COMPUTER SCIENCE

http://www.cs.iastate.edu

The undergraduate curriculum in Computer Science leading to the Bachelor of Science degree is accredited by the Computing Accreditation Commission of ABET, http://www.abet.org. This degree equips students with a sound knowledge of the foundations of Computer Science as well as problem solving and system design skills necessary to create robust, efficient, reliable, scalable, and flexible software systems. The B.S. degree in Computer Science prepares students for graduate study in Computer Science and for various business, industry, and government positions including computer scientists, information technologists, and software developers. 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 teaming 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.

Curriculum in Computer Science

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 46 credits in Computer Science and satisfaction of written and oral requirements. Students must earn at least a C- in Math 165, Math 166, and each Computer Science course taken to fulfill the Degree Program. The LAS College requires the major must contain at least 8 credits in courses taken at Iowa State University that are numbered 300 or above and in which the student's grade is C or higher.

The following courses are required:

COM S 101	Orientation	R
COM S 203	Careers in Computer Science	R
COM S 227	Introduction to Object-oriented Programming	4
COM S 228	Introduction to Data Structures	3
COM S 230	Discrete Computational Structures	3
COM S 309	Software Development Practices	3
COM S 311	Design and Analysis of Algorithms	3
COM S 321	Introduction to Computer Architecture and Machine-Level Programming	3
COM S 327	Advanced Programming Techniques	3
COM S 331	Theory of Computing	3
COM S 342	Principles of Programming Languages	3
COM S 352	Introduction to Operating Systems	3
At least 6 credits of 300-level courses, from the following:		
COM S 319	Software Construction and User Interfaces	3
COM S 336	Introduction to Computer Graphics	3
COM S 362	Object-Oriented Analysis and Design	3
COM S 363	Introduction to Database Management Systems	3
At least 3 credits of Computer Science Senior Project		
COM S 402	Computer Science Senior Project	3-4
At least 6 credits of 400-level courses, all with a grade of C- or better, from the following:		
COM S 409	Software Requirements Engineering	3
COM S 410	Distributed Development of Software	3
COM S 412	Formal Methods in Software Engineering	3
COM S 415	Software System Safety	3
COM S 417	Software Testing	3
COM S 418	Introduction to Computational Geometry	3
COM S 421	Logic for Mathematics and Computer Science	3
COM S 424	Introduction to High Performance Computing	3
COM S 425	High Performance Computing for Scientific and Engineering Applications	3
COM S 426	Introduction to Parallel Algorithms and Programming	4
COM S 430	Advanced Programming Tools	3
COM S 433	Computational Models of Nanoscale Self-Assembly	3
COM S 435	Algorithms for Large Data Sets: Theory and Practice	3

COM S 437	Computer Game and Media Programming	3
COM S 440	Principles and Practice of Compiling	3
COM S 441	Programming Languages	3
COM S 444	Bioinformatic Analysis	4
COM S 454	Distributed Systems	3
COM S 455	Simulation: Algorithms and Implementation	3
COM S 461	Principles and Internals of Database Systems	3
COM S 472	Principles of Artificial Intelligence	3
COM S 474	Introduction to Machine Learning	3
COM S 477	Problem Solving Techniques for Applied Computer Science	3
COM S 481	Numerical Methods for Differential Equations	3
COM S 486	Fundamental Concepts in Computer Networking	3
COM S 487	Network Programming, Applications, and Research Issues	3
CPR E 431	Basics of Information System Security	3
CPR E 458	Real Time Systems	3
CPR E 489	Computer Networking and Data Communications	4

Com S 414 **may not** be applied towards fulfilling the 400-level electives.

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

PHIL 343	Philosophy of Technology	3
SP CM 212	Fundamentals of Public Speaking	3
At least 14 credits of Math and Statistics		14
MATH 165	Calculus I	4
MATH 166	Calculus II	4
One Statistics course from:		
STAT 305	Engineering Statistics	3
STAT 330	Probability and Statistics for Computer Science	3
STAT 341	Introduction to the Theory of Probability and Statistics I	4
At least one Math course from:		
MATH 207	Matrices and Linear Algebra	3
MATH 265	Calculus III	4
MATH 266	Elementary Differential Equations	3
MATH 267	Elementary Differential Equations and Laplace Transforms	4
MATH 304	Combinatorics	3
MATH 314	Graph Theory	3
MATH 317	Theory of Linear Algebra	4
One of the following 2-course Natural Science sequences (with labs):		
BIOL 211 & 211L	Principles of Biology I and Principles of Biology Laboratory I	4
BIOL 212 & 212L	Principles of Biology II and Principles of Biology Laboratory II	4
BIOL 255 & 255L	Fundamentals of Human Anatomy and Fundamentals of Human Anatomy Laboratory	4
BIOL 256 & 256L	Fundamentals of Human Physiology and Fundamentals of Human Physiology Laboratory	4
CHEM 177 & 177L	General Chemistry I and Laboratory in General Chemistry I	5
CHEM 178 & 178L	General Chemistry II and Laboratory in College Chemistry II	4
GEOL 100 & 100L	The Earth and The Earth: Laboratory	4
GEOL 102 & 102L	History of the Earth and History of the Earth: Laboratory	4
PHYS 221	Introduction to Classical Physics I	5
or PHYS 241	Principles and Symmetries in Classical Physics I	
PHYS 222	Introduction to Classical Physics II	5
or PHYS 242	Principles and Symmetries in Classical Physics II	

The following courses meet the communication proficiency requirement:

LIB 160	Information Literacy	1
ENGL 150	Critical Thinking and Communication	3
ENGL 250	Written, Oral, Visual, and Electronic Composition	3
One of the following		
ENGL 302	Business Communication	3
ENGL 305	Creative Writing: Nonfiction	3
ENGL 309	Proposal and Report Writing	3
ENGL 314	Technical Communication	3

According to the university-wide Communication Proficiency Grade Requirement, students must demonstrate their communication proficiency by earning a grade of C or better in ENGL 250. The Department requires a C or higher in the upper-level ENGL course (302, 305, 309, 314).

To obtain a bachelor's degree from the College of Liberal Arts and Sciences, curriculum in liberal arts and sciences, a student must earn at least 45 credits at the 300 level or above taken at a four-year college. All such credits, including courses taken on a pass/not pass basis, may be used to meet this requirement.

Students must take at least 15 credits of Computer Science courses at the 300 level or higher at Iowa State University while resident here.

Computer Science transfer courses need to be a minimum grade of C or higher to be considered for course substitution.

Students must earn a C- or better in each Computer Science course 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 16 credits in computer science courses. Com S 414 cannot be used to fulfill minor requirements.

A minimum grade of C- is required in Com S 227 and Com S 228. A minimum grade of C is required in both Com S 311 and the three credits of 300-level Computer Science courses and above. At least 6 credits of the minor must be in courses numbered 300 and above and taken at ISU with a grade of C or higher. The minor must include at least 9 credits that are not used to meet any other department, college, or university requirement.

COM S 227	Introduction to Object-oriented Programming	4
COM S 228	Introduction to Data Structures	3
COM S 230	Discrete Computational Structures	3
COM S 311	Design and Analysis of Algorithms	3
3 credits in ComS courses at the 300 level or above		3

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.

Computer Science, B.S.

Freshman

Fall	Credits	Spring	Credits
COM S 101	0	COM S 228	3
COM S 227	4	COM S 230	3
MATH 165	4	MATH 166	4
ENGL 150	3	Social Science	3
LIB 160	1	Foreign Language 102/ Elective	3-4
Foreign Language 101/ Elective	3-4		
	15-16		16-17

Sophomore

Fall	Credits	Spring	Credits
COM S 203	0	COM S 321	3
COM S 311	3	COM S 331	3
COM S 327	3	STAT 300-level elective	3
ENGL 250	3	Social science	3
Natural Science 1	5	Natural Science 2	5
	14		17

Junior

Fall	Credits	Spring	Credits
COM S 309	3	COM S 352	3
COM S 342	3	COM S 300-level elective	3
COM S 300-level elective	3	ENGL 300-level Elective	3
MATH elective	3-4	Arts and Humanities	3
SP CM 212	3	Elective	3
	15-16		15

Senior

Fall	Credits	Spring	Credits
COM S Senior Project 1	2	COM S Senior Project 2	2
COM S 400-level Elective	3	COM S 400-level Elective	3
PHIL 343	3	Arts and Humanities	3
Arts and Humanities	3	Social Science	3
Elective	3	Elective	3
	14		14

Graduate Study

The department offers graduate programs leading to degrees of Master of Science (MS) and Doctor of Philosophy (PhD) 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 for the 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 or a creative component 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 PhD requirements are governed by the student's program of study committee within established guidelines of the department and the graduate college. They include coursework (demonstrating breadth and depth of knowledge), 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.

Courses primarily for undergraduates:

COM S 101: Orientation
Cr. R.
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.
Introduction to computer literacy and applications. Applications: Windows, Internet browser/HTML, word processing, spreadsheets, database management and presentation software. Literacy: history of computing, structure of computers, telecommunications, computer ethics, computer crime, and history of programming languages. No prior computer experience necessary. Offered online only. Attendance at an orientation session the first week of class is required. Only one of COM S 103 and COM S 113 may count toward graduation.

COM S 104: Introduction to Programming
(1.5-1) Cr. 2.
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 105: Short Course in Computer Programming
Cr. 2.
Prereq: Com S 104
8-week course in programming, including instruction in syntax and semantics, of the following current programming languages.

COM S 105A: Short Course in Computer Programming: Perl
(1-2) Cr. 2.
Prereq: Com S 104
8-week course in programming using Perl.

COM S 105B: Short Course in Computer Programming: MATLAB
(2-0) Cr. 2.
Prereq: Com S 104
8-week course in programming using MATLAB.

COM S 106: Introduction to Web Programming
(3-0) Cr. 3.
Introduction to Web programming basics. Fundamentals of developing Web pages using a comprehensive Web development life cycle. In-depth experience with current Web design techniques such as HTML5 and cascading style sheets. Programming with JavaScript, jQuery, PHP, SQL, and MySQL. Strategies for accessibility, usability and search engine optimization.

COM S 107: Applied Computer Programming
(3-0) Cr. 3.
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 108: Applied Computer Programming II
(3-0) Cr. 3.
Prereq: Com S 107 or equivalent
Advanced programming applications in Visual Basic for non-majors. Emphasis on programming projects including sorting, file processing, database processing, web programming, and graphics and animation. Students will learn problem solving techniques and advanced programming skills to build real-world applications.

COM S 113: Introduction to Spreadsheets and Databases
(2-2) Cr. 3.
Using Microsoft Excel spreadsheets and Microsoft Access databases to input, store, process, manipulate, query, and analyze data for business and industrial applications. Credit in Com S 113 may not be applied toward graduation in the COM S, S E, and CPR E majors. Only one of COM S 103 and COM S 113 may count toward graduation.

COM S 127: Introduction to Programming for Problem Solving
(3-2) Cr. 4.
Prereq: MATH 140
Introduction to computer programming with an emphasis on problem solving. Topics include: program structures, expressions, variables, decision and logic, iteration, collections, input and output. Program construction and testing. Programming assignments including games and applications. No prior programming experience necessary. This course is intended for Computer Science majors.

COM S 203: Careers in Computer Science
Cr. R.
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.

COM S 207: Fundamentals of Computer Programming
(Cross-listed with MIS). (3-1) Cr. 3.
Prereq: MATH 150 or placement into MATH 140 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.

COM S 208: Intermediate Computer Programming
(3-1) Cr. 3.
Prereq: MIS/COM S 207, credit or enrollment in 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. 4.
Prereq: Placement into MATH 143, 165, or higher; recommended: a previous high school or college course in programming or equivalent experience.
Introduction to object-oriented design and programming techniques. Symbolic and numerical computation, recursion and iteration, modularity procedural and data abstraction, and specifications and subtyping. Object-oriented techniques including encapsulation, inheritance and polymorphism. Imperative programming. Emphasis on principles of programming and object-oriented design through extensive practice in design, writing, running, debugging, and reasoning. Course intended for Com S 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. 3.
Prereq: Minimum of C- in COM S 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 algoritms. Emphasis on object-oriented design, writing and documenting medium-sized programs. This course is designed for majors.

COM S 230: Discrete Computational Structures
(3-1) Cr. 3.
Prereq: Minimum of C- in COM S 227 and MATH 165; ENGL 150
Concepts in discrete mathematics as applied to computer science. Logic, proof techniques, set theory, relations, graphs, combinatorics, discrete probability and number theory.

COM S 252: Linux Operating System Essentials
(3-0) Cr. 3.
Prereq: COM S 107 or COM S 127 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. Repeatable, maximum of 6 credits.
Prereq: Permission of instructor
No more than 6 credits of Com S 290 or Com S 290H may be counted toward graduation.

COM S 290H: Independent Study: Honors
Cr. arr. Repeatable, maximum of 6 credits.
Prereq: Permission of instructor
No more than 6 credits of Com S 290 or Com S 290H may be counted toward graduation.

COM S 309: Software Development Practices
(3-1) Cr. 3.
Prereq: Minimum of C- in COM S 228 and MATH 165
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.

COM S 311: Design and Analysis of Algorithms
(3-1) Cr. 3.
Prereq: Minimum of C- in COM S 228; MATH 166, ENGL 150, and COM S 230 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.

COM S 319: Software Construction and User Interfaces
(Cross-listed with S E). (3-0) Cr. 3.
Prereq: COM S 228
Basic theory of grammars, parsing. Language paradigms. State transition and table-based software design. Review of principles of object orientation, object oriented analysis using UML. Frameworks and APIs. User interface architecture, evaluation of user interface. Design of windows, menus, and commands. Introduction to formal specification and model-based software design. Introduction to domain-specific software engineering.

COM S 321: Introduction to Computer Architecture and Machine-Level Programming
(3-1) Cr. 3.
Prereq: Minimum of C- in COM S 228 and MATH 165; COM S 230 or CPR E 281; 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 programming.

COM S 327: Advanced Programming Techniques
(3-0) Cr. 3.
Prereq: Minimum of C- in COM S 228 and MATH 165
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 331: Theory of Computing
(Cross-listed with LING). (3-1) Cr. 3.
Prereq: Minimum of C- in COM S 228, MATH 166, and in COM S 230 or CPR E 310; ENGL 250
Models of computation: finite state automata, pushdown automata and Turing machines. Study of grammars and their relation to automata. Limits of digital computation, unsolvability and Church-Turing thesis. Chomsky hierarchy and relations between classes of languages.

COM S 336: Introduction to Computer Graphics
(3-0) Cr. 3.
Prereq: COM S 327, CoReq MATH 207 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. 3.
Prereq: Minimum of C- in COM S 228 and MATH 165; COM S 230 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.

COM S 350: Number Theory
(Cross-listed with MATH). (3-0) Cr. 3. S.
Prereq: MATH 201 or COM S 230
Divisibility, integer representations, primes and divisors, linear diophantine equations, congruences, and multiplicative functions. Applications to cryptography.

COM S 352: Introduction to Operating Systems
(3-1) Cr. 3.
Prereq: COM S 321, and COM S 327; 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.

COM S 362: Object-Oriented Analysis and Design
(3-0) Cr. 3.
Prereq: Minimum of C- in COM S 228 and MATH 165; ENGL 250
Object-oriented requirements analysis and systems design. Design notations such as the Unifed Modeling Language. Design Patterns. Group design and programming with large programming projects.

COM S 363: Introduction to Database Management Systems
(3-0) Cr. 3.
Prereq: Minimum of C- in COM S 228 and MATH 165; ENGL 250
Relational, object-oriented, and semistructured 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.

COM S 398: Cooperative Education
Cr. R. Repeatable.
Prereq: Permission of department chair
Required of all cooperative students. Students must register for this course prior to commencing each work period.

COM S 402: Computer Science Senior Project
Cr. 2-3. Repeatable, maximum of 6 credits.
Students work as individuals and teams to complete the planning, design, and implementation of a significant project in the topic area. Oral and written reports. No more than 6 credits of 402A, 402B, and 402C may be used toward graduation.

COM S 402A: Computer Science Senior Project: Multimedia and Computer Gaming I
Cr. 2-3. Repeatable, maximum of 6 credits.
Prereq: COM S 437
Students conceive, plan, architect and design a computer game. Student registered in this course will work with students in ARTIS 409. Oral and written reports. No more than 6 credits of 402A, 402B, and 402C may be used toward graduation.

COM S 402B: Computer Science Senior Project: Multimedia and Computer Gaming II
Cr. 2-3. Repeatable, maximum of 6 credits.
Prereq: COM S 402A
Students implement, test, and present a completed production computer game. Students in this class will work with students in ARTIS 409. Oral and written reports. No more than 6 credits of 402A, 402B, and 402C may be used toward graduation.

COM S 402C: Computer Science Senior Project: Project in Computer Science
(0-6) Cr. 2-3. Repeatable, maximum of 6 credits.
Prereq: Permission of instructor
Students work as individuals and teams to complete the planning, design, and implementation of a significant project in the topic area. Oral and written reports. No more than 6 credits of 402A, 402B, and 402C may be used toward graduation.

COM S 409: Software Requirements Engineering
(Dual-listed with COM S 509). (3-0) Cr. 3.
Prereq: COM S 309; for graduate credit: graduate standing or permission of instructor
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 and software projects will be required.

COM S 410: Distributed Development of Software
(Dual-listed with COM S 510). (3-0) Cr. 3.
Prereq: COMS 228, COMS 309, COMS 327; for graduate credit: graduate standing or permission of instructor
Team with students at foreign universities to develop a software application. Importance of distributed development. Design for distributed development, effective processes for distributed development, and cultural issues in distributed development, organizing for distributed development, communication techniques and skills for distributed development,including oral presentations. Graduate credit requires in--depth study of concepts.

COM S 412: Formal Methods in Software Engineering
(Dual-listed with COM S 512). (Cross-listed with CPR E, S E). (3-0) Cr. 3.
Prereq: COM S 230 or CPR E 310; COM S 311, STAT 330; for graduate credit: graduate standing or permission of instructor
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 414: Gerontechnology in Smart Home Environments
(Dual-listed with COM S 514). (3-0) Cr. 3.
Prereq: COM S 227 or (COM S 207 or GERON 377 or ARTGR 271) or equivalent; for graduate credit: graduate standing or permission of instructor
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: Software System Safety
(Dual-listed with COM S 515). (3-0) Cr. 3.
Prereq: For graduate credit: graduate standing or permission of instructor
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 417: Software Testing
(Cross-listed with S E). (3-0) Cr. 3.
Prereq: COM S 309; COM S 230 or CPR E 310; 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), test adequacy criteria,integration, regression, system testing methods, and software testing tools.

COM S 418: Introduction to Computational Geometry
(Dual-listed with COM S 518). (3-0) Cr. 3.
Prereq: COM S 311 or permission of instructor; for graduate credit: graduate standing 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.

COM S 421: Logic for Mathematics and Computer Science
(Cross-listed with MATH). (3-0) Cr. 3.
Prereq: MATH 301 or MATH 207 or MATH 317 or COM S 230
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.

COM S 424: Introduction to High Performance Computing
(Cross-listed with CPR E, MATH). (2-2) Cr. 3. F.
Prereq: MATH 265; MATH 207 or MATH 317
UNIX, serial programming for high performance, OpenMP for high performance, shared memory parallelization. Semester project required.

COM S 425: High Performance Computing for Scientific and Engineering Applications
(Cross-listed with CPR E). (3-1) Cr. 3.
Prereq: COM S 311, COM S 230, 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.

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, CPR E 315 or 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.

COM S 430: Advanced Programming Tools
(3-1) Cr. 3.
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.

COM S 433: Computational Models of Nanoscale Self-Assembly
(Dual-listed with COM S 533). (3-0) Cr. 3.
Prereq: Minimum of C- in COM S 331 or consent of the instructor; for graduate credit: graduate standing or permission of 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 435: Algorithms for Large Data Sets: Theory and Practice
(Dual-listed with COM S 535). (3-0) Cr. 3.
Prereq: COM S 228, COM S 230 or CPR E 310, COM S 311 or equivalent
Challenges involved in solving computational problems on massive data sets. Discussion of computational problems that arise in the context of web search, social network analysis, recommendation systems, and online advertising etc. Theoretical aspects include modeling the computational problems using graphs, study of similarity measures and hash functions, and design of efficient algorithms for graphs. Practical aspects include implementation and performance evaluation of the algorithms on real world data sets. Graduate credit requires a written report on current research.

COM S 437: Computer Game and Media Programming
(3-0) Cr. 3.
Prereq: COM S 336 or permission of instructor
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.

COM S 440: Principles and Practice of Compiling
(Dual-listed with COM S 540). (3-1) Cr. 3.
Prereq: COM S 331, COM S 342, ENGL 250, SP CM 212; for graduate credit: graduate standing or permission of instructor
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 441: Programming Languages
(Dual-listed with COM S 541). (3-1) Cr. 3.
Prereq: COM S 342 or COM S 440; for graduate credit: graduate standing or permission of instructor
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 444: Bioinformatic Analysis
(Cross-listed with BCB, 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: bioinformatic data processing, Perl programming, genome assembly, database search, sequence alignment, gene prediction, next-generation sequencing, comparative and functional genomics, and systems biology.

COM S 454: Distributed Systems
(Dual-listed with COM S 554). (Cross-listed with CPR E). (3-1) Cr. 3.
Prereq: COM S 311, COM S 352; for graduate credit: graduate standing or permission of instructor
Theoretical and practical issues of design and implementation of distributed systems. The client server paradigm, inter-process communications, synchronization and concurrency control, naming, consistency and replication, fault tolerance, and distributed file systems. Graduate credit requires additional in-depth study of concepts. Programming projects and written reports.

COM S 455: Simulation: Algorithms and Implementation
(Dual-listed with COM S 555). (3-0) Cr. 3.
Prereq: COM S 311 and COM S 230, STAT 330, ENGL 150, SP CM 212; for graduate credit: graduate standing or permission of instructor
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.

COM S 461: Principles and Internals of Database Systems
(Dual-listed with COM S 561). (3-1) Cr. 3.
Prereq: COM S 311, ENGL 250, SP CM 212; for graduate credit: graduate standing or permission of instructor
Models for structured and semistructured data. Algebraic, first order, and user-oriented query languages. Database schema design. Physical storage, access methods, and query processing. Transaction management, concurrency control, and crash recovery. Database security. Information integration using data warehouses, mediators, wrappers, and data mining. Parallel and distributed databases, and special purpose databases. Students enrolling in Com S 561 will require additional study of advanced concepts in database systems.

COM S 472: Principles of Artificial Intelligence
(Dual-listed with COM S 572). (3-1) Cr. 3.
Prereq: COM S 311, COM S 230 or CPR E 310, STAT 330, ENGL 250, SP CM 212, COM S 342 or comparable programming experience; for graduate credit: graduate standing or permission of instructor
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. A research project and a written report is required for students enrolled in Com S 572.

COM S 474: Introduction to Machine Learning
(Dual-listed with COM S 574). (3-1) Cr. 3.
Prereq: COM S 311, COM S 230 or CPR E 310, STAT 330, MATH 165, ENGL 250, SP CM 212, COM S 342 or comparable programming experience
Basic principles, techniques, and applications of Machine Learning. Design, analysis, implementation, and applications of learning algorithms. Topics include: statistical learning, pattern classification, function approximation, Bayesian learning, linear models, artificial neural networks, support vector machines, decision trees, instance based learning, probabilistic graphical models, unsupervised learning, selected applications in automated knowledge acquisition, pattern recognition, and data mining.

COM S 477: Problem Solving Techniques for Applied Computer Science
(Dual-listed with COM S 577). (3-0) Cr. 3.
Prereq: COM S 228; COM S 230 or CPR E 310, MATH 166, MATH 207 or MATH 317, or consent of the instructor; for graduate credit: graduate standing or permission of 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. Homogeneous coordinates and transformations, perspective projection, quaternions and rotations, polynomial interpolation, roots of polynomials, resultants, solution of linear and nonlinear equations, approximation, data fitting, Fourier series and fast Fourier transform, linear programming, nonlinear optimization, Lagrange multipliers, parametric and algebraic curves, curvature, Frenet formulas, Bezier curves. Programming components. A scholarly report is required for graduate credit.

COM S 481: Numerical Methods for Differential Equations
(Cross-listed with MATH). (3-0) Cr. 3. S.
Prereq: MATH 265 and either MATH 266 or MATH 267
First order Euler method, high order Runge-Kutta methods, and multistep methods for solving ordinary differential equations. Finite difference and finite element methods for solving partial differential equations. Local truncation error, stability, and convergence for finite difference method. Numerical solution space, polynomial approximation, and error estimate for finite element method. Computer programming required.

COM S 486: Fundamental Concepts in Computer Networking
(3-0) Cr. 3.
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.

COM S 487: Network Programming, Applications, and Research Issues
(Dual-listed with COM S 587). (3-0) Cr. 3.
Prereq: Com S 352 or CPR E 489 or equivalent; for graduate credit: graduate standing or permission of instructor
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. A written report and an oral presentation is required for students enrolling in Com S 587.

COM S 490: Independent Study
Cr. arr. Repeatable, maximum of 9 credits.
Prereq: 6 credits in computer science, permission of instructor
No more than 9 credits of Com S 490 or Com S 490H may be counted toward graduation.

COM S 490H: Independent Study: Honors
Cr. arr. Repeatable, maximum of 9 credits.
Prereq: 6 credits in computer science, permission of instructor
No more than 9 credits of Com S 490 or Com S 490H may be counted toward graduation.

Courses primarily for graduate students, open to qualified undergraduates:

COM S 509: Software Requirements Engineering
(Dual-listed with COM S 409). (3-0) Cr. 3.
Prereq: COM S 309; for graduate credit: graduate standing or permission of instructor
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 and software projects will be required.

COM S 510: Distributed Development of Software
(Dual-listed with COM S 410). (3-0) Cr. 3.
Prereq: COMS 228, COMS 309, COMS 327; for graduate credit: graduate standing or permission of instructor
Team with students at foreign universities to develop a software application. Importance of distributed development. Design for distributed development, effective processes for distributed development, and cultural issues in distributed development, organizing for distributed development, communication techniques and skills for distributed development,including oral presentations. Graduate credit requires in--depth study of concepts.

COM S 511: Advanced Design and Analysis of Algorithms
(Cross-listed with CPR E). (3-0) Cr. 3.
Prereq: COM S 311
A study of algorithm design and analysis techniques. Advanced data structures, amortized analysis and randomized algorithms. Applications to sorting, graphs, and geometry. NP-completeness and approximation algorithms.

COM S 512: Formal Methods in Software Engineering
(Dual-listed with COM S 412). (3-0) Cr. 3.
Prereq: COM S 230 or CPR E 310; COM S 311, STAT 330; for graduate credit: graduate standing or permission of instructor
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.
Prereq: COM S 227 or (COM S 207 or GERON 377 or ARTGR 271) or equivalent; for graduate credit: graduate standing or permission of instructor
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 515: Software System Safety
(Dual-listed with COM S 415). (3-0) Cr. 3.
Prereq: For graduate credit: graduate standing or permission of instructor
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.
Prereq: COM S 311 or permission of instructor; for graduate credit: graduate standing 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.

COM S 525: Numerical Analysis of High Performance Computing
(Cross-listed with CPR E, MATH). (3-0) Cr. 3. S.
Prereq: CPR E 308 or MATH 481; experience in scientific programming; knowledge of FORTRAN or C
Introduction to parallelization techniques and numerical methods for distributed memory high performance computers. A semester project in an area related to each student's research interests is required.

COM S 526: Introduction to Parallel Algorithms and Programming
(Dual-listed with COM S 426). (Cross-listed with CPR E). (3-2) Cr. 4. F.
Prereq: CPR E 308 or COM S 321, CPR E 315 or 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.

COM S 531: Theory of Computation
(3-0) Cr. 3.
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.

COM S 533: Computational Models of Nanoscale Self-Assembly
(Dual-listed with COM S 433). (3-0) Cr. 3.
Prereq: Minimum of C- in COM S 331 or consent of the instructor; for graduate credit: graduate standing or permission of 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 535: Algorithms for Large Data Sets: Theory and Practice
(Dual-listed with COM S 435). (3-0) Cr. 3.
Prereq: COM S 228, COM S 230 or CPR E 310, COM S 311 or equivalent
Challenges involved in solving computational problems on massive data sets. Discussion of computational problems that arise in the context of web search, social network analysis, recommendation systems, and online advertising etc. Theoretical aspects include modeling the computational problems using graphs, study of similarity measures and hash functions, and design of efficient algorithms for graphs. Practical aspects include implementation and performance evaluation of the algorithms on real world data sets. Graduate credit requires a written report on current research.

COM S 540: Principles and Practice of Compiling
(Dual-listed with COM S 440). (3-1) Cr. 3.
Prereq: COM S 331, COM S 342, ENGL 250, SP CM 212; for graduate credit: graduate standing or permission of instructor
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
(Dual-listed with COM S 441). (3-1) Cr. 3.
Prereq: COM S 342 or COM S 440; for graduate credit: graduate standing or permission of instructor
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: Fundamentals of Bioinformatics
(Cross-listed with BCB, CPR E, GDCB). (4-0) Cr. 4. F.
Prereq: MATH 165 or STAT 401 or equivalent
A practical, hands-on overview of how to apply bioinformatics to biological research. Recommended for biologists desiring to gain computational molecular biology skills. Topics include: sequence analysis, genomics, proteomics, phylogenetic analyses, ontology enrichment, systems biology, data visualization and emergent technologies.

COM S 549: Advanced Algorithms in Computational Biology
(Cross-listed with CPR E). (3-0) Cr. 3.
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.

COM S 550: Evolutionary Problems for Computational Biologists
(3-0) Cr. 3.
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.
Prereq: COM S 311 and some knowledge of programming
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.

COM S 552: Principles of Operating Systems
(3-0) Cr. 3.
Prereq: For graduate credit: graduate standing or permission of instructor
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.

COM S 554: Distributed Systems
(Dual-listed with COM S 454). (Cross-listed with CPR E). (3-1) Cr. 3.
Prereq: COM S 311, COM S 352; for graduate credit: graduate standing or permission of instructor
Theoretical and practical issues of design and implementation of distributed systems. The client server paradigm, inter-process communications, synchronization and concurrency control, naming, consistency and replication, fault tolerance, and distributed file systems. Graduate credit requires additional in-depth study of concepts. Programming projects and written reports.

COM S 555: Simulation: Algorithms and Implementation
(Dual-listed with COM S 455). (3-0) Cr. 3.
Prereq: COM S 311 and COM S 230, STAT 330, ENGL 150, SP CM 212; for graduate credit: graduate standing or permission of instructor
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.

COM S 556: Analysis Algorithms for Stochastic Models
(3-0) Cr. 3.
Prereq: Graduate standing or permission of instructor
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.

COM S 557: Computer Graphics and Geometric Modeling
(Cross-listed with CPR E, M E). (3-0) Cr. 3. F.
Prereq: M E 421, programming experience in C
Fundamentals of computer graphics technology. Data structures. Parametric curve and surface modeling. Solid model representations. Applications in engineering design, analysis, and manufacturing.

COM S 558: Introduction to the 3D Visualization of Scientific Data
(Cross-listed with GEOL, HCI). (2-2) Cr. 3. Alt. F., offered even-numbered years.
Prereq: Graduate-student standing in the mathematical or natural sciences or engineering; basic programming knowledge
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, and engineering). Class project in interactive 3D visualization using the ParaView, Mayavi, TVTK, VTK or a similar system.

COM S 561: Principles and Internals of Database Systems
(Dual-listed with COM S 461). (3-1) Cr. 3.
Prereq: COM S 311, ENGL 250, SP CM 212; for graduate credit: graduate standing or permission of instructor
Models for structured and semistructured data. Algebraic, first order, and user-oriented query languages. Database schema design. Physical storage, access methods, and query processing. Transaction management, concurrency control, and crash recovery. Database security. Information integration using data warehouses, mediators, wrappers, and data mining. Parallel and distributed databases, and special purpose databases. Students enrolling in Com S 561 will require additional study of advanced concepts in database systems.

COM S 567: Bioinformatics I (Bioinformatics Algorithms)
(Cross-listed with BCB, CPR E). (3-0) Cr. 3.
Prereq: COM S 228; COM S 330; credit or enrollment in BIOL 315, STAT 430
Biology as an information science. A review of the algorithmic principles that are driving the advances in bioinformatics and computational biology.

COM S 568: Bioinformatics II (Statistical Bioinformatics)
(Cross-listed with BCB, GDCB, STAT). (3-0) Cr. 3. S.
Prereq: BCB 567 or (BIOL 315 and STAT 430), credit or enrollment in GEN 409
Statistical models for sequence data, including applications in genome annotation, motif discovery, variant discovery, molecular phylogeny, gene expression analysis, and metagenomics. Statistical topics include model building, inference, hypothesis testing, and simple experimental design, including for big data/complex models.

COM S 569: Bioinformatics III (Structural Bioinformatics)
(Cross-listed with BBMB, BCB, CPR E, GDCB). (3-0) Cr. 3. F.
Prereq: BCB 567, BBMB 316, GEN 409, STAT 430
Molecular structures including genes and gene products: protein, DNA and RNA structure. Structure determination methods, structural refinement, structure representation, comparison of structures, visualization, and modeling. Molecular and cellular structure from imaging. Analysis and prediction of protein secondary, tertiary, and higher order structure, disorder, protein-protein and protein-nucleic acid interactions, protein localization and function, bridging between molecular and cellular structures. Molecular evolution.

COM S 570: Bioinformatics IV (Systems Biology)
(Cross-listed with BCB, CPR E, GDCB, STAT). (3-0) Cr. 3. S.
Prereq: BCB 567 or COM S 311, COM S 228, GEN 409, STAT 430
Algorithmic and statistical approaches in computational functional genomics and systems biology. Analysis of high throughput biological data obtained using system-wide measurements. Topological analysis, module discovery, and comparative analysis of gene and protein networks. Modeling, analysis, and inference of transcriptional regulatory networks, protein-protein interaction networks, and metabolic networks. Dynamic systems and whole-cell models. Ontology-driven, network based, and probabilistic approaches to information integration.

COM S 572: Principles of Artificial Intelligence
(Dual-listed with COM S 472). (3-1) Cr. 3.
Prereq: COM S 311, COM S 230 or CPR E 310, STAT 330, ENGL 250, SP CM 212, COM S 342 or comparable programming experience; for graduate credit: graduate standing or permission of instructor
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. A research project and a written report is required for students enrolled in Com S 572.

COM S 573: Machine Learning
(3-1) Cr. 3.
Prereq: Graduate standing or permission of instructor
Algorithmic models of learning. Design, analysis, implementation and applications of learning algorithms. Learning of concepts, classification rules, functions, relations, grammars, probability distributions, value functions, models, skills, behaviors and programs. Agents that learn from observation, examples, instruction, induction, deduction, reinforcement and interaction. Computational learning theory. Data mining and knowledge discovery using artificial neural networks, support vector machines, decision trees, Bayesian networks, association rules, dimensionality reduction, feature selection and visualization. Learning from heterogeneous, distributed, dynamic data and knowledge sources. Learning in multi-agent systems. Selected applications in automated knowledge acquisition, pattern recognition, program synthesis, bioinformatics and Internet-based information systems. Oral and written reports.

COM S 574: Introduction to Machine Learning
(Dual-listed with COM S 474). (3-1) Cr. 3.
Prereq: COM S 311, COM S 230 or CPR E 310, STAT 330, MATH 165, ENGL 250, SP CM 212, COM S 342 or comparable programming experience
Basic principles, techniques, and applications of Machine Learning. Design, analysis, implementation, and applications of learning algorithms. Topics include: statistical learning, pattern classification, function approximation, Bayesian learning, linear models, artificial neural networks, support vector machines, decision trees, instance based learning, probabilistic graphical models, unsupervised learning, selected applications in automated knowledge acquisition, pattern recognition, and data mining.

COM S 575: Computational Perception
(Cross-listed with CPR E, HCI). (3-0) Cr. 3. S.
Prereq: Graduate standing or permission of instructor
Statistical and algorithmic methods for sensing, recognizing, and interpreting the activities of people by a computer. Focuses on machine perception techniques that facilitate and augment human-computer interaction. 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.

COM S 577: Problem Solving Techniques for Applied Computer Science
(Dual-listed with COM S 477). (3-0) Cr. 3.
Prereq: COM S 228; COM S 230 or CPR E 310, MATH 166, MATH 207 or MATH 317, or consent of the instructor; for graduate credit: graduate standing or permission of 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. Homogeneous coordinates and transformations, perspective projection, quaternions and rotations, polynomial interpolation, roots of polynomials, resultants, solution of linear and nonlinear equations, approximation, data fitting, Fourier series and fast Fourier transform, linear programming, nonlinear optimization, Lagrange multipliers, parametric and algebraic curves, curvature, Frenet formulas, Bezier curves. Programming components. A scholarly report is required for graduate credit.

COM S 581: Computer Systems Architecture
(Cross-listed with CPR E). (3-0) Cr. 3. F.
Prereq: CPR E 381
Quantitative principles of computer architecture design, instruction set design, processor architecture: pipelining and superscalar design, instruction level parallelism, memory organization: cache and virtual memory systems, multiprocessor architecture, cache coherency, interconnection networks and message routing, I/O devices and peripherals.

COM S 583: Reconfigurable Computing Systems
(Cross-listed with CPR E). (3-0) Cr. 3.
Prereq: Background in computer architecture, design, and organization
Introduction to reconfigurable computing, FPGA technology and architectures, spatial computing architectures such as systolic and bit serial adaptive network architectures, static and dynamic rearrangeable interconnection architectures, processor architectures incorporating reconfigurabiltiy.

COM S 586: Computer Network Architectures
(3-0) Cr. 3.
Prereq: COM S 511, COM S 552 or CPR E 489
Design and implementation of computer communication networks: layered network architectures, local area networks, data link protocols, distributed routing, transport services, network programming interfaces, network applications, error control, flow/congestion control, interconnection of heterogeneous networks, TCP/IP, ATM networks, multimedia communications, IP and application multicast, overlay networks, network security and web computing.

COM S 587: Network Programming, Applications, and Research Issues
(Dual-listed with COM S 487). (3-0) Cr. 3.
Prereq: Com S 352 or CPR E 489 or equivalent; for graduate credit: graduate standing or permission of instructor
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. A written report and an oral presentation is required for students enrolling in Com S 587.

COM S 590: Special Topics
Cr. arr. Repeatable.
Prereq: Permission of instructor
Offered on a satisfactory-fail basis only.

COM S 592: Research Colloquia
Cr. 1.
Prereq: Graduate classification
Attend Computer Science Research Colloquia. Written summary is required. Offered on a satisfactory-fail basis only.

COM S 598: Graduate Internship
Cr. R. Repeatable.
Prereq: Graduate Classification
Supervised internship working in professional settings appropriate to the student's degree program. Academic work under faculty supervision.

COM S 599: Creative Component
Cr. 1-3.
Creative component for nonthesis option of Master of Science degree. Offered on a satisfactory-fail basis only.

Courses for graduate students:

COM S 610: Seminar
Cr. arr.
Offered on a satisfactory-fail basis only.

COM S 611: Advanced Topics in Analysis of Algorithms
(3-0) Cr. 3. Repeatable.
Prereq: COM S 511, COM S 531
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
(3-0) Cr. 3.
Prereq: COM S 511 or COM S 531
The theory of distributed computation. Algorithms, lower bounds and impossibility results. Leader Elections, mutual exlusion, consensus and clock synchronization algorithms. Synchronous, asynchronous and partially synchronous distributed systems models. Shared memory and message passing systems. Fault-tolerance and randomization. Broadcast and multicast. Wait-free object simulations. Distributed shared memory.

COM S 626: Parallel Algorithms for Scientific Applications
(Cross-listed with CPR E). (3-0) Cr. 3.
Prereq: CPR E 526
Algorithm design for high-performance computing. Parallel algorithms for multidimensional tree data structures, space-filling curves, random number generation, graph partitioning and load balancing. Applications to grid and particle-based methods and computational biology.

COM S 631: Advanced Topics in Computational Complexity
(3-0) Cr. 3. Repeatable.
Prereq: COM S 531
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
(3-0) Cr. 3. Repeatable.
Prereq: COM S 531
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
(3-0) Cr. 3.
Prereq: COM S 330
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 multiagent environment, properties of knowledge, agreements, and common knowledge. Reasoning about and representing uncertainty, probabilities, and beliefs. Uncertainty in multiagent 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
(3-0) Cr. 3. Repeatable.
Prereq: COM S 531, COM S 541
Operational and other mathematical models of programming language semantics. Type systems and their soundness. Applications of semantics on areas such as program correctness, language design or translation.

COM S 652: Advanced Topics in Distributed Operating Systems
(3-0) Cr. 3. Repeatable.
Prereq: COM S 552
Concepts and techniques for network and distributed operating systems: Communications protocols, processes and threads, name and object management, synchronization, consistency and replications for consistent distributed data, fault tolerance, protection and security, distributed file systems, design of reliable software, performance analysis.

COM S 657: Advanced Topics in Computer Graphics
(3-0) Cr. 3. Repeatable, maximum of 2 times.
Prereq: COM S 228, I E 557/M E 557/CPR E 557/COM S 557
Modern lighting models: Rendering Equation, Spherical Harmonics, Lafortune, Cook-Torrance. Non-polygonal primitives: volumes, points, particles. Textures: filtering, reflections creation. Graphics hardware: pipeline, performance issues, programmability in vertex and fragment path. Per-pixel lighting. Nonphotorealistic rendering. Radiosity; Ray tracing.

COM S 661: Advanced Topics in Database Systems
(3-0) Cr. 3. Repeatable.
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.
Prereq: COM S 572 or COM S 573 or COM S 472 or COM S 474
Selected topics in Computational Learning Theory (PAC learning, Sample complexity, VC Dimension, Occam Learning, Boosting, active learning, Kolomogorov Complexity, Learning under helpful distributions, Mistake Bound Analysis). Selected topics in Bayesian and Information Theoretic Models (ML, MAP, MDL, MML). Advanced statistical methods for machine learning. Selected topics in reinforcement learning.

COM S 673: Advanced Topics in Computational Intelligence
(3-0) Cr. 3. Repeatable.
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.

COM S 681: Advanced Topics in Computer Architecture
(Cross-listed with CPR E). (3-0) Cr. 3. Alt. S., offered odd-numbered years.
Prereq: CPR E 581. 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 699: Research
Cr. arr. Repeatable.
Prereq: Approval of instructor
Offered on a satisfactory-fail basis only.