and a professionally oriented program for non-majors.

For full details on these programs, including how to apply, please refer to the relevant section under Joint Degree Programs elsewhere in the catalog.

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 constructing 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 (HTML5), Cascading Style Sheets (CSS3), and Common Gateway Interface (CGI) scripts (using PHP). The primary programming emphasis will be on using 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.

Instructor(s): Staff Terms Offered: TBD

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 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.

CMSC 11000. Multimed Prog: Interdisc 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 framework (LiveCode) that allows for the quick creation of useful "apps" for all platforms - mobile, tablet, and desktop. LiveCode is a free application that can be installed on student computers. 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 especially since programming is an integral part of almost all disciplines. 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 (CMSC 11100) consists of several scientific and humanistic projects such as Turing Machines, biological modeling, and language manipulation with another final project. It will not be offered in academic year 2016-17.)

Instructor(s): W. Sterner Terms Offered: Winter

Prerequisite(s): Placement into MATH 13100 or higher, or by consent.

Note(s): This course meets the general education requirement in the mathematical sciences.

CMSC 11100. Multimedia Programming as an Interdisciplinary Art II. 100 Units.

The second course consists of several scientific and humanistic projects such as Turing Machines, biological modeling, and language manipulation with another final project.

Terms Offered: Spring

Prerequisite(s): Placement into MATH 13100 or higher, or by consent.

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 11800. Introduction to Data Science - I. 100 Units.

Data science provides tools for gaining insight into specific problems using data, through computation, statistics and visualization. This course introduces students to all aspects of a data analysis process, from posing questions, designing data collection strategies, management+storing and processing of data, exploratory tools and visualization, statistical inference, prediction, interpretation and communication of results. Simple techniques for data analysis are used to illustrate both effective and fallacious uses of data science tools. Although this course is designed to be at the level of mathematical sciences courses in the Core, with little background required, we expect the students to develop computational skills that will allow them to analyze data. Computation will be done using Python and Jupyter Notebook.

Instructor(s): Michael J. Franklin, Dan Nicolae Terms Offered: Autumn. Note: This course will be crosslisted with STAT 11800.

Prerequisite(s): None

Equivalent Course(s): STAT 11800

CMSC 11900. Introduction to Data Science II. 100 Units.

This course is the second quarter of a two-quarter systematic introduction to the foundations of data science, as well as to practical considerations in data analysis. A broad background on probability and statistical methodology as well as a basic proficiency in RStudio will be provided. More advanced topics on data privacy and ethics, reproducibility in science, data encryption, and basic machine learning will be introduced. We will explore these concepts with real-world problems from different domains.

Terms Offered: Winter

Equivalent Course(s): STAT 11900

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, etc. 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.

CMSC 12100. Computer Science with Applications I. 100 Units.

This course is the first in a three-quarter sequence that teaches computational thinking and skills to students in the sciences, mathematics, economics, etc. The course will cover abstraction and decomposition, simple modeling, basic algorithms, and programming in Python. 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 simulate a model of housing segregation, determine the number of machines needed at a polling place, and analyze tweets from presidential debates.

Instructor(s): B. Sotomayor, B. Ur Terms Offered: Autumn

Prerequisite(s): Placement into MATH 15100 or completion of MATH 13100.

Note(s): This course meets the general education requirement in the mathematical sciences.

CMSC 12200. Computer Science with Applications II. 100 Units.

This course is the second in a three-quarter sequence that teaches computational thinking and skills to students in the sciences, mathematics, economics, etc. Lectures cover topics in (1) data representation, (2) basics of relational databases, (3) shell scripting, (4) data analysis algorithms, such as clustering and decision trees, and (5) data structures, such as hash tables and heaps. Applications and datasets 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 a course search engine and a system to do speaker identification. Students will program in Python and do a quarter-long programming project.

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 Python, 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 hypotheses, implement a prototype that performs an initial exploration of the data, and a final system to process the entire dataset.

Instructor(s): M. Wachs Terms Offered: Spring

Prerequisite(s): CMSC 12200.

CMSC 12200. Computer Science with Applications II. 100 Units.

This course is the second in a three-quarter sequence that teaches computational thinking and skills to students in the sciences, mathematics, economics, etc. Lectures cover topics in (1) data representation, (2) basics of relational databases, (3) shell scripting, (4) data analysis algorithms, such as clustering and decision trees, and (5) data structures, such as hash tables and heaps. Applications and datasets 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 a course search engine and a system to do speaker identification. Students will program in Python and do a quarter-long programming project.

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 Python, 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 hypotheses, implement a prototype that performs an initial exploration of the data, and a final system to process the entire dataset.

Instructor(s): M. Wachs 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. Intro 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, Summer, Winter

Prerequisite(s): Placement into MATH 15100 or completion of MATH 13100.

Note(s): This course meets the general education requirement in the mathematical sciences. 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 15200. Intro To Computer Science-2. 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. Feldman (Winter), D. Franklin (Winter, Spring), A. Shaw (Winter) Terms Offered: Spring, Summer, Winter

Prerequisite(s): CMSC 15100, CMSC 16100, CMSC 12100, or CMSC 10500.

Note(s): This course meets the general education requirement in the mathematical sciences. 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 15200. Intro To Computer Science-2. 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. Feldman (Winter), D. Franklin (Winter, Spring), A. Shaw (Winter) Terms Offered: Spring, Summer, Winter

Prerequisite(s): CMSC 15100, CMSC 16100, CMSC 12100, or CMSC 10500.

Note(s): This course meets the general education requirement in the mathematical sciences. 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 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 (Spring), H. Hoffmann (Spring), M. Wachs (Autumn, Spring) Terms Offered: Autumn, Spring
Prerequisite(s): CMSC 12100, 15100, or 16100, and CMSC 15200, 16200, or 12300.

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.

Instructor(s): R. Chugh, S. Kurtz Terms Offered: Autumn

Prerequisite(s): Placement into MATH 16100 or equivalent and programming experience, or by consent.

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): F. Chong Terms Offered: Winter

Prerequisite(s): CMSC 16100, or CMSC 15100 and by consent.

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): F. Chong Terms Offered: Winter

Prerequisite(s): CMSC 16100, or CMSC 15100 and by consent.

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 20900. Computers for Learning. 100 Units.

Over time, technology has occupied an increasing role in education, with mixed results. Massive Open Online Courses (MOOCs) were created to bring education to those without access to universities, yet most of the students who succeed in them are those who are already successful in the current educational model. This course covers technology, psychology (e.g., motivation, engagement), and pedagogy (e.g., constructivism) as they apply to educational technology so that students can design and build an educational learning application. Labs focus on developing expertise in technology, and readings supplement lecture discussions on the human components of education.

Instructor(s): D. Franklin Terms Offered: Autumn

Prerequisite(s): CMSC 15400

Equivalent Course(s): MAAD 20900

CMSC 21010. Mathematical Foundations. 100 Units.

This course is an introduction to formal tools and techniques which can be used to better understand linguistic phenomena. A major goal of this course is to enable students to formalize and evaluate theoretical claims.

Equivalent Course(s): LING 31010, CMSC 31010, LING 21010

CMSC 22000. Introduction to Software Development. 100 Units.

Besides providing an introduction to the software development process and the lifecycle of a software project, this course focuses on imparting a number of skills and industry best practices that are valuable in the development of large software projects, such as source control techniques and workflows, issue tracking, code reviews, testing, continuous integration, working with existing codebases, integrating APIs and frameworks, generating documentation, deployment, and logging and monitoring. The course also emphasizes the importance of collaboration in real-world software development, including interpersonal collaboration and team management. The course will be organized primarily around the development of a class-wide software project, with students organized into teams. Collaboration both within and across teams will be essential to the success of the project.

Instructor(s): B. Sotomayor Terms Offered: Spring

Prerequisite(s): CMSC 12200, CMSC 15200, or CMSC 16200

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: Autumn

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: TBD

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 is an introduction to scientific programming language design, whereby design choices are made according to rigorous and well-founded lines of reasoning. The curriculum emphasizes the lambda calculus, type systems, formal semantics, logic and proof, and includes a light introduction to machine assisted formal reasoning. While this course is not a survey of different programming languages, we examine the design decisions embodied by various popular languages when viewed as formal systems.

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): F. Chong (Spring), Y. Li (Autumn) Terms Offered: Autumn, Spring

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 required, CMSC 15100 or CMSC 16100 recommended.

CMSC 22600. Compilers for Computer Languages. 100 Units.

This course covers principles of modern compiler design and implementation. Topics include lexical analysis, parsing, type checking, optimization, and code generation. This is a project oriented course in which students will construct a fully working compiler, using Standard ML as the implementation language.

Instructor(s): J. Reppy Terms Offered: Alternate years. Not offered in 2017-2018.

Prerequisite(s): CMSC 15400 required; CMSC 22100 recommended. (Note: Prior experience with ML programming not required.)

Note(s): This course is offered in alternate years.

CMSC 23000. Operating Systems. 100 Units.

This course provides an introduction to basic Operating System principles and concepts that form as fundamental building blocks for many modern systems from personal devices to Internet-scale services. Basic topics include processes, threads, concurrency, synchronization, memory management, virtual memory, segmentation, paging, caching, process and I/O scheduling, file systems, storage devices. The course will also cover special topics such as journaling/transactions, SSD, RAID, virtual machines, and data-center operating systems. The course project will revolve around the implementation of a mini x86 operating system kernel.

Instructor(s): H. Gunawi Terms Offered: Autumn

Prerequisite(s): CMSC 15400 and one of CMSC 22200, CMSC 22600, CMSC 22610, CMSC 23300, CMSC 23400, CMSC 23500, CMSC 23700, CMSC 27310, or CMSC 23800 strongly recommended.

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 22200, CMSC 23000, CMSC 23300; or by consent.

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 23210. Usable Security and Privacy. 100 Units.

Regardless of how secure a system is in theory, failing to consider how humans actually use the system leads to disaster in practice. This course will examine how to design for security and privacy from a user-centered perspective by combining insights from computer systems, human-computer interaction (HCI), and public policy. We will introduce core security and privacy technologies, as well as HCI techniques for conducting robust user studies. Topics will include usable authentication, user-centered web security, anonymity software, privacy notices, security warnings, and data-driven privacy tools in domains ranging from social media to the Internet of Things. Students will complete weekly problem sets, as well as conduct novel research in a group capstone project. No prior experience in security, privacy, or HCI is required.

Instructor(s): B. Ur Terms Offered: Spring

Prerequisite(s): CMSC 12300 or CMSC 15400.

CMSC 23280. Cryptocurrencies. 100 Units.

This course will cover both the computer science aspects and economic aspects of cryptocurrencies. Topics to be discussed will include network and system building blocks, consensus protocols, cryptographic algorithms, security and privacy issues, pricing of cryptocurrencies, bubbles, monetary policy issues and regulatory concerns.

Instructor(s): D. Cash, H. Uhlig, B. Zhao Terms Offered: Winter

Prerequisite(s): CMSC 10500, 12100, 15100, or 16100 and ECON 10000 (ECON 19800) or ECON 10200 (ECON 19900)
Equivalent Course(s): ECON 23040

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 (IP, IPv6, ARP, etc.); end-to-end 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.

CMSC 23310. Advanced Distributed Systems. 100 Units.

In recent years, large distributed systems have taken a prominent role not just in scientific inquiry, but also in our daily lives. When we perform a search on Google, stream content from Netflix, place an order on Amazon, or catch up on the latest comings-and-goings on Facebook, our seemingly minute requests are processed by complex systems that sometimes include hundreds of thousands of computers, connected by both local and wide area networks. Recent papers in the field of Distributed Systems have described several solutions (such as MapReduce, BigTable, Dynamo, Cassandra, etc.) for managing large-scale data and computation. However, building and using these systems pose a number of more fundamental challenges: How do we keep the system operating correctly even when individual machines fail? How do we ensure that all the machines have a consistent view of the system's state? (And how do we ensure this in the presence of failures?) How can we determine the order of events in a system where we can't assume a single global clock? Many of these fundamental problems were identified and solved over the course of several decades, starting in the 1970s. To better appreciate the challenges of recent developments in the field of Distributed Systems, this course will guide students through seminal work in Distributed Systems from the 1970s, '80s, and '90s, leading up to a discussion of recent work in the field.

Instructor(s): B. Sotomayor Terms Offered: Not offered 2018-2019.

Prerequisite(s): CMSC 23300 with at least a B+, or by consent.

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 new techniques to capture 3d models (depth sensors, stereo vision), drones that enable targeted, adaptive, focused sensing, and new 3d interactive applications (augmented reality, cyberphysical, and virtual reality). 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 15400. CMSC 23000 or 23300 recommended. Knowledge of Java required.

CMSC 23500. Introduction to Database Systems. 100 Units.

This course is an introduction to database design and implementation. Topics include DBMS architecture, entity-relationship and relational models, relational algebra, concurrency control, recovery, indexing, physical data organization, and modern database systems. The lab section guides students through the 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 software systems 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 course covers both the foundations of 3D graphics (coordinate systems and transformations, lighting, texture mapping, and basic geometric algorithms and data structures), and the practice of real-time rendering using programmable shaders. Students are required to complete both written assignments and programming projects using OpenGL.

Instructor(s): J. Reppy Terms Offered: Autumn

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 example, 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 computing with scientific data. Programming projects will be in C and C++.

Instructor(s): G. Kindlmann Terms Offered: Winter

Prerequisite(s): CMSC 15400 and knowledge of linear algebra, or by consent.

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 2018-2019.

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): MPC5 53800

CMSC 23900. Data Visualization. 100 Units.

Data visualizations provide a visual setting in which to explore, understand, and explain datasets. This class describes mathematical and perceptual principles, methods, and applications of "data visualization" (as it is popularly understood to refer 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.

Instructor(s): G. Kindlmann Terms Offered: Spring

Prerequisite(s): CMSC 12200, CMSC 15200 or CMSC 16200.

CMSC 25010. Artificial Intelligence. 100 Units.

This course introduces the theoretical, technical, and philosophical issues of AI. The emphasis is on computational and mathematical modes of inquiry into the structure and function of intelligent systems. Topics include learning and inference, speech and language, vision and robotics, search and reasoning.

Terms Offered: Spring. Not offered 2013-14.

Prerequisite(s): CMSC 15300, CMSC 15400

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): Staff Terms Offered: Spring

Prerequisite(s): CMSC 15400 or CMSC 12200 and STAT 22200 or STAT 23400, or by consent.

Note(s): The prerequisites are under review and may change.

Equivalent Course(s): STAT 37601

CMSC 25050. Topics: 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: Not offered in 2014-15

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; artificial neural networks; 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 or CMSC 12300. 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): S. Kurtz (Winter), J. Simon (Autumn) Terms Offered: Autumn, Winter

Prerequisite(s): CMSC 12300 or CMSC 15400, or MATH 16300 or higher, or by consent.

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 27130. Honors Discrete Mathematics. 100 Units.

We emphasize 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. The honors version of Discrete Mathematics covers topics at a deeper level.

Instructor(s): A. Razborov Terms Offered: Autumn

Prerequisite(s): (CMSC 12300 or CMSC 15400), or MATH 16300 or higher, or by consent.

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, graph algorithms, elementary algorithmic number theory, combinatorial optimization, randomized algorithms, as well as techniques to deal with intractability, like approximation algorithms. 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, amortized analysis, analysis of probabilistic algorithms, the concepts of polynomial-time algorithms, and of NP-completeness.

Instructor(s): J. Simon Terms Offered: Spring, Winter

Prerequisite(s): CMSC 27100 or CMSC 27130 or CMSC 37110, or by consent.

CMSC 27230. Honors 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. The honors version of Theory of Algorithms covers topics at a deeper level.

Instructor(s): A. Drucker Terms Offered: Winter

Prerequisite(s): CMSC 27100 or CMSC 27130 or CMSC 37110 or consent of the 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 15900 or MATH 25400, or CMSC 27100, or by consent. Experience with mathematical proofs.

Note(s): This course is offered in alternate years.

Equivalent Course(s): MATH 28410

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: Not offered 2017-2018.

Prerequisite(s): CMSC 27100, or MATH 20400 or higher.

CMSC 27600. Computational Biology. 100 Units.

This course serves as a general introduction to the basic algorithms used to understand current problems in biology.

Topics may include sequence alignment algorithms to study DNA and protein sequences, algorithms and experiments for protein structure prediction, dynamics, and folding, clustering and machine learning methods for gene expression analysis, computational models of RNA structure, and DNA computing and self-assembly.

Terms Offered: Winter. Generally offered alternate years.

Prerequisite(s): Familiarity with basic discrete mathematics/statistics/algorithms and biology recommended but not required.

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 15200 or higher, and CMSC 12200 or CMSC 15200 or CMSC 16200. High school chemistry helpful.

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. Prerequisite(s): MATH 25400 or MATH 25700 or (CMSC 15400 and (MATH 15910 or MATH 15900 or MATH 19900 or MATH 16300))

Equivalent Course(s): CMSC 27700

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 27900. Chaos, Complexity And Computers. 100 Units.

This course presents the mathematical bases for the complex, scale-independent behavior seen in chaotic dynamics and fractal patterns. It illustrates these principles from physical and biological phenomena. It explores these behaviors concretely using extensive computer simulation exercises, thus developing simulation and data analysis skills. L.

Instructor(s): Staff Terms Offered: Autumn

Prerequisite(s): PHYS 13300 or 14300; PHYS 25000 or prior programming experience.

Equivalent Course(s): PHYS 25100, MATH 29200

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: Spring

Prerequisite(s): CMSC 12300 or CMSC 15400, or MATH 15900 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: Autumn

Prerequisite(s): CMSC 27100, or MATH 15900 or MATH 25500; experience with mathematical proofs.

Equivalent Course(s): MATH 28100

CMSC 28130. Honors 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): Ketan Mulmuley Terms Offered: Autumn

Prerequisite(s): CMSC 27100 or CMSC 27130, or MATH 15900 or MATH 19900 or MATH 25500; experience with mathematical proofs.

CMSC 28400. Introduction to Cryptography. 100 Units.

Cryptography is the use of algorithms to protect information from adversaries. Though its origins are ancient, cryptography now underlies everyday technologies including the Internet, wifi, cell phones, payment systems, and more. This course is an introduction to the design and analysis of cryptography, including how "security" is defined, how practical cryptographic algorithms work, and how to exploit flaws in cryptography. The course will cover algorithms for symmetric-key and public-key encryption, authentication, digital signatures, hash functions, and other primitives. Weekly problem sets will include both theoretical problems and programming tasks. No experience in security is required.

Instructor(s): David Cash Terms Offered: Autumn

CMSC 28510. Introduction to Scientific Computing. 100 Units.

Basic processes of numerical computation are examined from both an experimental and theoretical point of view. This course deals with numerical linear algebra, approximation of functions, approximate integration and differentiation, Fourier transformation, solution of nonlinear equations, and the approximate solution of initial value problems for ordinary differential equations. We concentrate on a few widely used methods in each area covered.

Instructor(s): T. Dupont Terms Offered: Autumn. Generally offered alternate years.

Prerequisite(s): A year of calculus (MATH 15300 or higher), a quarter of linear algebra (MATH 19620 or higher), and CMSC 10600 or higher; or consent of instructor

CMSC 28515. Introduction to Numerical Partial Differential Equations. 100 Units.

This course deals with finite element and finite difference methods for second-order elliptic equations (diffusion) and the associated parabolic and hyperbolic equations. Some methods for solving linear algebraic systems will be used. Scalar first-order hyperbolic equations will be considered.

CMSC 29512. Entrepreneurship in Technology. 100 Units.

The core theme for the Entrepreneurship in Technology course is that computer science students need exposure to the broad challenges of capturing opportunities and creating companies. Most of the skills required for this process have nothing to do with one's technical capacity. We'll explore creating a story, pitching the idea, raising money, hiring, marketing, selling, and more. Real-world examples, case-studies, and lessons-learned will be blended with fundamental concepts and principles. The course will involve a business plan, case-studies, and supplemental reading to provide students with significant insights into the resolve required to take an idea to market. Class discussion will also be a key part of the student experience.

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51043 or 51100

Note(s): Non-MPCS students need to complete a course request form.

Equivalent Course(s): MPCS 51250

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: Autumn, Spring, Summer, Winter

Prerequisite(s): By 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.

Open to fourth-year students who are candidates for honors in Computer Science.

Terms Offered: Autumn Spring Summer Winter

Prerequisite(s): By consent of instructor and approval of department counselor.



Font Notice

This document should contain certain fonts with restrictive licenses. For this draft, substitutions were made using less legally restrictive fonts. Specifically:

Times was used instead of Trajan.

Times was used instead of Palatino.

The editor may contact Leepfrog for a draft with the correct fonts in place.