COMPUTER SCIENCE

Computer Science is the area of study that encompasses all of the theory and practice of computing. The mission of the Department of Computer Science at New Mexico State University is to provide formal education in the core disciplines of computer science, as well as to prepare our graduates for research, development and academic careers. For more information on the Department of Computer Science, please visit the website www.cs.nmsu.edu.

Undergraduate Program Information

The undergraduate computer science program prepares students for graduate study in computer science and for employment in positions involving the design, construction and application of computer systems. Undergraduate degree programs include a Bachelor of Science (ABET accredited), Bachelor of Arts, and four minor degree tracks. The B.S. degree is the traditional computer science degree program, while the B.A. degree offers a more open, flexible degree plan that is easier to combine with studies in other disciplines. The minors offer specialized tracks in algorithm theory, bioinformatics, computer systems and software development. With technology underpinning almost every area of human endeavor today, students across NMSU should consider pursuing a minor or at least taking some computer science courses. Computer science majors should review their programs of study in consultation with their advisors each semester, preferably using the most recent Undergraduate Catalog.

Graduate Program Information

The department offers both Master of Science and Doctor of Philosophy graduate degrees in computer science, along with a Master of Science in Bioinformatics. We also encourage students in other disciplines to do a graduate minor in computer science. Graduate students typically work closely with a faculty member in a specific area of research. The department offers expertise in several research areas, such as: artificial intelligence and knowledge representation; computer and wireless networks; data mining and machine learning; game design and human-computer interaction; bioinformatics; high performance computing; software engineering and programming languages; theory of computing; and assistive technologies.

A number of laboratories have been established to coordinate research activities, including

- the Knowledge representation, Logic and Advanced Programming (KLP) lab;
- the Play and Interactive Experiences for Learning (PiXL) lab;
- the Knowledge Discovery and Data Mining (KDD) lab;
- the Programming Languages, Environments, and Automated Software Engineering (PLEASE) lab;
- the Bioinformatics Research lab;
- the Multimedia Management lab;
- the Cryptography, Privacy and Security Research (CrySPR) lab; and
- the Network and Systems Optimization Lab (NSOL).

Department members are also directing the iCREDITS interdisciplinary Center of Research Excellence in Design of Intelligent Technologies for Smartgrids, offering educational and research opportunities in smartgrids.

Entrance Requirements for Graduate Study in Computer Science

The Graduate Record Exam (GRE) General Test is not required for admission; however, high GRE scores will strengthen a candidate’s application and are highly regarded in the awarding of Graduate Assistantships. To be admitted without undergraduate deficiencies, an entering student must have completed undergraduate preparation substantially equivalent to that required for the Bachelor of Science degree in Computer Science at New Mexico State University; in particular, this includes courses equivalent to

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>C S 172</td>
<td>Computer Science I</td>
<td>4</td>
</tr>
<tr>
<td>C S 271</td>
<td>Object Oriented Programming</td>
<td>4</td>
</tr>
<tr>
<td>C S 272</td>
<td>Introduction to Data Structures</td>
<td>4</td>
</tr>
<tr>
<td>C S 273</td>
<td>Machine Programming and Organization</td>
<td>4</td>
</tr>
<tr>
<td>C S 278</td>
<td>Discrete Mathematics for Computer Science</td>
<td>4</td>
</tr>
<tr>
<td>C S 370</td>
<td>Compilers and Automata Theory</td>
<td>4</td>
</tr>
<tr>
<td>C S 371</td>
<td>Software Development</td>
<td>4</td>
</tr>
<tr>
<td>C S 471</td>
<td>Programming Language Structure I</td>
<td>3</td>
</tr>
<tr>
<td>C S 474</td>
<td>Operating Systems I</td>
<td>3</td>
</tr>
</tbody>
</table>

Deficiencies should be satisfied as early in the student graduate program as possible, through the regular undergraduate courses, the C S 460 - C S 468 transition courses, or through tests administered by faculty members in the relevant areas. Students should consult with their Graduate Advisor to address issues related to deficiencies. Deficiencies are also assigned to applicants whose transcripts denote low grades in selected areas. Admission is often denied to candidates with little background in Computer Science. Instructions for prospective applicants can be found at https://www.cs.nmsu.edu.

Entrance Requirements for Graduate Study in Bioinformatics

The Graduate Record Exam (GRE) General Test is not required for admission; however, high GRE scores will strengthen a candidate’s application and are highly regarded in the awarding of Graduate Assistantships. Students wishing to enroll in the Master program in Bioinformatics must meet the following criteria:

1. Hold a BS degree, from an accredited institution of higher learning, in either a computational field (e.g., Computer Science) or in life sciences (preferably Biology, Biochemistry, or Environmental Sciences)

2. Hold a minimum grade point average of 3.2

Applicants will be expected to provide a Career statement, motivating the interest in bioinformatics and a minimum of three letters of reference.

Graduate Assistantships

Graduate assistantships (in the form of Teaching and Research assistantships) are expected to be available during the academic year. Inquiries should be addressed to the departmental Graduate Committee. Research assistantships are available at the discretion of individual research project leaders in the Department or elsewhere on campus. Submitting detailed vitae, letters of reference, and GRE test scores are encouraged when applying for any assistantship.
Degrees for the Department

Bachelor Degree(s)
- Computer Science - Bachelor of Arts
- Computer Science - Bachelor of Science
- Computer Science (Algorithm Theory) - Bachelor of Science
- Computer Science (Artificial Intelligence) - Bachelor of Science
- Computer Science (Big Data and Data Science) - Bachelor of Science
- Computer Science (Computer Networking) - Bachelor of Science
- Computer Science (Cybersecurity) - Bachelor of Science
- Computer Science (Human Computer Interaction) - Bachelor of Science
- Computer Science (Software Development) - Bachelor of Science
- Cybersecurity - Bachelor of Science

Bachelor-Master Degree(s)
- Computer Science - Bachelor of Science/Master of Science

Master Degree(s)
- Bioinformatics - Master of Science
- Computational Data Analytics - Master of Computational Data Analytics

Doctoral Degree(s)
- Computer Science - Doctor of Philosophy

Minors for the Department

Undergraduate
A student cannot earn more than one of the undergraduate minors unless they pass at least 6 credits in the second minor beyond the requirements of the first minor. The maximum number of undergraduate minors that a student may earn is two. Most courses for the minors listed below have prerequisites. Please check the undergraduate catalog for individual course prerequisites. Students interested in pursuing a computer science minor are encouraged to pick up more information at the departmental office.

- Algorithm Theory - Undergraduate Minor
- Bioinformatics - Undergraduate Minor
- Computer Systems - Undergraduate Minor
- Software Development - Undergraduate Minor

Graduate

- Computer Science - Graduate Minor

Faculty
Professor Son Tran, Department Head

Professors Cook, Pontelli, Misra, Song, Tran; Associate Professors Cao, Pivkina, Toups; Assistant Professors Hamilton, Le, Nagarkar, Vishwanathan, Wang; College Professors Cooper, Steiner

J. Cook, Ph.D. (Colorado)– software engineering, component-based systems; H. Cao, Ph.D. (Hong-Kong)– data mining, databases, data integration; W. Hamilton, Ph.D. (Texas A&M)– media design, online communities, online education, and video game design/culture; T. Le, Ph.D. (Singapore Management University, Singapore)– data mining and machine learning, dimensionality reduction, visualization, topic models, embedding and generative models; S. Misra, Ph.D. (Arizona State)– communication networks, social networks, high performance computing, security and privacy; P. Nagarkar, Ph.D. (Arizona State)– query optimization, indexing, data analytics, big data; I. Pivkina, Ph.D. (Kentucky)– artificial intelligence, computer science education, data mining, E. Pontelli, Ph.D. (New Mexico State)– parallel processing, logic programming, knowledge representation, bioinformatics, assistive technologies; M. Song, Ph.D. (Washington)– statistical computing, systems biology, bioinformatics, computer vision; Z. Toups, Ph.D. (Texas A&M)– digital games, human-computer interaction, mixed reality; S. Tran, Department Head, Ph.D. (Texas-El Paso)– artificial intelligence, knowledge representation, planning, logic programming, non-monotonic reasoning; R. Vishwanathan, Ph.D. (North Texas)– cryptography, theoretical and applied, security, privacy; T. Wang, Ph.D. (University of South Florida)– cyber-physical system security, web security, network security.

College Faculty:

S. Cooper, Ph.D. (New Mexico State)– computer networks; E. Steiner, Ph.D. (Oklahoma State)– computer science education

Computer Science Courses

C S 111. Computer Science Principles
4 Credits (3+2P)
This course provides a broad and exciting introduction to the field of computer science and the impact that computation has today on every aspect of life. It focuses on exploring computing as a creative activity and investigates the key foundations of computing: abstraction, data, algorithms, and programming. It looks into how connectivity and the Internet have revolutionized computing and demonstrates the global impact that computing has achieved, and it reveals how a new student in computer science might become part of the computing future.

Prerequisite(s): MATH 1215 or higher.

C S 117. Introduction to Computer Animation
3 Credits (3)
Introductory course for learning to program with computer animation as well as learning basic concepts in computer science. Students create interactive animation projects such as computer games and learn to use software packages for creating animations in small virtual worlds using 3D models. Recommended for students considering a minor/major in computer science or simply interested in beginning computer animation or programming.

Prerequisite(s): MATH 1215 or higher.

C S 151. C++ Programming
3 Credits (2+2P)
Introduction to object-oriented programming in the C++ language. May be repeated up to 3 credits.

Prerequisite(s): MATH 1215 or higher.

C S 152. Java Programming
3 Credits (2+2P)
Programming in the Java language. May be repeated up to 3 credits.

Prerequisite(s): MATH 1215 or higher.
C S 153. Python Programming I
3 Credits (3)
This course is an introduction to programming in the Python language, covering fundamental scripts, data types and variables, functions, and simple object creation and usage. The focus will be on preparing students to use Python in their own areas. No prior programming experience is required.
Prerequisite(s): MATH 1215 or higher.

C S 154. Python Programming II
3 Credits (3)
This course covers advanced Python programming, including classes, objects, and inheritance, embedded programming in domain applications, database interaction, and advanced data and text processing. The focus will be on preparing students to use Python in their own areas.
Prerequisite(s): C S 153 or C S 453.

C S 157. Topics in Software Programming and Applications
3 Credits (2+2P)
Current topics in computer programming and software applications. Topic announced in the Schedule of Classes. May be repeated if subtitle is different.

C S 158. R Programming I
3 Credits (3)
This course is an introduction to data processing in the R language, covering fundamental script configuration, data types and data collections, R control structures, and basic creation of graphs and data visualizations. This course will not focus on the statistical capabilities of R, though some basic statistical computations will be used.
Prerequisite(s): MATH 1220G.

C S 171G. Introduction to Computer Science
4 Credits (3+2P)
Computers are now used widely in all area of modern life. This course provides understanding of the theoretical and practical foundations for how computers work, and provides practical application and programming experience in using computers to solve problems efficiently and effectively. The course covers broad aspects of the hardware, software, and mathematical basis of computers. Weekly labs stress using computers to investigate and report on data-intensive scientific problems. Practical experience in major software applications includes an introduction to programming, word processing, spreadsheets, databases, presentations, and Internet applications.
Prerequisite(s): MATH 1130G or MATH 1215 or higher.

C S 172. Computer Science I
4 Credits (3+2P)
Computational problem solving; problem analysis; implementation of algorithms using Java. Object-oriented concepts, arrays, searching, sorting, and recursion. May be repeated up to 4 credits. Crosslisted with: C S 460.
Prerequisite(s): (A C or better in either MATH 1250G or MATH 1430G) OR (A C or better in MATH 1220G and a 1 or better in the CS Placement Test).

C S 209. Special Topics
1-3 Credits
May be repeated for a maximum of 12 credits.

C S 271. Object Oriented Programming
4 Credits (3+2P)
Introduction to problem analysis and problem solving in the object-oriented paradigm. Practical introduction to implementing solutions in the C++ language. Pointers and dynamic memory allocation. Hands-on experience with useful development tools. May be repeated up to 4 credits.
Prerequisite(s): At least a C- in C S 172 or E E 112.

C S 272. Introduction to Data Structures
4 Credits (3+2P)
Design, implementation, use of fundamental abstract data types and their algorithms: lists, stacks, queues, deques, trees; imperative and declarative programming. Internal sorting; time and space efficiency of algorithms.
Prerequisite(s): At least a C- in C S 172, or placement.

C S 273. Machine Programming and Organization
4 Credits (3+2P)
Computer structure, instruction execution, addressing techniques; programming in machine and assembly languages. May be repeated up to 4 credits.
Prerequisite(s): At least a C- in C S 172 or E E 112.

C S 274. Discrete Mathematics for Computer Science
4 Credits (3+2P)
Discrete mathematics required for Computer Science, including the basics of logic, number theory, methods of proof, sequences, mathematical induction, set theory, counting, and functions.
Prerequisite(s): At least a C- in C S 172.

C S 343. Algorithm Design & Implementation
3 Credits (3)
Prerequisite(s): At least a C- in C S 272, or consent of instructor.

C S 370. Compilers and Automata Theory
4 Credits (3+2P)
Methods, principles, and tools for programming language processor design; basics of formal language theory (finite automata, regular expressions, context-free grammars); development of compiler components. May be repeated up to 4 credits.
Prerequisite(s): At least a C- in C S 271 and C S 272.

C S 371. Software Development
4 Credits (3+2P)
Software specification, design, testing, maintenance, documentation; informal proof methods; team implementation of a large project.
Prerequisite(s): At least a C- in C S 271 and C S 272.

C S 372. Data Structures and Algorithms
4 Credits (3+2P)
Prerequisite(s): At least a C- in CS 272 and C S 278.
C S 380. Introduction to Cryptography
3 Credits (3)
The course covers basic cryptographic primitives, such as symmetric, public-key ciphers, digital signature schemes, and hash functions, and their mathematical underpinnings. Course helps students understand basic notions of security in a cryptographic sense: chosen plaintext and chosen ciphertext attacks, games, and reductions. Course also covers computational number theory relevant to cryptography. Consent of Instructor required. Crosslisted with: C S 525.
Prerequisite(s): C S 278 (or equivalent) with a C or better.

C S 382. Modern Web Technologies
3 Credits (3)
In this course, we will take a full-stack approach to modern web application design. We will start with the fundamentals including HTML5, CSS3, Javascript, JSON, and the underlying networking concepts and protocols driving the modern web. We will then move on to more advanced topics including javascript backend development with Node.js, NoSQL database design with MongoDB, cloud computing, and responsive web design. Finally, we cover advanced topics including the design and implementation of browser extensions and real-time web technologies like WebRTC and WebSockets. Consent of Instructor required. Crosslisted with: C S 532.

C S 409. Independent Study
1-6 Credits (1-6)
Faculty supervised investigation, to culminate in a written report. May be repeated up to 6 credits.
Prerequisite(s): Written agreement with faculty supervisor.

C S 419. Computing Ethics and Social Implications of Computing
1 Credit (1)
An overview of ethics for computing majors includes: history of computing, intellectual property, privacy, ethical frameworks, professional ethical responsibilities, and risks of computer-based systems. May be repeated up to 1 credits. Graded: S/U Grading (S/U, Audit).
Prerequisite(s): At least a C- in C S 371.

C S 448. Senior Project
4 Credits (4)
Capstone course in which C S majors work in teams and apply computer science skills to complete a large project. May be repeated up to 4 credits. Consent of Instructor required. Restricted to: C S majors.
Prerequisite(s): At least a C- in C S 370 and C S 371.

C S 449. Senior Thesis
4 Credits (4)
Capstone course in which C S majors apply computer science skills to complete a research project, culminating in a written thesis report. May be repeated up to 4 credits. Consent of Instructor required. Restricted to: C S majors.
Prerequisite(s): At least a C- in C S 370 and C S 371.

C S 451. C++ Programming
3 Credits (3)
Programming in the C language. More advanced than C S 151. Recommended for nonmajors only. Not for CS undergraduate students. May be repeated up to 3 credits.

C S 452. Java Programming
3 Credits (2+2P)
Programming in the Java language. More advanced than C S 152. Recommended for nonmajors only. Not for CS undergraduate standing. May be repeated up to 3 credits.

C S 453. Python Programming I
3 Credits (3)
This course is an introduction to programming in the Python language, covering fundamental scripts, data types and variables, functions, and simple object creation and usage. The focus will be on preparing students to use Python in their own areas. No prior programming experience is required. More advanced than C S 153. May be repeated up to 3 credits.

C S 454. Python Programming II
3 Credits (3)
This course covers advanced Python programming, including classes, objects, and inheritance, embedded programming in domain applications, database interaction, and advanced data and text processing. The focus will be on preparing students to use Python in their own areas. For graduate students only. More advanced work than C S 154, and does not count towards CS major requirements. Not for CS undergraduate students. May be repeated up to 3 credits. Restricted to: exclude C S majors.
Prerequisite(s): C S 153 or C S 453.

C S 457. Topics in Software Programming and Applications
3 Credits (2+2P)
Current topics in computer programming and software applications. Topic announced in the Schedule of Classes. More advanced than C S 157. recommended for non-majors only. May be repeated if subtitle is different.
Prerequisite(s): Graduate standing.

C S 458. R Programming I
3 Credits (3)
This course is an introduction to data processing in the R language, covering fundamental script configuration, data types and data collections, R control structures, and basic creation of graphs and data visualizations. This course will not focus on the statistical capabilities of R, though some basic statistical computations will be used. For graduate students only. More advanced work than C S 158. Does not count towards CS major requirements. May be repeated up to 3 credits.
Prerequisite(s): Good understanding of college algebra or higher.

C S 460. Computer Science I Transition
3 Credits (3)
Computational problem solving; problem analysis; implementation of algorithms. Recursive structures and algorithms. For C S graduate students only; cannot be used to meet a C S student’s program of study. Taught with C S 172. May be repeated up to 3 credits. Consent of Instructor required. Crosslisted with: C S 172.

C S 462. Object Oriented Programming Transition
3 Credits (3)
Introduction to problem analysis and problem solving in the object-oriented paradigm. Practical introduction to implementing solutions in the C++ language. Hands-on experience with useful development tools. Cannot be used in a C S student’s program of study. Consent of Instructor required.
Prerequisite(s): At least a C- in C S 172 or C S 460 or consent of instructor.
C S 463. Introduction to Data Structures Transition
3 Credits (3)
Design, implementation, use of fundamental abstract data types and
their algorithms: lists, stacks, queues, deques, trees; imperative and
declarative programming. Internal sorting; time and space efficiency of
algorithms. Cannot be used in a C S student’s program of study. Consent
of Instructor required.
Prerequisite(s): At least a C- in C S 172 or C S 460 or consent of
instructor.

C S 464. Machine Programming and Organization Transition
3 Credits (3)
Computer structure, instruction execution, addressing techniques;
programming in machine and assembly languages. Cannot be used in a C
S student's program of study. Consent of Instructor required.
Prerequisite(s): At least a C- in C S 172 or C S 460 or consent of
instructor.

C S 465. Discrete Math for Computer Science Transition
3 Credits (3)
Logical connectives, sets, functions, relations, graphics, trees, proofs,
induction, and application to computer science. Cannot be used in a C S
student's program of study. Consent of Instructor required.
Prerequisite(s): At least a C- in C S 172 or C S 460 or consent of
instructor.

C S 466. Compilers and Automata Transition
3 Credits (3)
Methods, principles, and tools for programming language processor
design; basics of formal language theory (finite automata, regular
expressions, context-free grammars); development of compiler
components. For C S graduate students only; cannot be used in a
students program of study. May be repeated up to 3 credits. Consent of
Instructor required.
Prerequisite(s): At least a C in (C S 271 or C S 462), in (C S 272 or
C S 463), in (C S 273 or C S 464), or consent of instructor.

C S 468. Software Development Transition
3 Credits (3)
Software specification, design, testing, maintenance, documentation;
informal proof methods; team implementation of a large project. Cannot
be used in a C S student’s program of study. Consent of Instructor
required.
Prerequisite(s): At least a C- in C S 271 or C S 462, in C S 272 or
C S 463, in C S 273 or C S 464, or consent of instructor.

C S 469. Data Structure and Algorithms Transition
3 Credits (3)
Introduction to efficient data structure and algorithm design. Order
notation and asymptotic run-time of algorithms. Recurrence relations and
solutions. Abstract data type dynamic set and red-black trees. Classic
algorithm design paradigms: divide-and-conquer, dynamic programming,
greedy algorithms. Cannot be used in a C S student’s program of study.
Consent of Instructor required. Restricted to: C S majors.
Prerequisite(s): At least a C- in C S 272 or C S 463, in C S 278 or C S 465,
or consent of instructor.

C S 471. Programming Language Structure I
3 Credits (3)
Syntax, semantics, implementation, and application of programming
languages; abstract data types; concurrency. Not for C S graduate
students. May be repeated up to 3 credits.
Prerequisite(s): At least a C- in C S 370 and C S 371.

C S 473. Architectural Concepts I
3 Credits (3)
Comparison of architectures to illustrate concepts of computer
organization; relationships between architectural and software features.
Not for C S graduate students.
Prerequisite(s): At least a C- in C S 273 and C S 370.

C S 474. Operating Systems I
3 Credits (3)
Operating system principles and structures, and interactions with
architectures. Not for C S graduate students.
Prerequisite(s): At least a C- in C S 273, C S 371, and C S 372.

C S 475. Artificial Intelligence I
3 Credits (3)
Fundamental principles and techniques in artificial intelligence
systems. Intelligent Agents; solving problems by searching; local search
techniques; game-playing agents; constraint satisfaction problems;
knowledge representation and reasoning. Further selected topics may
also be covered. Not for C S graduate students.
Prerequisite(s): At least a C- in C S 272 and C S 278.

C S 476. Computer Graphics I
3 Credits (3)
Languages, programming, devices, and data structures for representation
and interactive display of complex objects. Not for C S graduate students.
Prerequisite(s): At least C- in C S 370 or C S 371.

C S 477. Digital Game Design
3 Credits (3)
An introduction to digital game design. Topics include design,
development, and playtesting of games. The course is structured to use
team-based learning. Not for C S graduate students.
Prerequisite(s)/Corequisite(s): C S 371 or consent of instructor.

C S 478. Computer Security
3 Credits (3)
Introduction to the art and science of computer security. Fundamentals
of computer security including elementary cryptography, authentication
and access control, security threats, attacks, detection and prevention in
application software, operating systems, networks and databases. May
be repeated up to 3 credits.
Prerequisite(s): At least a C- in C S 272, C S 373 or consent of instructor.

C S 479. Special Topics
1-12 Credits
Topics announced in the Schedule of Classes. May be repeated under
different subtitles. Not for C S graduate students. May be repeated up to
12 credits.

C S 480. Linux System Administration
3 Credits (3)
Basic system administration for Linux environments. Topics include user
managements, file systems, security, backups, system monitoring, kernel
configuration and other relevant aspects of system administration. Not
for Computer Science graduate students.

C S 481. Visual Programming
3 Credits (3)
Design and implementation of programs using visual (i.e. dataflow or
diagrammatic) programming techniques, with an emphasis on real-time
data processing. Students will learn how to design visual programs,
including how to handle cycles and state maintenance, and will learn to
process audio, video, and other data using visual programs. Not for C S
graduate students. May be repeated up to 3 credits.
Prerequisite(s): At least a C- in C S 272 and C S 278.
C S 482. Database Management Systems I
3 Credits (3)
Database design and implementation; models of database management systems; privacy, security, protection, recovery. Not for C S graduate students.
Prerequisite(s): At least a C- in C S 272 and C S 278.

C S 483. Introduction to Robotics
3 Credits (3)
Basic AI-based robotic architecture and concepts, with an emphasis on building and programming mobile robots. Not for C S graduate students. Consent of Instructor required.
Prerequisite(s): At least a C- in C S 272 and C S 273.

C S 484. Computer Networks I
3 Credits (3)
Fundamental concepts of computer communication networks: layered network architecture, network components, protocol stack and service. Example of application, transport, network and data link layers, protocols primarily drawn from the Internet (TCP, UDP, and IP) protocol multimedia networks; network management and security. Not for C S graduate students. Consent of Instructor required.
Prerequisite(s): At least a C- in C S 272 and CS 273.

C S 485. Human-Centered Computing
3 Credits (3)
Covers iterative, human-centered interface design, including prototyping and evaluation. Basics of graphic design and visualization. Not for C S graduate students.
Prerequisite(s): At least C- in C S 371.

C S 486. Bioinformatics
3 Credits (3)
Introduction to bioinformatics and computational biology. Computational approaches to sequences analysis, protein structure prediction and analysis, and selected topics from current advances in bioinformatics. Not for C S graduate students.
Prerequisite(s): At least a C- in C S 272 and C S 278.

C S 487. Applied Machine Learning I
3 Credits (3)
An introductory course on practical machine learning. An overview of concepts for both unsupervised and supervised learning. Topics include classification, regression, clustering, and dimension reduction. Classical methods and algorithms such as linear regression, neural networks, support vector machines, and ensemble approaches. Recent techniques such as deep learning. Focused on applying of machine learning techniques in application domains. Not for Graduate Majors. Crosslisted with: C S 519.
Prerequisite(s): At least a C- in C S 272, MATH 1511G; or consent of instructor.

C S 488. Introduction to Data Mining
3 Credits (3)
Techniques for exploring large data sets and discovering patterns in them. Data mining concepts, metrics to measure its effectiveness. Methods in classification, clustering, frequent pattern analysis. Selected topics from current advances in data mining. Taught with C S 508.
Prerequisite(s): At least a C- in C S 272 and C S 278.

C S 489. Bioinformatics Programming
3 Credits (3)
Computer programming to analyze high-throughput molecular biology data including genomic sequences, bulk and single-cell transcriptome, epigenome, and other omics data. Quality control, library size normalization, confounding effect removal, clustering, statistical modeling, trajectory inference, and visualization. Taught with C S 509. May be repeated up to 3 credits.

C S 491. Parallel Programming
3 Credits (3)
Programming of shared memory and distributed memory machines; tools and languages for parallel programming; techniques for parallel programming; parallel programming environments. Not for C S graduate students. May be repeated up to 3 credits.
Prerequisite(s): At least a C- in C S 370 or consent of instructor.

C S 493. Algorithm Design and Implementation
3 Credits (3)
This course introduces the basic knowledge of designing classical algorithms and implementing these algorithms using a programming language. In particular, the course teaches various data structures, including graphs and balanced binary search trees, and efficient schemes to implement these data structures. This course also teaches basic algorithm design techniques including divide-and-conquer, greedy scheme, and dynamic programming. This course covers graph algorithms, including graph traversals (depth-first search and breadth-first search), connectivity, shortest paths, and minimum spanning trees. May be repeated up to 3 credits. Graduate standing. Not for CS students. Restricted to: C S majors.
Prerequisite(s): At least a C- in C S 272, or Consent of Instructor.

C S 496. Cloud and Edge Computing
3 Credits (3)
The course presents a top-down view of cloud computing, from applications and administration to programming and infrastructure. Its main focus is on the concepts of networking and parallel programming for cloud computing and large scale distributed systems which form the cloud infrastructure. The topics include: overview of cloud computing, cloud systems, parallel processing in the cloud, distributed storage systems, virtualization, security in the cloud, and multicore operating systems. Students will study state-of-the-art approaches to cloud computing followed by large cloud corporations, namely Google, Amazon, Microsoft, and Yahoo. Students will also apply what they learn through project developments using Amazon Web Services. Not for graduate CS majors. Crosslisted with: C S 522.
Prerequisite(s): At least a C- in C S 372, background in C S 484/504 is preferred or consent of instructor.

C S 502. Database Management Systems I
3 Credits (3)
Database design and implementation; models of database management systems; privacy, security, protection, recovery; requires more advanced graduate work than C S 482. Students are expected to have solid knowledge of data structures and discrete mathematics.

C S 503. Introduction to Robotics
3 Credits (3)
Basic AI-based robotic architectures and concepts, with an emphasis on building and programming mobile robots; requires more advanced graduate work than C S 483. Students are expected to have solid knowledge of data structures and machine-level programming.
C S 504. Computer Networks I  
3 Credits (3)  
Fundamental concepts of computer communication networks: layered network architecture, network components, protocol stack and service. Example of application, transport, network and data link layers, protocols primarily drawn from the Internet (TCP, UDP, and IP) protocol suite; local and wide area networks, wireless and mobile networks, multimedia networks; network management and security; requires more advanced graduate work than C S 484. Students are expected to have solid knowledge of data structures, machine-level programming. Knowledge of statistics (at the level of MATH 371 or MATH 470) is recommended.

C S 505. Artificial Intelligence I  
3 Credits (3)  
Fundamental principles and techniques in artificial intelligence systems. Knowledge representation formalisms; heuristic problem solving techniques; automated logical deduction; robot planning methods; algorithmic techniques for natural language understanding, vision and learning; requires more advanced graduate work than C S 475. Students are expected to have strong knowledge of algorithms and data structures (at the level of C S 372).

C S 506. Computer Graphics I  
3 Credits (3)  
Languages, programming, devices, and data structures for representation and interactive display of complex objects. Requires more advanced graduate work than C S 476. Students are expected to have knowledge of compilers design and software engineering equivalent to CS 370 and CS 371.

C S 508. Introduction to Data Mining  
3 Credits (3)  
Techniques for exploring large data sets and discovering patterns in them. Data mining concepts, metrics to measure its effectiveness. Methods in classification, clustering, frequent pattern analysis. Selected topics from current advances in data mining. Students are expected to have a preparation in Discrete Mathematics and Data Structures equivalent to C S 272 and C S 278. Requires more advanced graduate work than C S 488. Crosslisted with: C S 488.

C S 509. Bioinformatics Programming  
3 Credits (3)  
Computer programming to analyze high-throughput molecular biology data including genomic sequences, bulk and single-cell transcriptome, epigenome, and other omics data. Quality control, library size normalization, confounding effect removal, clustering, statistical modeling, trajectory inference, and visualization. Taught with C S 489. May be repeated up to 3 credits.

C S 510. Automata, Languages, Computability  
3 Credits (3)  
Regular and context-free languages, pushdown and finite-state automata, Turing machines, models of computation, halting problems. Students are expected to have knowledge of compilers design and algorithms equivalent to C S 370 and C S 372. May be repeated up to 3 credits.

C S 513. Computer Security  
3 Credits (3)  
Introduction to the art and science of computer security. Fundamentals of computer security including elementary cryptography, authentication and access control, security threats, attacks, detection and prevention in application software, operating systems, networks and databases. Recommended knowledge of materials in C S 272 and C S 273. May be repeated up to 3 credits.  
Prerequisite(s): At least a C in C S 273 or consent of instructor.

C S 514. Introduction to Smart Grids  
3 Credits (3)  
This course is an introduction to the technologies and design strategies associated with the Smart Grid. The emphasis will be on the development of communications, energy delivery, coordination mechanisms, and management tools to monitor transmission and distribution networks. Topics include: Smart grid introduction and evolution; Power systems; Networking and transport control; Artificial intelligence & agent coordination; Data mining for smart grids. Requires more advanced work than C S 494.  
Prerequisite(s): At least a C- in C S 272 and a C- in E E 201 or 280; or Consent of instructor.

C S 515. Human-Centered Computing  
3 Credits (3)  
Covers iterative, human-centered interface design, including prototyping and evaluation. Basics of graphic design and visualization. Requires more advanced graduate work than C S 485 with an emphasis on studying recent research in human-computer interaction. Students are expected to have knowledge of software engineering equivalent to C S 371.

C S 516. Bioinformatics  
3 Credits (3)  
Introduction to bioinformatics and computational biology. Computational approaches to sequences analysis, protein structure prediction and analysis, and selected topics from current advances in bioinformatics; requires more advanced graduate work than C S 486. Students are expected to have a knowledge of algorithms and data structures equivalent to C S 372 or exposure to Biology (equivalent to BIOL 2310 or BIOL 311).

C S 517. Digital Game Design  
3 Credits (3)  
An introduction to digital game design. Topics include design, development, and playtesting of games. The course is structured to use team-based learning. Requires more advanced graduate work than C S 477 with deeper attention to a team game project.

C S 518. Visual Programming  
3 Credits (3)  
Design and implementation of programs using visual (i.e. dataflow or diagrammatic) programming techniques, with an emphasis on real-time data processing. Students will learn how to design visual programs, including how to handle cycles and state maintenance, and will learn to process audio, video, and other data using visual programs. Requires more advanced graduate work than C S 481. May be repeated up to 3 credits.  
Prerequisite(s): graduate standing.

C S 519. Applied Machine Learning I  
3 Credits (3)  
An introductory course on practical machine learning. An overview of concepts for both unsupervised and supervised learning. Topics include classification, regression, clustering, and dimension reduction. Classical methods and algorithms such as linear regression, neural networks, support vector machines, and ensemble approaches. Recent techniques such as deep learning. Focused on applying of machine learning techniques in application domains. Crosslisted with: C S 487.
C S 521. Parallel Programming
3 Credits (3)
Programming of shared memory and distributed memory machines; tools and languages for parallel programming; parallelizing compilers; parallel programming environments; requires more advanced graduate work than C S 491. Students are expected to have knowledge of programming and machine organization equivalent to C S 271 and C S 273.

C S 522. Cloud and Edge Computing
3 Credits (3)
The course presents a top-down view of cloud computing, from applications and administration to programming and infrastructure. Its main focus is on the concepts of networking and parallel programming for cloud computing and large scale distributed systems which form the cloud infrastructure. The topics include: overview of cloud computing, cloud systems, parallel processing in the cloud, distributed storage systems, virtualization, security in the cloud, and multicore operating systems. Students will study state-of-the-art approaches to cloud computing followed by large cloud corporations, namely Google, Amazon, Microsoft, and Yahoo. Students will also apply what they learn through project developments using Amazon Web Services. Might have additional requirements for graduate students. Crosslisted with: C S 496.
Prerequisite(s): background in C S 484/504 is preferred or consent of instructor.

C S 525. Introduction to Cryptography
3 Credits (3)
The course covers basic cryptographic primitives, such as symmetric, public-key ciphers, digital signature schemes, and hash functions, and their mathematical underpinnings. Course helps students understand basic notions of security in a cryptographic sense: chosen plaintext and chosen ciphertext attacks, games, and reductions. Course also covers computational number theory relevant to cryptography. Consent of Instructor required. Crosslisted with: C S 380.
Prerequisite(s): C S 278 (or equivalent) with a C or better.

C S 531. Principles of Virtual Reality
3 Credits (3)
This course is an introduction to building systems and doing research in virtual reality. We cover system design, development, and evaluation, with an emphasis on recent research in the space. We cover a range of methods, qualitative and quantitative, in order to develop insights into effective VR designs. Students in this class will develop a foundation in VR development; learn about current topics in VR; and design, develop, evaluate, and report on a VR system. Consent of Instructor required. Crosslisted with: C S 381.
Prerequisite(s): C S 485 (or equivalent).

C S 532. Modern Web Technologies
3 Credits (3)
In this course, we will take a full-stack approach to modern web application design. We will start with the fundamentals including HTML5, CSS3, Javascript, JSON, and the underlying networking concepts and protocols driving the modern web. We will then move on to more advanced topics including javascript backend development with Node.js, NoSQL database design with MongoDB, cloud computing, and responsive web design. Finally, we cover advanced topics including the design and implementation of browser extensions and real-time web technologies like WebRTC and WebSockets. Consent of Instructor required. Crosslisted with: C S 382.

C S 570. Analysis of Algorithms
3 Credits (3)
Techniques for design and analysis of algorithms; time and space complexity; proving correctness of programs. Particular algorithms such as sorting, searching, dynamic programming. NP complete problems. Students are expected to have knowledge of algorithms and data structures equivalent to C S 372.

C S 571. Programming Language Structure II
3 Credits (3)
Formal semantics of programming languages. Students are expected to have knowledge of algorithms and data structures equivalent to C S 372, and knowledge of principles of programming languages equivalent to C S 471.

C S 573. Architectural Concepts II
3 Credits (3)
Advanced topics related to computer architecture, guided by the current literature. Students are expected to have knowledge of computer architectures equivalent to C S 473 and of operating systems equivalent to C S 474. Crosslisted with: E E564.

C S 574. Operating Systems II
3 Credits (3)
Advanced topics related to operating system principles, guided by the current literature. Students are expected to have knowledge of computer architectures and operating systems equivalent to C S 473 and C S 474.

C S 575. Artificial Intelligence II
3 Credits (3)
Covers advanced theory and application of artificial intelligence. Concentration on several specific research areas, such as knowledge representation, problem solving, common-sense reasoning, natural language understanding, automated tutoring systems, learning systems. Students are expected to have knowledge of artificial intelligence equivalent to C S 475.

C S 579. Special Topics
1-6 Credits
Topic announced in the Schedule of Classes.

C S 581. Advanced Software Engineering
3 Credits (3)
Advanced tools and methods for developing large software systems. Topics include object-oriented modeling and design, component architectures, templates and generic programming, software configuration and revision control, static and dynamic analysis tools, model, checking, advanced testing, and verification. Students are expected to have knowledge of software engineering equivalent to C S 371.

C S 582. Database Management Systems II
3 Credits (3)
Advanced data models and abstractions, dependencies, implementations, languages, database machines, and other advanced topics. Students are expected to have knowledge of data base management systems equivalent to CS 482.
C S 583. Advanced Cryptography
3 Credits (3)
This is an advanced cryptography course, which will cover cryptographic protocols such as zero-knowledge proofs, secret sharing schemes, secure two-party/multi-party computation, and more. We will also briefly cover real-world applications of these protocols. Students will also be exposed to recent research topics in cryptography (exact topics might vary every offering).
Prerequisite(s): At least a C- in C S 465.

C S 584. Computer Networks II
3 Credits (3)
Advanced topics in computer networks. Covers advanced topics in networking, with emphasis on wireless, and IP networks. Students are expected to have knowledge of computer networks equivalent to C S 484, and of statistics equivalent to MATH 371 or MATH 470.

C S 586. Algorithms in Systems Biology
3 Credits (3)
The course will introduce important algorithms and computational models used in systems biology to study molecular mechanisms for cellular dynamics, processes, and systems. Cellular processes, such as metabolism and signal transduction, are studied as systems and networks quantitatively from high throughput molecular measurements. The topics include molecular biological systems, network alignment, model simulation, network inference, model optimization, and hybrid models. Students will be able to construct models and analyze their properties in the context of molecular biological systems. Students are expected to have knowledge of algorithms and data structures equivalent to C S 372.

C S 587. Advanced Human-Centered Computing
3 Credits (3)
Covers a range of methods available for performing research in the field of human-computer interaction in order to develop insights into effective human-centered designs of computing systems. Students will develop insights into how humans live, play, and work and best practices in system design.
Prerequisite(s): Consent of instructor or at least a C- in one of the following: C S 485/C S 515, SOCI 352, PSYC 430/PSYC 530, or ANTH 520.

C S 589. Special Research Problems
1-6 Credits
Faculty-supervised investigation, to culminate in a written report. May be repeated; maximum of 6 credits may be applied toward M.S. degree. Restricted to majors.
Prerequisite: written agreement with faculty supervisor.

C S 598. Master's Project
1-6 Credits
Project-oriented capstone course to be completed by M.S. students under supervision of their advisor. Maximum of 6 credits may be applied toward M.S. degree. Restricted to C S majors.
Prerequisite: written agreement with instructor.

C S 599. Master's Thesis
1-6 Credits (1-6)
Thesis to be developed by M.S. Students under supervision of their advisor. May be repeated for a maximum of 6 credits. Restricted to majors.
Prerequisite: consent of instructor.

C S 600. Pre-dissertation Research
1-15 Credits
Pre-dissertation research.