Computer Science

Undergraduate Study

The department offers curricula leading to the baccalaureate degree in computer science, a minor in computer science, and participates in curriculum leading to the baccalaureate degree in software engineering.

Major in Computer Science

The curriculum leading to the baccalaureate degree in computer science is designed to prepare students for positions as computer scientists with business, industry, or government, or for graduate study in computer science. The main educational objectives of the computer science program at Iowa State University are that its graduates demonstrate expertise, engagement, and learning within three to five years after graduation.

• Expertise: Graduated students should have the ability to establish peer-recognized expertise in the discipline. They should have the ability to articulate this expertise by formulating and solving problems of interest, by creating or deriving value through the application of technology, and by using mathematical foundations, algorithmic principles, and computer science theory in designing, implementing, and evaluating computer-based systems and processes which meet the desired needs of their employers.

• Engagement: Graduated students should have the ability to be engaged in the profession through the practice of computer science in industry, academia, or the public sector. They should demonstrate effective learning and commitment to working with others by applying communications skills and professional knowledge.

• Learning: Graduated students should have the ability to engage in sustained learning through graduate work, professional improvement opportunities, and self-study so that they can adapt to the role played by information processing in ever-changing areas of science, technology, and society.

Graduate Study

The department offers work for the degrees of Master of Science and Doctor of Philosophy with a major in Computer Science. The Doctor of Philosophy degree may also be earned with computer science as a co-major with some other discipline. Additionally, the department offers a minor to students majoring in other disciplines.

Established research areas include algorithms, artificial intelligence, computational complexity, computer architecture, bioinformatics, computational biology, computer networks, database systems, formal methods, information assurance, machine learning and neural networks, multimedia, operating systems, parallel and distributed computing, programming languages, robotics, and software engineering. There are also numerous opportunities for interdisciplinary research.

Typically, students beginning graduate work in Computer Science have completed a bachelor's degree or equivalent in Computer Science. However, some students with undergraduate majors in other areas, such as Mathematical, physical, or biological science or engineering become successful graduate students in Computer Science.

For the degree Master of Science, a minimum of 31 semester credits is required. A thesis demonstrating research and the ability to organize and express significant ideas in computer science is required.

The purpose of the doctoral program is to train students to do original research in Computer Science. Each student is also required to attain knowledge and proficiency commensurate with a leadership role in the field. The Ph.D. requirements are governed by the student's program of study committee within established guidelines of the department and the graduate college. They include coursework, demonstrated proficiency in four areas of Computer Science, a research skills requirement, a preliminary examination, and a doctoral dissertation and final oral examination. The department recommends that all graduate students majoring in Computer Science teach as part of their training for an advanced degree.

Curriculum in Computer Science

Students wishing to pursue the B.S. degree in computer science must first successfully complete the pre-major program consisting of Com S 227, Com S 228, and Math 165; all with a grade of C- or above.

A student seeking a B.S. degree in computer science must satisfy the requirements of the University and College of Liberal Arts and Sciences (see Liberal Arts and Sciences, Curriculum) and the departmental requirements.

The departmental requirements consist of a minimum of 44 credits in computer science and satisfaction of written and oral requirements. Students must earn at least a C- in each course taken to fulfill the Degree Program.

The following courses are required:

<table>
<thead>
<tr>
<th>Course</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM S 101</td>
<td>Orientation</td>
</tr>
<tr>
<td>COM S 203</td>
<td>Careers in Computer Science</td>
</tr>
<tr>
<td>CPR E 281</td>
<td>Digital Logic</td>
</tr>
<tr>
<td>COM S 227</td>
<td>Introduction to Object-oriented Programming</td>
</tr>
<tr>
<td>COM S 228</td>
<td>Introduction to Data Structures</td>
</tr>
<tr>
<td>COM S 229</td>
<td>Advanced Programming Techniques</td>
</tr>
<tr>
<td>COM S 309</td>
<td>Software Development Practices</td>
</tr>
<tr>
<td>COM S 311</td>
<td>Design and Analysis of Algorithms</td>
</tr>
<tr>
<td>COM S 321</td>
<td>Introduction to Computer Architecture and Machine-Level Programming</td>
</tr>
<tr>
<td>COM S 330</td>
<td>Discrete Computational Structures</td>
</tr>
<tr>
<td>COM S 331</td>
<td>Theory of Computing</td>
</tr>
<tr>
<td>COM S 342</td>
<td>Principles of Programming Languages</td>
</tr>
<tr>
<td>COM S 352</td>
<td>Introduction to Operating Systems</td>
</tr>
<tr>
<td>COM S 362</td>
<td>Object-Oriented Analysis and Design</td>
</tr>
<tr>
<td>or COM S 363</td>
<td>Introduction to Database Management Systems</td>
</tr>
</tbody>
</table>

Total Credits: 38

In addition, students are required to take two 400-level courses from following groups - one from Group B and one from Group B, W, or N:

**Group B** (courses in this group require oral as well as written reports):

<table>
<thead>
<tr>
<th>Course</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM S 401</td>
<td>Projects in Computing Applications</td>
</tr>
<tr>
<td>S E 416</td>
<td>Software Evolution and Maintenance</td>
</tr>
<tr>
<td>(Or CPR E 416)</td>
<td>Software Evolution and Maintenance ((Or S E 416))</td>
</tr>
<tr>
<td>(Or S E 416)</td>
<td>Software Evolution and Maintenance</td>
</tr>
<tr>
<td>COM S 417</td>
<td>Software Testing</td>
</tr>
<tr>
<td>COM S 425</td>
<td>High Performance Computing for Scientific and Engineering Applications</td>
</tr>
<tr>
<td>COM S 430</td>
<td>Advanced Programming Tools</td>
</tr>
<tr>
<td>COM S 437</td>
<td>Computer Game and Media Programming</td>
</tr>
<tr>
<td>COM S 455</td>
<td>Simulation: Algorithms and Implementation</td>
</tr>
<tr>
<td>COM S 461</td>
<td>Principles and Internals of Database Systems</td>
</tr>
<tr>
<td>COM S 472</td>
<td>Principles of Artificial Intelligence</td>
</tr>
<tr>
<td>COM S 474</td>
<td>Elements of Neural Computation</td>
</tr>
</tbody>
</table>

**Group W** (courses in this group require written reports):

<table>
<thead>
<tr>
<th>Course</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM S 418</td>
<td>Introduction to Computational Geometry</td>
</tr>
<tr>
<td>MATH 421</td>
<td>Logic for Mathematics and Computer Science</td>
</tr>
<tr>
<td>MATH 481</td>
<td>Numerical Methods for Differential Equations and Interpolation</td>
</tr>
<tr>
<td>CPR E 431</td>
<td>Basics of Information System Security</td>
</tr>
<tr>
<td>CPR E 458</td>
<td>Real Time Systems</td>
</tr>
<tr>
<td>CPR E 489</td>
<td>Computer Networking and Data Communications</td>
</tr>
</tbody>
</table>

**Group N:**

<table>
<thead>
<tr>
<th>Course</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM S 418</td>
<td>Introduction to Computational Geometry</td>
</tr>
<tr>
<td>MATH 421</td>
<td>Logic for Mathematics and Computer Science</td>
</tr>
<tr>
<td>MATH 481</td>
<td>Numerical Methods for Differential Equations and Interpolation</td>
</tr>
<tr>
<td>CPR E 431</td>
<td>Basics of Information System Security</td>
</tr>
<tr>
<td>CPR E 458</td>
<td>Real Time Systems</td>
</tr>
<tr>
<td>CPR E 489</td>
<td>Computer Networking and Data Communications</td>
</tr>
</tbody>
</table>

Toward satisfying requirements of the College of Liberal Arts and Sciences, the following courses should be included:

<table>
<thead>
<tr>
<th>Course</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHIL 343</td>
<td>Philosophy of Technology</td>
</tr>
<tr>
<td>SP CM 212</td>
<td>Fundamentals of Public Speaking</td>
</tr>
<tr>
<td>14 credits of Math and Statistics</td>
<td></td>
</tr>
</tbody>
</table>
The following courses meet the communication proficiency requirement:

- PHYS 221 or HIGHER
- PSYCH 310
- MTEOR 301
- MTEOR 206 & 215L
- GEOL 451
- GEOL 201
- GEOL 108
- GEOL 105
- GEOL 102
- GEOL 101
- GEOL 100
- GEN 320
- GEN 313L
- GEN 313
- GEN 260
- FS HN 167
- ENT 370
- CHEM 177 & 177L
- CHEM 178 & 178L
- PHYS 221
- PHYS 222
- MATH 304
- MATH 265
- MATH 266
- MATH 304
- MATH 314
- MATH 265
- BIOL 211 & 211L
- BIOL 212 & 212L
- CHEM 177 & 177L
- CHEM 178 & 178L
- BIOL 211 & 211L
- BIOL 355
- CHEM 163
- ENT 370
- ENV S 324
- FS HN 167
- GEN 260
- GEN 313
- GEN 313L
- GEN 320
- GEO 101
- GEO 102
- GEO 105
- GEO 108
- GEO 201
- GEO 451
- MAT E 215 & 215L
- MTEOR 206
- MTEOR 301
- PSYCH 310
- PHYS 221 or HIGHER

The following courses meet the communication proficiency requirement:

- ENGL 150 Critical Thinking and Communication
- ENGL 250 Written, Oral, Visual, and Electronic Composition
- ENGL 302 Business Communication
- ENGL 305 Creative Writing—Nonfiction
- ENGL 309 Report and Proposal Writing
- ENGL 314 Technical Communication

The minimum grade accepted in each of the three required English courses is a C-.

Students transferring from other institutions must take at least 45 credits at the 300 level or higher at Iowa State University while resident here; of these at least 15 credits should be in our department.

Students must earn a C- or better in each course in the department which is a prerequisite to a course listed in the student’s degree program.

Undergraduate Minor in Computer Science

The Department of Computer Science offers an undergraduate minor in Computer Science. The minor requires at least 19 credits in computer science courses.

- COM S 227 Introduction to Object-oriented Programming 4
- COM S 228 Introduction to Data Structures 3
- COM S 229 Advanced Programming Techniques 3

9 credits in courses 300 level or above 9

Undergraduate Curriculum in Software Engineering

The Department of Computer Science together with the Department of Electrical and Computer Engineering also offer a curriculum leading to an undergraduate degree in software engineering. The software engineering curriculum offers emphasis areas in software engineering principles, process, and practice.

Students may also take elective courses in computer engineering and computer science.

See Index, Software Engineering. For curriculum information, see also College of Engineering and College of Liberal Arts and Sciences.

Courses primarily for undergraduates:

- COM S 101. Orientation. Cr. 4. F.S.S.
- Introduction to the procedures and policies of Iowa State University and the Department of Computer Science, test-outs, honorary societies, etc. Issues relevant to student adjustment to college life will also be discussed. Offered on a satisfactory-fail basis only.

- COM S 103. Computer Applications. Cr. 4. F.S.S.S.
- Introduction to computer literacy and applications. Applications: Windows, Internet browser/HTML, word processing, spreadsheets, database management and presentation software. Literature: history of computing, structure of computers, telecommunications, computer ethics, computer crime, and history of programming languages. No prior computer experience necessary. Course is offered online only. Students must attend an orientation session the first week of class.

- COM S 104. Introduction to Computers. (1.5-1) Cr. 2. F.S.
- Offered first 8 weeks and last 8 weeks. Use of personal computer and workstation operating systems and beginning programming. Project-oriented approach to computer operation and programming, including use of tools to aid in programming. Topics from computer history, using basic Windows and Unix tools, program structure, expression, variables, decision and logic, and iteration. No prior computer experience necessary.

- COM S 107. Applied Computer Programming. (3-0) Cr. 3. F.S.
- Introduction to computer programming for non-majors using a language such as the Visual Basic language. Basics of good programming and algorithm development. Graphical user interfaces.

- COM S 201. Computer Programming in COBOL. (3-0) Cr. 3. S.S.
- Computer programming in COBOL. Emphasis on the design, writing, debugging, and testing of business applications programs in a transaction-oriented environment.
COM S 203. Careers in Computer Science.  
Cr. R. F.S.  
Computer science as a profession. Introduction to career fields open to computer science majors. Relationship of coursework to careers. Presentations by computer science professionals. Offered on a satisfactory-fail basis only.

(Cross-listed with MIS). (3-1) Cr. 3. F.S. Prereq: MATH 150 or placement into MATH 140/MATH 141/MATH 142 or higher  
An introduction to computer programming using an object-oriented programming language. Emphasis on the basics of good programming techniques and style. Extensive practice in designing, implementing, and debugging small programs. Use of abstract data types. Interactive and file I/O. Exceptions/error-handling. This course is not designed for computer science, software engineering, and computer engineering majors. Credit may not be applied toward graduation for both Com S 207/MIS 207 and Com S 227.

(3-1) Cr. 3. S. Prereq: MATH 150 or placement into MATH 151, MATH 160, or MATH 165  
Intermediate-level programming techniques. Emphasis on designing, writing, testing, debugging, and documenting medium-sized programs. Data structures and their uses. Dynamic memory usage. Inheritance and polymorphism. Algorithm design and efficiency: recursion, searching, and sorting. Event-driven and GUI programming. The software development process. This course is not designed for computer science, software engineering and computer engineering majors. Credit may not be applied toward the major in computer science, software engineering, or computer engineering.

COM S 227. Introduction to Object-oriented Programming.  
(3-2) Cr. F.S.  
An introduction to object-oriented design and programming techniques. Symbolic and numerical computation. Recursion and iteration. Modularity procedural and data abstraction, specifications and subtyping. Object-oriented techniques. Imperative programming. Emphasis on principles of programming and object-oriented design through extensive practice in design, writing, running, debugging, and reasoning about programs. This course is designed for majors. Credit may not be applied toward graduation for both Com S 207 and 227.

COM S 228. Introduction to Data Structures.  
(3-1) Cr. F.S. Prereq: C- or better in 227, credit or enrollment in MATH 165  
An object-oriented approach to data structures and algorithms. Object-oriented analysis, design, and programming, with emphasis on data abstraction, inheritance and subtype polymorphism. Abstract data type specification and correctness. Collections and associated algorithms, such as stacks, queues, lists, trees. Searching and sorting algorithms. Graphs. Data on secondary storage, analysis of algorithms. Emphasis on object-oriented design, writing and documenting medium-sized programs. This course is designed for majors.

(3-0) Cr. F.S. Prereq: COM S 228, credit or enrollment in MATH 166  
Object-oriented programming experience using a language suitable for exploring advanced topics in programming. Topics include memory management, parameter passing, inheritance, compiling, debugging, and maintaining programs. Significant programming projects.

COM S 252. Linux Operating System Essentials.  
(3-0) Cr. F.S. Prereq: COM S 107 or COM S 207 or COM S 227  
Introduction to installation, utilization, and administration of Linux systems. Topics include open-source software, package installation and management, shell programming and command-line utilities, process and service management, account management, network configuration, file sharing, interoperation with other computers and operating systems, automation, and system security.

COM S 290. Independent Study.  
Cr. arr. F.S. Prereq: Permission of instructor  
Offered on a satisfactory-fail basis only.

COM S 290H. Independent Study: Honors.  
Cr. arr. F.S. Prereq: Permission of instructor  
Offered on a satisfactory-fail basis only.

(3-1) Cr. F.S. Prereq: COM S 228 with C- or better  
A practical introduction to methods for managing software development. Process models, requirements analysis, structured and object-oriented design, coding, testing, maintenance, cost and schedule estimation, metrics. Programming projects. Nonmajor graduate credit.

COM S 311. Design and Analysis of Algorithms.  
(3-1) Cr. F.S. Prereq: COM S 228 with C- or better, MATH 166, ENGL 250, and COM S 330 or CPR E 310  
Basic techniques for design and analysis of efficient algorithms. Sorting, searching, graph algorithms, computational geometry, string processing and NP-completeness. Design techniques such as dynamic programming and the greedy method. Asymptotic, worst-case, average-case and amortized analyses. Data structures including heaps, hash tables, binary search trees and red-black trees. Programming projects. Nonmajor graduate credit.

COM S 319. Software Construction and User Interfaces.  
(Cross-listed with ECE). (3-0) Cr. F. Prereq: COM S 228  

(3-1) Cr. F.S. Prereq: C- or higher in COM S 228, CPR E 281 and ENGL 250  
Introduction to computer architecture and organization. Emphasis on evaluation of performance, instruction set architecture, datapath and control, memory-hierarchy design, and pipelining. Assembly language on a simulator. Nonmajor graduate credit.

(3-1) Cr. F.S. Prereq: C- or higher in COM S 228, C- or higher in MATH 166 and ENGL 250  
Concepts in discrete mathematics as applied to computer science. Logic, proof techniques, set theory, relations, graphs, combinators, discrete probability and number theory. Nonmajor graduate credit.

(Cross-listed with LING). (3-1) Cr. F.S. Prereq: C- or higher in COM S 228, C- or higher in COM S 330 or CPR E 310, C- or higher in MATH 166, and ENGL 250  

(3-0) Cr. F. Prereq: COM S 229. Coreq MATH 307 or MATH 317  
Basic algorithms, design, and programming of interactive computer graphics systems and hardware. Topics include 2D and 3D transformations, 3D viewing, visible surface algorithms, collision detection, illumination models, shading, ray tracing, shadows, transparency and texture mapping.

COM S 342. Principles of Programming Languages.  
(Cross-listed with S E). (3-1) Cr. F.S. Prereq: C- or better in COM S 228, COM S 330 or CPR E 310  
Study of concepts in programming languages and major programming paradigms, especially functional programming. Special emphasis on design tradeoffs that enable students to make sound choices of programming languages for a given software development task. Programming projects. Nonmajor graduate credit.

(Cross-listed with MATH). (3-0) Cr. 3. S. Prereq: MATH 166  
Divisibility, integer representations, primes and divisors, linear diophantine equations, congruences, and multiplicative functions. Applications to cryptography. Nonmajor graduate credit.

COM S 352. Introduction to Operating Systems.  
(3-1) Cr. F.S. Prereq: COM S 229, and COM S 321; ENGL 250  
Survey of operating system issues. Introduction to hardware and software components including: processors, peripherals, interrupts, management of processes, threads and memory, deadlocks, file systems, protection, virtual machines and system organization, and introduction to distributed operating systems. Programming projects. Nonmajor graduate credit.

COM S 362. Object-Oriented Analysis and Design.  
(3-0) Cr. F.S. Prereq: COM S 228 with C- or better, ENGL 250  
Object-oriented requirements analysis and systems design. Design notations such as the Unified Modeling Language. Design Patterns. Group design and programming with large programming projects. Nonmajor graduate credit.
COM S 363. Introduction to Database Management Systems. (3-0) Cr. 3. F.S. Prereq: COM S 228 with C- or better, ENGL 250
Relational, object-oriented, and semi-structured data models and query languages, SQL, ODMG, and XML standards. Database design using entity-relationship model, data dependencies and object definition language. Application development in SQL-like languages and general purpose host languages with application program interfaces. Information integration using data warehouses, mediators and wrappers. Programming Projects. Nonmajor graduate credit.

COM S 396. Cooperative Education. Cr. R. Prereq: Permission of department chair Required of all cooperative students. Students must register for this course prior to commencing each work period.

COM S 401. Projects in Computing Applications. (2-2) Cr. 3. F. Prereq: ENGL 250, SP CM 212, COM S 309, and either COM S 362 or COM S 363
Applications of software development methods (requirements collection and analysis, software design, project management, documentation and testing), programming techniques, database designs and administration, network application programming to solve computing needs in business settings. A study of practical applications of emerging technologies in computing. Emphasis on semester-long team programming projects. Lab assignments. Oral and written reports. Nonmajor graduate credit.

COM S 409. Software Requirements Engineering. (Dual-listed with COM S 509). (Cross-listed with S E). (3-0) Cr. 3. F. Prereq: COM S 309, ENGL 250, SP CM 212
The requirements engineering process, including identification of stakeholders, requirements elicitation techniques such as interviews and prototyping, analysis fundamentals, requirements specification, and validation. Use of Models: State-oriented, Function-oriented, and Object-oriented. Documentation for Software Requirements. Informal, semi-formal, and formal representations. Structural, informational, and behavioral requirements. Non-functional requirements. Use of requirements repositories to manage and track requirements through the life cycle. Case studies, software projects, written reports, and oral presentations will be required. Nonmajor graduate credit.

COM S 412. Formal Aspects of Specification and Verification. (Cross-listed with CPR E, S E). (3-0) Cr. 3. S. Prereq: S E 319, COM S 309
Introduction to prepositional/predicate/temporal logic, program verification using theorem proving, model-based verification using model checking, and tools for verification. Nonmajor graduate credit.

COM S 414. Gerontechnology in Smart Home Environments. (Dual-listed with COM S 514). (3-0) Cr. 3. F. Prereq: Com S 227 or (Com S 207 or Geron 377 or ArtGr 271) or equivalent.
An interdisciplinary course designed for students who are interested in assistive technology, pervasive computing, mobile computing and principles of universal and inclusive design for end users, in particular, the elderly population. Students will work in semester-long projects as interdisciplinary teams to apply knowledge obtained from lectures and mutual presentations. For graduate credit students are required to submit a research report and give an oral presentation.

COM S 415. Gerontechnology in Smart Home Environments. (Cross-listed with GERON). (3-0) Cr. 3. F. Prereq: Com S 227 or (Com S 207 or Geron 377)
An interdisciplinary course designed for students who are interested in assistive technology, pervasive computing, mobile computing and principles of universal software design for end users, in particular the elderly population. Students will have the chance to learn both about the theories and principles about aging and assistive technology, as well as to engage in the practical semester-long project while working with students from other disciplines. Nonmajor graduate credit.

COM S 417. Software Testing. (Cross-listed with S E). (3-0) Cr. 3. S. Prereq: COM S 309, COM S 319, ENGL 250, SP CM 212
Comprehensive study of software testing, principles, methodologies, management strategies and techniques. Test models, test design techniques (black box and white box testing techniques), integration, regression, system testing methods, and software testing tools. Nonmajor graduate credit.

COM S 418. Introduction to Computational Geometry. (Dual-listed with COM S 518). (3-0) Cr. 3. Alt. S., offered 2013. Prereq: COM S 311 or permission of instructor
Introduction to data structures, algorithms, and analysis techniques for computational problems that involve geometry. Line segment intersection, polygon triangulation, 2D linear programming, range queries, point location, arrangements and duality, Voronoi diagrams and Delaunay triangulation, convex hulls, robot motion planning, visibility graphs. Other selected topics. Programming assignments. Nonmajor graduate credit.

COM S 421. Logic for Mathematics and Computer Science. (Cross-listed with MATH). (3-0) Cr. 3. S. Prereq: MATH 301 or MATH 317 or COM S 330
Propositional and predicate logic. Topics selected from Horn logic, equational logic, resolution and unification, foundations of logic programming, reasoning about programs, program specification and verification, model checking and binary decision diagrams, temporal logic and modal logic. Nonmajor graduate credit.

COM S 425. High Performance Computing for Scientific and Engineering Applications. (Cross-listed with CPR E). (3-1) Cr. 3. S. Prereq: COM S 311, COM S 330, ENGL 250, SP CM 212
Introduction to high performance computing platforms including parallel computers and workstation clusters. Discussion of parallel architectures, performance, programming models, and software development issues. Sample applications from science and engineering. Practical issues in high performance computing will be emphasized via a number of programming projects using a variety of programming models and case studies. Oral and written reports. Nonmajor graduate credit.

COM S 426. Introduction to Parallel Algorithms and Programming. (Dual-listed with COM S 526). (Cross-listed with CPR E). (3-2) Cr. 4. F. Prereq: CPR E 308 or COM S 321, COM S 311
Models of parallel computation, performance measures, basic parallel constructs and communication primitives, parallel programming using MPI, parallel algorithms for selected problems including sorting, matrix, tree and graph problems, fast Fourier transforms. Nonmajor graduate credit.

COM S 430. Advanced Programming Tools. (3-1) Cr. 3. F. Prereq: COM S 311, COM S 362 or COM S 363, ENGL 250, SP CM 212
Topics in advanced programming techniques and tools widely used by industry (e.g., event-driven programming and graphical user interfaces, standard libraries, client/server architectures and techniques for distributed applications). Emphasis on programming projects in a modern integrated development environment. Oral and written reports. Nonmajor graduate credit.

COM S 433. Computational Models of Nanoscale Self-Assembly. (Dual-listed with COM S 533). (3-0) Cr. 3. S. Prereq: C- or higher in COM S 331 or consent of the instructor
Modeling and analysis of natural and engineered systems that spontaneously assemble themselves from small components. Topics include biomolecular self-assembly, tile assembly models, computation via self-assembly, distributed folding, origami models, and self-repair. Emphasis on mathematical methods of describing, simulating, programming, and verifying the behaviors of self-assembling systems. Graduate credit requires a written or oral report on current research. Nonmajor graduate credit.

COM S 437. Computer Game and Media Programming. (3-0) Cr. 3. S. Prereq: COM S 362, COM S 321, COM S 336
Students will learn video game programming using current game engine interfaces with real hardware. Particular attention is paid to the console architecture, development environment, tool chains, 2D graphics, 3D graphics, controllers, memory management, and audio systems. Students will complete the course by writing a simple game that runs on console hardware. Nonmajor graduate credit.

Theory of compiling and implementation issues of programming languages. Programming projects leading to the construction of a compiler. Projects with different difficulty levels will be given for 440 and 540. Topics: lexical, syntax and semantic analyses, syntax-directed translation, runtime environment and library support. Written reports. Nonmajor graduate credit.
COM S 444. Introduction to Bioinformatics. (Cross-listed with BCS, BCBIO, BIOL, CPR E, GEN). (4-0) Cr. 4. F. Prereq: MATH 165 or STAT 401 or equivalent
Broad overview of bioinformatics with a significant problem-solving component, including hands-on practice using computational tools to solve a variety of biological problems. Topics include: database searching, sequence alignment, gene prediction, RNA and protein structure prediction, construction of phylogenetic trees, comparative and functional genomics, systems biology. Nonmajor graduate credit.

Laboratory course dealing with practical issues of design and implementation of distributed and network operating systems and distributed computing environments (DCE). The client server paradigm, inter-process communications, layered communication protocols, synchronization and concurrency control, and distributed file systems. Graduate credit requires additional in-depth study of advanced operating systems. Written reports.

Introduction to discrete-event simulation with a focus on computer science applications, including performance evaluation of networks and distributed systems. Overview of algorithms and data structures necessary to implement simulation software. Discrete and continuous stochastic models, random number generation, elementary statistics, simulation of queueing and inventory systems. Monte Carlo simulation, point and interval parameter estimation. Graduate credit requires additional in-depth study of concepts. Oral and written reports. Nonmajor graduate credit.

COM S 461. Principles and Internals of Database Systems. (Dual-listed with COM S 561). (3-1) Cr. 3. F. Prereq: COM S 311, ENGL 250, SP CM 212 and COM S 363

COM S 471. Computational Linear Algebra and Fixed Point Iteration. (Cross-listed with MATH). (3-0) Cr. 3. Alt. F., offered 2011 S. Prereq: Math 265 and either Math 266, or 267; knowledge of a programming language
Computational error, solutions of linear systems, least squares, similarity methods for eigenvalues, solution of nonlinear equations in one and several variables. Nonmajor graduate credit.

COM S 472. Principles of Artificial Intelligence. (Dual-listed with COM S 572). (3-1) Cr. 3. F. Prereq: COM S 311, COM S 330 or CPR E 310, STAT 330, ENGL 250, SP CM 212, COM S 342 or comparable programming experience
Specification, design, implementation, and selected applications of intelligent software agents and multi-agent systems. Computational models of intelligent behavior, including problem solving, knowledge representation, reasoning, planning, decision making, learning, perception, action, communication and interaction. Reactive, deliberative, rational, adaptive, learning and communicative agents and multiagent systems. Artificial intelligence programming. Graduate credit requires a research project and a written report. Oral and written reports. Nonmajor graduate credit.

COM S 474. Elements of Neural Computation. (3-1) Cr. 3. Alt. F., offered 2012. Prereq: COM S 311, COM S 330 or CPR E 310, STAT 330, MATH 165, ENGL 250, SP CM 212, COM S 342 or comparable programming experience
Introduction to theory and applications of neural computation and computational neuroscience. Computational models of neurons and networks of neurons. Neural architectures for associative memory, knowledge representation, inference, pattern classification, function approximation, stochastic search, decision making, and behavior. Neural architectures and algorithms for learning including perceptrons, support vector machines, kernel methods, bayesian learning, instance based learning, reinforcement learning, unsupervised learning, and related techniques. Applications in Artificial Intelligence and cognitive and neural modeling. Hands-on experience is emphasized through the use of simulation tools and laboratory projects. Oral and written reports. Nonmajor graduate credit.

COM S 477. Problem Solving Techniques for Applied Computer Science. (Dual-listed with COM S 577). (3-0) Cr. 3. Alt. F., offered 2012. Prereq: COM S 228; COM S 330 or CPR E 310, MATH 166, MATH 307 or MATH 317, or consent of the instructor
Selected topics in applied mathematics and modern heuristics that have found applications in areas such as geometric modeling, graphics, robotics, vision, human machine interface, speech recognition, computer animation, etc. Polynomial interpolation, roots of polynomials, resultants, solution of linear and nonlinear equations, approximation, data fitting, fast Fourier transform, linear programming, nonlinear optimization, Lagrange multipliers, genetic algorithms, integration of ODEs, curves, curvature, Frenet Formulas, cubic splines, and Bezier curves. Programming components. Written report for graduate credit. Nonmajor graduate credit.

COM S 481. Numerical Methods for Differential Equations and Interpolation. (Cross-listed with MATH). (3-0) Cr. 3. S. Prereq: MATH 265 and either MATH 266 or MATH 267; knowledge of a programming language

COM S 486. Fundamental Concepts in Computer Networking. (3-0) Cr. 3. S. Prereq: COM S 352
An introduction to fundamental concepts in the design and implementation of computer communication in both the wired and wireless networks, their protocols, and applications. Layered network architecture in the Internet, applications, transport, Socket APIs, network, and data link layers and their protocols, multimedia networking, and network security. Nonmajor graduate credit.

COM S 487. Principles of Distributed and Network Programming. (Dual-listed with COM S 587). (3-0) Cr. 3. F. Prereq: COM S 352 and either MATH 487 or E 489 or equivalent
Programming paradigms for building distributed and networking applications, including multithreaded client-server programming, socket programming, distributed object frameworks and programming suites, and web computing and security. Introduction to some on-going research issues in distributed and networking applications, including peer-to-peer computing, multimedia communications, and mobile computing and networking. Graduate credit requires a written report and an oral presentation. Nonmajor graduate credit.

COM S 490. Independent Study.
Cr. arr. Repeatable, maximum of 9 credits. F-S. Prereq: 6 credits in computer science, permission of instructor
Offered on a satisfactory-fail basis only. No more than 9 credits of Com S 490 may be counted toward graduation.

COM S 490H. Independent Study: Honors.
Cr. arr. Repeatable, maximum of 9 credits. F-S. Prereq: 6 credits in computer science, permission of instructor
Offered on a satisfactory-fail basis only. No more than 9 credits of Com S 490 may be counted toward graduation.

Courses primarily for graduate students, open to qualified undergraduates:

(Cross-listed with CAS). (1-0) Cr. 1. F. S. Prereq: Admission to CAS minor
Understanding core techniques in artificial life is based on basic readings in complex adaptive systems. Techniques of complex system analysis methods including: evolutionary computation, neural nets, agent based simulations (agent based computational economics). Large-scale simulations are to be emphasized, e.g. power grids, whole ecosystems.

(Cross-listed with CAS). (3-0) Cr. 3. S. Prereq: Admission to CAS minor or related field
Survey of complex systems and their analysis. Examples are drawn from engineering, computer science, biology, economics and physics.
COM S 509. Software Requirements Engineering.
(Dual-listed with COM S 409). (Cross-listed with S E). (3-0) Cr. 3. F. Prereq: COM S 309
The requirements engineering process including identification of stakeholders requirements elicitation techniques such as interviews and prototyping, analysis fundamentals, requirements specification, and validation. Use of Models: State-oriented, Function-oriented, and Object-oriented. Documentation for Software Requirements. Infallible, semi-formal, and formal representations. Structural, informational, and behavioral requirements. Non-functional requirements. Use of requirements repositories to manage and track requirements through the life cycle. Case studies, software projects, written reports, and oral presentations will be required.

COM S 510. Distributed Software Development.
(3-0) Cr. 3. F.
Team with students at a foreign university to develop a software application. Importance of distributed development. Design for distributed development, effective processes for distributed development, cultural issues in distributed development, organizing for distributed development, communication techniques and skills for distributed development.

(Cross-listed with CPR E). (3-0) Cr. 3. F. Prereq: COM S 311
A study of basic algorithm design and algorithm analysis techniques. Advanced data structures, amortized analysis and randomized algorithms. Applications to sorting, graphs, and geometry. NP-completeness and approximation algorithms.

(3-0) Cr. 3. S. Prereq: COM S 311, COM S 330
A study of formal techniques for model-based specification and verification of software systems. Topics include logics, formalisms, graph theory, numerical computations, algorithms and tools for automatic analysis of systems. Graduate credit requires in-depth study of concepts.

COM S 514. Gerontechnology in Smart Home Environments.
(Dual-listed with COM S 414). (3-0) Cr. 3. F. Prereq: Com S 227 or (Com S 207 or Geron 377 or ArtGr 271) or equivalent.
An interdisciplinary course designed for students who are interested in assistive technology, pervasive computing, mobile computing and principles of universal and inclusive design for end users, in particular, the elderly population. Students will work in semester-long projects as interdisciplinary teams to apply knowledge obtained from lectures and mutual presentations. For graduate credit students are required to submit a research report and give an oral presentation.

(3-0) Cr. 3. F. Prereq: COM S 309 or COM S 311, COM S 342
An introduction to the analysis, design, and testing of software for safety-critical and high-integrity systems. Analysis techniques, formal verification, fault identification and recovery, model checking, and certification issues. Emphasizes a case-based and systematic approach to software’s role in safe systems.

COM S 518. Introduction to Computational Geometry.
(Dual-listed with COM S 418). (3-0) Cr. 3. Alt. S., offered 2013. Prereq: COM S 311 or permission of instructor
Introduction to data structures, algorithms, and analysis techniques for computational problems that involve geometry. Line segment intersection, polygon triangulation, 2D linear programming, range queries, point location, arrangements and duality, Voronoi diagrams and Delaunay triangulation, convex hulls, robot motion planning, visibility graphs. Other selected topics. Programming assignments. A scholarly report must be submitted for graduate credit.

(Cross-listed with MATH, CPR E). (3-0) Cr. 3. Alt. S., offered 2015. Prereq: CPR E 308 or MATH 481; experience in scientific programming; knowledge of FORTRAN or C
Introduction to parallelization techniques and numerical methods for state-of-the-art high performance computers. A major component will be a final project in an area related to each student’s research interests.

(Dual-listed with COM S 426). (Cross-listed with CPR E). (3-2) Cr. 4. F. Prereq: CPR E 308 or COM S 321, COM S 311
Models of parallel computation, performance measures, basic parallel constructs and communication primitives, parallel programming using MPI, parallel algorithms for selected problems including sorting, matrix, tree and graph problems, fast Fourier transforms.

(3-0) Cr. 3. S. Prereq: COM S 331
A systematic study of the fundamental models and analytical methods of theoretical computer science. Computability, the Church-Turing thesis, decidable and undecidable problems, and the elements of recursive function theory. Time complexity, logic, Boolean circuits, and NP-completeness. Role of randomness in computation.

(Dual-listed with COM S 433). (3-0) Cr. 3. S. Prereq: C- or higher in COM S 331 or consent of the instructor
Modeling and analysis of natural and engineered systems that spontaneously assemble themselves from small components. Topics include biomolecular self-assembly, tile assembly models, computation via self-assembly, distributed folding, origami models, and self-repair. Emphasis on mathematical methods of describing, simulating, programming, and verifying the behaviors of self-assembling systems. Graduate credit requires a written or oral report on current research.

COM S 540. Principles and Practice of Compiling.
(Dual-listed with COM S 440). (3-1) Cr. 3. Alt. S., offered 2013. Prereq: COM S 331, COM S 342, ENGL 250, SP CM 212
Theory of compiling and implementation issues of programming languages. Programming projects leading to the construction of a compiler. Projects with different difficulty levels will be given for 440 and 540. Topics: lexical, syntax and semantic analyses, syntax-directed translation, runtime environment and library support. Written reports.

COM S 541. Programming Languages.
(3-1) Cr. 3. F. Prereq: COM S 342 or COM S 440
Survey of the goals and problems of language design. Formal and informal studies of a wide variety of programming language features including type systems. Creative use of functional and declarative programming paradigms.

COM S 544. Introduction to Bioinformatics.
(Cross-listed with BCB, CPR E, GDCB). (4-0) Cr. 4. F. Prereq: MATH 165 or STAT 401 or equivalent
Broad overview of bioinformatics with a significant problem-solving component, including hands-on practice using computational tools to solve a variety of biological problems. Topics include: database searching, sequence alignment, gene prediction, RNA and protein structure prediction, construction of phylogenetic trees, comparative, functional genomics, and systems biology.

(Cross-listed with CPR E). (3-0) Cr. 3. Alt. S., offered 2012. Prereq: COM S 311 and either COM S 228 or COM S 208
Design and analysis of algorithms for applications in computational biology, pairwise and multiple sequence alignments, approximation algorithms, string algorithms including in-depth coverage of suffix trees, semi-numerical string algorithms, algorithms for selected problems in fragment assembly, phylogenetic trees and protein folding. No background in biology is assumed. Also useful as an advanced algorithms course in string processing.

(3-0) Cr. 3. Alt. F., offered 2011. Prereq: COM S 311 and some knowledge of programming
Discussion and analysis of basic evolutionary principles and the necessary knowledge in computational biology to solve real world problems. Topics include character and distance based methods, phylogenetic tree distances, and consensus methods, and approaches to extract the necessary information from sequence-databases to build phylogenetic trees.

COM S 551. Computational Techniques for Genome Assembly and Analysis.
(3-0) Cr. 3. Alt. F., offered 2011. Prereq: COM S 311 and some knowledge of programming
Huang. Introduction to practical sequence assembly and comparison techniques. Topics include global alignment, local alignment, overlapping alignment, banded alignment, linear-space alignment, word hashing, DNA-protein alignment, DNA-cDNA alignment, comparison of two sets of sequences, construction of contigs, and generation of consensus sequences. Focus on development of sequence assembly and comparison programs.

(3-0) Cr. 3. Alt. S., offered 2012. Prereq: COM S 362
A comparative study of high-level language facilities for process synchronization and communication. Formal analysis of deadlock, concurrency control and recovery. Protection issues including capability-based systems, access and flow control, encryption, and authentication. Additional topics chosen from distributed operating systems, soft real-time operating systems, and advanced security issues.
(Dual-listed with COM S 454). (Cross-listed with CPR E). (3-1) Cr. 3. Alt. S., offered 2013. Prereq: COM S 311, COM S 352
Laboratory course dealing with practical issues of design and implementation of distributed and network operating systems and distributed computing environments (DCE). The client server paradigm, inter-process communications, layered communication protocols, synchronization and concurrency control, and distributed file systems. Graduate credit requires additional in-depth study of advanced operating systems. Written reports.

(Dual-listed with COM S 455). (3-0) Cr. 3. Alt. F., offered 2012. Prereq: COM S 311 and COM S 330, STAT 330
Introduction to discrete-event simulation with a focus on computer science applications, including performance evaluation of networks and distributed systems. Overview of algorithms and data structures necessary to implement simulation software. Discrete and continuous stochastic models, random number generation, elementary statistics, simulation of queuing and inventory systems, Monte Carlo simulation, point and interval parameter estimation. Graduate credit requires additional in-depth study of concepts. Oral and written reports.

(3-0) Cr. 3. Alt. S., offered 2013. Prereq: COM S 331, MATH 307, and STAT 330
Introduction to the use of stochastic models to study complex systems, including network communication and distributed systems. Data structures and algorithms for analyzing discrete-state models expressed in high-level formalisms. State space and reachability graph construction, model checking, Markov chain construction and numerical solution, computation of performance measures, product-form models, approximations, and advanced techniques.

(Cross-listed with CPR E, M E). (3-0) Cr. 3. F.S. Prereq: M E 421, programming experience in C

COM S 558. Introduction to the 3D Visualization of Scientific Data.
(Cross-listed with HCI, GEOL). (2-2) Cr. 3. Alt. S., offered 2013. Prereq: Graduate student standing in the mathematical or natural sciences
Introduction to visualizing scientific information with 3D computer graphics and their foundation in human perception. Overview of different visualization techniques and examples of 3D visualization projects from different disciplines (natural sciences, medicine, engineering). Class project in interactive 3D visualization using the OpenDX, VTK or a similar system.


(3-0) Cr. 3. F. Prereq: COM S 461 or COM S 561
Implementation topics and projects are chosen from the following: Storage architecture, buffer management and caching, access methods, design, parsing and compilation of query languages and update operations, application programming interfaces (APIs), user interfaces, query optimization and processing, and transaction management for relational, object-oriented, semistructured (XML), and special purpose database models; client-server architectures, metadata and middleware for database integration, web databases.

(Cross-listed with BCB, CPR E). (3-0) Cr. 3. S. Prereq: COM S 208; COM S 330; STAT 341; credit or enrollment in BIL 315, STAT 430

COM S 568. Bioinformatics II (Advanced Genome Informatics).
(Cross-listed with GDCB, STAT, BCB). (3-0) Cr. 3. S. Prereq: BCB 567, BBMB 301, BIL 315, STAT 430, credit or enrollment in GEN 411

COM S 569. Bioinformatics III (Structural Genome Informatics).
(Cross-listed with BCB, BBMB, CPR E). (3-0) Cr. 3. F. Prereq: BCB 567, GEN 411, STAT 430

COM S 570. Bioinformatics IV (Computational Functional Genomics and Systems Biology).
(Cross-listed with CPR E, BCB, GDCB, STAT). (3-0) Cr. 3. S. Prereq: BCB 567, BIOL 315, COM S 311 and either 208 or 228, GEN 411, STAT 430

(Dual-listed with COM S 472). (3-1) Cr. 3. F. Prereq: COM S 311, COM S 331, STAT 330, COM S 342 or comparable programming experience
Specification, design, implementation, and selected applications of intelligent software agents and multi-agent systems. Computational models of intelligent behavior, including problem solving, knowledge representation, reasoning, planning, decision making, learning, perception, action, communication and interaction. Reactive, deliberative, rational, adaptive, learning and communicative agents. Artificial intelligence programming. Graduate credit requires a research project and a written report. Oral and written reports.

(3-1) Cr. 3. S. Prereq: COM S 311, COM S 362, STAT 330

COM S 574. Intelligent Multagent Systems.
(3-0) Cr. 3. Alt. F., offered 2011. Prereq: STAT 330; COM S 311; COM S 572, COM S 573, COM S 472, or COM S 474
Specification, design, implementation, and applications of multi-agent systems. Intelligent agent architectures; infrastructures, languages and tools for design and implementation of distributed multi-agent systems; Multi-agent organizations, communication, interaction, cooperation, team formation, negotiation, competition, and learning. Selected topics in decision theory, game theory, contract theory, bargaining theory, auction theory, and organizational theory. Selected topics in knowledge representation and ontologies. Agent-based systems and the Semantic Web. Applications in distributed intelligent information networks for information retrieval, information integration, inference, and discovery from heterogeneous, autonomous, distributed, dynamic information sources.

COM S 575. Computational Perception.
(Cross-listed with HCI, CPR E). (3-0) Cr. 3. S. Prereq: Graduate standing or permission of instructor
This class covers statistical and algorithmic methods for sensing, recognizing, and interpreting the activities of people by a computer. This semester we will focus on machine perception techniques that facilitate and augment human-computer interaction. The main goal of the class is to introduce computational perception on both theoretical and practical levels. Participation in small groups to design, implement, and evaluate a prototype of a human-computer interaction system that uses one or more of the techniques covered in the lectures.
Courses for graduate students:

**COM S 510. Seminar.**
Offered on a satisfactory-fail basis only.

**COM S 511. Advanced Topics in Analysis of Algorithms.**
Advanced algorithm analysis and design techniques. Topics include graph algorithms, algebraic algorithms, number-theoretic algorithms, randomized and parallel algorithms. Intractable problems and NP-completeness. Advanced data structures.

**COM S 612. Distributed Algorithms.**
Cr. arr. Repeatable. Alt. S., offered 2013. Prereq: COM S 511 or COM S 531
Parallel solutions of numerical and non-numerical problems, implementation of parallel programs on parallel machines, performance and other computational issues in parallel programming.

**COM S 626. Parallel Algorithms for Scientific Applications.**
Cr. arr. Repeatable. Alt. S., offered 2013. Prereq: COM S 511
Advanced study in the quantitative theory of computation. Time and space complexity of algorithmic problems. The structure of P, NP, PH, PSPACE, and other complexity classes, especially with respect to resource-bounded reducibilities and complete problems. Complexity relative to auxiliary information, including oracle computation and relativized classes, randomized algorithms, advice machines, Boolean circuits. Kolmogorov complexity and randomness.

**COM S 633. Advanced Topics in Computational Randomness.**
Cr. arr. Repeatable. Alt. S., offered 2013. Prereq: COM S 511
Advanced study of the role of randomness in computation. Randomized algorithms, derandomization, and probabilistic complexity classes. Kolmogorov complexity, algorithmic information theory, and algorithmic randomness. Applications chosen from cryptography, interactive proof systems, computational learning, lower bound arguments, mathematical logic, and the organization of complex systems.

**COM S 634. Theory of Games, Knowledge and Uncertainty.**
Cr. arr. Repeatable. Alt. S., offered 2013. Prereq: COM S 511
Fundamentals of Game Theory: individual decision making, strategic and extensive games, mixed strategies, backward induction, Nash and other equilibrium concepts. Discussion of Auctions and Bargaining. Repeated, Bayesian and evolutionary games. Interactive Epistemology: reasoning about knowledge in multilateral environment, properties of knowledge, agreements, and common knowledge. Reasoning about and representing uncertainty, probabilities, and beliefs. Uncertainty in multilateral environments. Aspects and applications of game theory, knowledge, and uncertainty in other areas, especially Artificial Intelligence and Economics, will be discussed.

**COM S 641. Advanced Topics in Programming Language Semantics.**
Advanced study in the quantitative theory of computation. Time and space complexity of algorithmic problems. The structure of P, NP, PH, PSPACE, and other complexity classes, especially with respect to resource-bounded reducibilities and complete problems. Complexity relative to auxiliary information, including oracle computation and relativized classes, randomized algorithms, advice machines, Boolean circuits. Kolmogorov complexity and randomness.
(3-0) Cr. 3. Alt. F., offered 2012. Prereq: COM S 228, I E 557/M E 557/CPR E 557/COM S 557

COM S 661. Advanced Topics in Database Systems. 
(3-0) Cr. 3. Repeatable. Alt. F., offered 2012. Prereq: COM S 461 or COM S 561
Advanced topics chosen from the following: database design, data models, query systems, query optimization, incomplete information, logic and databases, multimedia databases; temporal, spatial and belief databases, semistructured data, concurrency control, parallel and distributed databases, information retrieval, data warehouses, wrappers, mediators, and data mining.

COM S 672. Advanced Topics in Computational Models of Learning. 
(3-0) Cr. 3. Repeatable. Alt. S., offered 2012. Prereq: COM S 572 or COM S 573 or COM S 472 or COM S 474

COM S 673. Advanced Topics in Computational Intelligence. 
(3-0) Cr. 3. Repeatable. Alt. S., offered 2013. Prereq: COM S 572 or COM S 573 or COM S 472 or COM S 474
Advanced applications of artificial intelligence in bioinformatics, distributed intelligent information networks and the Semantic Web. Selected topics in distributed learning, incremental learning, multi-task learning, multi-strategy learning; Graphical models, multi-relational learning, and causal inference; statistical natural language processing; modeling the internet and the web; automated scientific discovery; neural and cognitive modeling.

Repeatable with Instructor permission
Current topics in computer architecture design and implementation. Advanced pipelining, cache and memory design techniques. Interaction of algorithms with architecture models and implementations. Tradeoffs in architecture models and implementations.

COM S 686. Advanced Topics in High-Speed Networks. 
(3-0) Cr. 3. Alt. S., offered 2013. Prereq: COM S 586
Advanced topics in IP networks and optical networks. QoS routing and scheduling, multicast, multiprotocol label switching (MPLS), traffic engineering. Optical network architectures, routing and wavelength assignment algorithms, optical multicast, traffic grooming, optical burst switching, lightpath protection/restoration schemes, and IP over WDM.

COM S 699. Research. 
Cr. arr. Repeatable. Prereq: Approval of instructor
Offered on a satisfactory-fail basis only.

(Dual-listed with COM S 412). (Cross-listed with COM S, S E). (3-0) Cr. 3. Prereq: S E 319, COM S 309
Introduction to prepositional/predicate/temporal logic, program verification using theorem proving, model-based verification using model checking, and tools for verification. Nonmajor graduate credit.