

# COMPUTER SCIENCE (COM S)

**Any experimental courses offered by COM S can be found at:**

registrar.iastate.edu/faculty-staff/courses/explistsings/ (<http://www.registrar.iastate.edu/faculty-staff/courses/explistsings/>)

**Courses primarily for undergraduates:**

## COM S 101: Orientation

Cr. R. F.S.

Required orientation class for all incoming students in the Computer Science major. Topics include academic planning and policies, campus resources, and supports. Opportunity to connect with other computer science peers, faculty, alumni, and employers. Offered on a satisfactory-fail basis only. Offered on a satisfactory-fail basis only.

## COM S 103: Computer Literacy and Applications

Cr. 4. F.S.SS.

Introduction to computer literacy and applications. Literacy: Impact of computer technology in today's societies, hardware, software, software programming, database and information systems, communication and networks, digital media technology, computer security and safety, ethics and privacy. Applications: In-depth hands-on experience with the operating systems, Microsoft word processing, spreadsheets, database management and presentation software. No prior computer experience necessary. Offered online only.

## COM S 104: Brief Introduction to Computer Programming for Non-Majors (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 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. F.S.

Introduction to web programming basics. Fundamentals of developing web pages using a comprehensive web development life cycle. Learn to design and code practical real-world homepage programs and earn adequate experience with current web design techniques such as HTML5 and cascading style sheets. Students also learn additional programming languages including JavaScript, jQuery, PHP, SQL, and MySQL. Strategies for accessibility, usability and search engine optimization. No prior computer programming experience necessary.

## COM S 107: Windows Application 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 108: Applied Computer Programming for Non-Majors

(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. F.S.SS.

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 S E and CPR E majors.

## COM S 127: Introduction to Computer Programming

(3-2) Cr. 4. F.S.

*Prereq: Credit or Enrollment in MATH 140 or higher*

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.

**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.

**COM S 207: Fundamentals of Computer Programming**

(Cross-listed with MIS). (3-1) Cr. 3. F.S.SS.

*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. 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: Object-oriented Programming**

(3-2) Cr. 4. F.S.SS.

*Prereq: Credit or Enrollment in MATH 143 or higher; COM S 127 or CPR E 185 or S E 185 or E E 285 or DS 201*

Computer programming using objects as the mechanism for modularity, abstraction, and code reuse. Instance variables, methods, and encapsulation. Review of control structures for conditionals and iteration. Developing algorithms on strings, arrays, and lists. Recursion, searching, and sorting. Text parsing and file I/O. Interfaces, inheritance, polymorphism, and abstract classes. Exception handling. Tools for unit testing and debugging. Emphasis on a disciplined approach to specification, code development, and testing. 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. F.S.SS.

*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, and generics. Abstract data type specification and correctness. Collections including lists, stacks, queues, trees, heaps, maps, hash tables, and graphs. Big-O notation and algorithm analysis. Searching and sorting. Graph search and shortest path algorithms. Emphasis on object-oriented design, writing and documenting medium-sized programs. This course is designed for majors.

**COM S 230: Discrete Computational Structures**

(Cross-listed with MATH). (3-1) Cr. 3. F.S.SS.

*Prereq: Minimum of C- in COM S 227 and MATH 165; ENGL 150*

Concepts in discrete mathematics as applied to computer science. Logic, set theory, functions, relations, combinatorics, discrete probability, graph theory and number theory. Proof techniques, induction and recursion.

**COM S 252: Linux Operating System Essentials**

(3-0) Cr. 3. F.

*Prereq: CPR E 185 or S E 185 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. F.S.SS.

*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. F.S.

*Prereq: Permission of instructor*

No more than 6 credits of Com S 290 or Com S 290H may be counted toward graduation.

**COM S 295: Programming-based problem solving practices**

Cr. 1. Repeatable, maximum of 3 times. F.S.

*Prereq: COM S 207 or COM S 227*

Basics of problem solving using programming techniques. Development and implementation of simple to advanced data structures and algorithms, evaluation of problem difficulty, design and implementation of solutions, debugging, and working under time pressure. Offered on a satisfactory-fail basis only.

**COM S 309: Software Development Practices**

(Cross-listed with S E). (3-1) Cr. 3. F.S.

*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: Introduction to the Design and Analysis of Algorithms**

(3-1) Cr. 3. F.S.SS.

*Prereq: Minimum of C- in COM S 228; MATH 166, ENGL 150; COM S 230 or CPR E 310*

Basic techniques for design and analysis of algorithms. Sorting, searching, graph algorithms, string matching, and NP-completeness. Design techniques such as dynamic programming, divide and conquer, greedy method, and approximation. Asymptotic, worst-case, average-case and amortized analyses. Topics from advanced data structures such as balanced trees and hashing. Programming projects.

**COM S 319: Construction of User Interfaces**

(Cross-listed with S E). (3-0) Cr. 3. F.S.

*Prereq: COM S 228*

Overview of user interface design. Evaluation and testing of user interfaces. Review of principles of object orientation, object oriented design and analysis using UML in the context of user interface design. Design of windows, menus and commands. Developing Web and Windows-based user-interfaces. Event-driven programming. Introduction to Frameworks and APIs for the construction of user interfaces.

**COM S 321: Introduction to Computer Architecture and Machine-Level Programming**

(3-1) Cr. 3. F.S.

*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 326: C for Programmers**

Cr. 1. F.S.

*Prereq: Minimum of C- in COM S 228; COM S 230 or CPR E 310*

Half-semester course. Design and implementation of libraries and applications in C, for students with prior programming background. Emphasis on differences between C and other languages, including file I/O, string processing, memory management, and buffer overruns. Using build systems, debuggers, and other development tools. Programming projects.

**COM S 327: Advanced Programming Techniques**

(3-0) Cr. 3. F.S.

*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. F.S.

*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. Relations between classes of languages.

**COM S 336: Introduction to Computer Graphics**

(3-0) Cr. 3. F.

*Prereq: COM S 327, CoReq MATH 207 or MATH 317*

Programming interactive computer graphics systems using standard low-level libraries (such as OpenGL or DirectX) with an emphasis on 3D rendering. The graphics pipeline and programmable shaders. Coordinate systems and transformations in two and three dimensions. Homogeneous coordinates, viewing transformations and perspective. Euler angles and quaternions. Visible surface algorithms. Lighting models and shading. Texture mapping, bump mapping, reflection, elementary ray tracing. Offscreen buffers, render-to-texture and related techniques.

**COM S 342: Principles of Programming Languages**

(Cross-listed with S E). (3-1) Cr. 3. F.S.

*Prereq: Minimum of C- in COM S 228 and MATH 165; COM S 230 or CPR E 310*

Study of concepts in programming languages, especially functional programming concepts. Overview of major programming paradigms, their relationship, and tradeoffs among paradigms enabling sound choices of programming language for application-specific development. Programming projects.

**COM S 350: Number Theory**

(Cross-listed with MATH). (3-0) Cr. 3. S.

*Prereq: MATH 201 or COM S 230 or CPR E 310*

Divisibility, integer representations, primes and divisors, linear diophantine equations, congruences, and multiplicative functions. Applications to cryptography. Additional topics, chosen at the discretion of the instructor.

**COM S 352: Introduction to Operating Systems**

(3-1) Cr. 3. F.S.

*Prereq: COM S 321 or CPR E 381; COM S 327 or CPR E 288; ENGL 250*

Survey of operating system, networking and parallel programming issues. Introduction of processes, threads, process synchronization, deadlocks, memory, file systems, networking, security threats and encryption. Programming projects.

**COM S 362: Object-Oriented Analysis and Design**

(Cross-listed with S E). (3-0) Cr. 3. F.S.

*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 Unified Modeling Language. Design Patterns. Group design and programming with large programming projects.

**COM S 363: Introduction to Database Management Systems**

(3-0) Cr. 3. F.S.

*Prereq: Minimum of C- in COM S 228 and MATH 165; ENGL 250*

Relational, object-oriented, semistructured and query languages. SQL, XML, and NO-SQL. Database design using entity-relationship model, data dependencies, and relational database design. Application development in SQL-like languages and general purpose host languages with application program interfaces and a commonly used Object Relational Mapping framework. Web application development. Programming Projects.

**COM S 398: Cooperative Education**

Cr. R. Repeatable. F.S.SS.

*Prereq: Permission of department chair*

Required of all cooperative education students. Students must register for this course prior to commencing each work period.

**COM S 401: Bioinformatics of Sequences**

(Cross-listed with BC BIO, BIOL, GEN). (3-0) Cr. 3. F.

*Prereq: BC BIO 322, basic programming experience (e.g. COM S 127, COM S 227 or permission of instructor). MATH 160 or MATH 165; and STAT 101 or STAT 104; and MATH 166 or STAT 301.*

Application of computer science and statistics to molecular biology with a significant problem-solving component, including hands-on programming using Python to solve a variety of biological problems. String algorithms, sequence alignments, homology search, pattern discovery, genotyping, genome assembly, genome annotation, comparative genomics, protein structure.

**COM S 402: Computer Science Senior Project**

Cr. 2-3. Repeatable, maximum of 6 credits. F.S.

*Prereq: COM S 309, COM S 311, COM S 321, COM S 331, and Senior Classification*

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. F.

*Prereq: COM S 309, COM S 311, COM S 321, COM S 331, and COM S 437, Senior Classification*

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. S.

*Prereq: COM S 402A, Senior Classification*

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**

(0-3) Cr. 3. F.S.

*Prereq: COM S 309, COM S 311, COM S 321, COM S 331, and Senior Classification*

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 406: Bioinformatics of OMICS**

(Cross-listed with BC BIO, BIOL, GEN). (3-0) Cr. 3. S.

*Prereq: BIOL 212*

Introduction to cutting edge OMICS analyses including transcriptome, proteome, metabolome, DNA-protein interactome, protein-protein interactome and methylome. Genomic analysis including transcriptome analysis, cancer genomics, comparative genomics, and regulatory network analysis.

**COM S 407: Applied Formal Methods**

(Dual-listed with COM S 507). (Cross-listed with AER E). Cr. 3. S.

*Prereq: AER E 361 for AER E majors. COM S 311 for COM S majors. AER E 361 or COM S 311, or an equivalent course, plus instructor permission for other majors.*

Introduction to the fundamentals of formal methods, a set of mathematically rigorous techniques for the formal specification, validation, and verification of safety- and security-critical systems. Tools, techniques, and applications of formal methods with an emphasis on real-world use-cases such as enabling autonomous operation. Build experience in writing mathematically analyzable specifications from English operational concepts for real cyberphysical systems, such as aircraft and spacecraft. Review capabilities and limitations of formal methods in the design, verification, and system health management of today's complex systems.

**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 elicitation, requirements analysis fundamentals, requirements specification and communication, and requirements evaluation. Modeling of functional and nonfunctional requirements, traceability, and requirements change management. Case studies and software projects.

**COM S 410: Distributed Development of Software**

(Dual-listed with COM S 510). (3-0) Cr. 3.

*Prereq: COM S 309; COM S 327 or CPR E 288; for graduate credit: graduate standing or permission of instructor*

Teams of students develop software applications in a modern software engineering environment. Importance, effective processes pertaining to team organization, management and communication, and cultural issues of distributed development. Graduate credit requires in-depth study of concepts and oral presentations.

**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 311; STAT 305 or STAT 330 or STAT 341; 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 413: Foundations and Applications of Program Analysis**

(Dual-listed with COM S 513). (3-0) Cr. 3.

*Prereq: COM S 327 or CPR E 288; COM S 342*

Algorithms and tools for automatically reasoning about code and program executions. Theory and foundations related to control flow analysis, dataflow analysis, abstract interpretation, and symbolic execution. Applications of program analysis to bug detection, test input generation, debugging, program repair, specification inference and trustworthy AI engineering. Concepts, algorithms, tools, benchmarks, methodologies for solving problems using program analysis and for preparing research in program analysis.

**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*

Interdisciplinary course designed for students interested in assistive technology, pervasive computing, mobile computing and principles of universal and inclusive design for end users, in particular, the elderly population. Students work in semester-long projects as interdisciplinary teams to apply knowledge obtained from lectures and mutual presentations. Research report and oral presentation required for graduate credit.

**COM S 415: Software System Safety**

(Dual-listed with COM S 515). (3-0) Cr. 3.

*Prereq: COM S 309 or COM S 311; for graduate credit: graduate standing or permission of instructor*

An introduction to the hazard analysis, safety requirements, design, and testing of software for safety-critical and high-dependability systems. Safety analysis techniques, fault identification and recovery, 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*

An introduction to software testing principles and techniques. Test models, test design, test adequacy criteria; regression, integration, and system testing; 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; for graduate credit: graduate standing or permission of instructor*

Introduction to data structures, algorithms, and analysis techniques for computational problems that involve geometry. Convex hulls, line segment intersection, polygon triangulation, 2D linear programming, range queries, point location, arrangements and duality, Voronoi diagrams, Delaunay triangulations, geometric data structures, robot motion planning, visibility graphs. Other selected topics. Programming assignments. Scholarly report required for graduate credit.

**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 or CPR E 310*

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; or permission of instructor.*

Unix, serial programming of scientific applications, OpenMP for shared-memory parallelization. No Unix, Fortran or C experience required.

**COM S 425: High Performance Computing for Scientific and Engineering Applications**

(Cross-listed with CPR E). (2-2) Cr. 3.

*Prereq: COM S 311, 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: Concurrent Programming in Practice**

(3-1) Cr. 3.

*Prereq: COM S 311, COM S 362 or COM S 363, ENGL 250, SP CM 212*

A practical course in concepts, techniques, languages, and frameworks for concurrent and asynchronous systems. Concurrency fundamentals: threads, synchronization locks, waiting and notification, memory visibility. Immutability and thread confinement. Concurrent data structures and utilities, thread pools. Asynchronous programming with callbacks, futures, and message passing. Issues of aliasing, ownership and borrowing. Transactional memory, immutable and versioned data structures. Alternatives to threads and locks: event-driven systems, the actor model, CSP, coroutines. Students will investigate several non-mainstream languages supporting different concurrency models. Oral and written reports.

**COM S 433: Molecular Programming of Nanoscale Devices and Processes**

(Dual-listed with COM S 533). (3-0) Cr. 3.

*Prereq: Minimum of C- in COM S 331 or permission of instructor; for graduate credit: graduate standing or permission of instructor*

Programming, modeling, and analysis of natural and engineered systems at the nanoscale. Topics include chemical reaction networks, strand displacement systems, models of self-assembly, biomolecular origami, and molecular robotics. Emphasis on mathematical methods of describing, simulating, programming, and assessing the computational power of such 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 311 or equivalent; for graduate credit: graduate standing or permission of instructor*

Algorithmic challenges involved in solving computational problems on massive data sets. Probabilistic data structures, Curse of Dimensionality and dimensionality reduction, locality sensitive hashing, similarity measures, matrix decompositions. Optimization problems in massive data analysis. Computational problems that arise in the context of web search, social network analysis, online advertising etc. 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*

Video game programming using current game engine interfaces with real hardware. Particular attention is paid to the development environment, tool chains, 2D graphics, 3D graphics, controllers, memory management, and audio systems.

**COM S 440: Principles and Practice of Compiling**

(Dual-listed with COM S 540). (3-1) Cr. 3.

*Prereq: COM S 331 or COM S 342; COM S 309; ENGL 250; 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 include: lexical, syntactic and semantic analyses, syntax-directed translation, code generation, runtime environment and library support.

**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 453: Privacy Preserving Algorithms and Data Security**

Cr. 3.

*Prereq: COM S 311*

Fundamentals of privacy preserving algorithms, data security, anonymization, and techniques and mechanisms to minimize disclosure of sensitive information while maintaining availability. Theory and fundamentals underpinning measures to evaluate the privacy and availability of data; implementation and deployment of privacy-preserving data operations including pre- and post-randomization techniques, homomorphisms, and secure function evaluation protocols. Theory and practice of the algorithmic limits on data privacy, including the cost in terms of time and space complexity.

**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 or CPR E 308; 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; STAT 305 or STAT 330 or STAT 341; ENGL 250; 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.

**COM S 461: Principles and Internals of Database Systems**

(3-1) Cr. 3.

*Prereq: COM S 311, ENGL 250, SP CM 212*

Database design including entity-relationship model, relational data model, and non-relational data models, data dependency, and normalization. Database management including physical storage, access methods, query processing, and transaction management. Database systems of special purposes such as spatial databases, mobile object databases, and multimedia databases. Introduction to current database research such as cloud data management and Internet information retrieval.

**COM S 472: Principles of Artificial Intelligence**

(Dual-listed with COM S 572). (3-1) Cr. 3.

*Prereq: COM S 311; STAT 330 or STAT 305 or STAT 341; ENGL 250; for graduate credit: graduate standing or permission of instructor*

Basic principles, techniques, and applications of artificial intelligence. Specification, design, implementation, and applications of intelligent agents. Computational models of intelligent behavior, including problem solving, knowledge representation and reasoning, planning, decision making, learning, perception, and communication. Artificial intelligence programming. Term project and written report for graduate credit.

**COM S 474: Introduction to Machine Learning**

(Dual-listed with COM S 574). (3-1) Cr. 3.

*Prereq: COM S 311; STAT 330 or STAT 305 or STAT 341; MATH 165; ENGL 250; for graduate credit: graduate standing or permission of instructor*

Introduction to concepts, tools and techniques of machine learning for applications. Selected machine learning algorithms in practical data mining tasks such as classification, regression, and clustering, e.g., association rules, decision trees, linear models, Bayesian learning, support vector machines, artificial neural networks, instance-based learning, probabilistic graphical models, ensemble learning, and clustering algorithms. Selected applications in data mining and pattern recognition.

**COM S 476: Motion Strategy Algorithms and Applications**

(Dual-listed with COM S 576). Cr. 3.

*Prereq: COM S 311, ENGL 250*

Recent techniques for developing algorithms that automatically generate continuous motions while satisfying geometric constraints. Applications in areas such as robotics and graphical animation. Basic path planning. Kinematics, configuration space, and topological issues. Collision detection. Randomized planning. Nonholonomic systems. Optimal decisions and motion strategies. Coordination of Multiple Bodies. Representing and overcoming uncertainties. Visibility-based motion strategies. Implementation of software that computes motion strategies. Written reports.

**COM S 477: Foundations of Robotics and Computer Vision**

(Dual-listed with COM S 577). (3-0) Cr. 3.

*Prereq: COM S 228; COM S 230 or CPR E 310; MATH 207 or MATH 317, or consent of the instructor; for graduate credit: graduate standing or permission of instructor*

Selected topics in applied mathematics, algorithms, and geometry that have found applications in areas such as robotics and computer vision, as well as computer graphics, geometric modeling, visualization, human-computer interaction and speech recognition, to name a few. Homogeneous coordinates and transformations, perspective projection, rotations in space, quaternions, roots of polynomials and polynomial systems, solution of linear and nonlinear equations, parametric and algebraic curves, curvature, torsion, Frenet formulas, surfaces, fundamental forms, principal curvatures, Gaussian and mean curvatures, geodesics, approximation, Fourier series and fast Fourier transform, linear programming, simplex method, nonlinear optimization, Lagrange multipliers, data fitting, least squares, calculus of variations. Programming components. Scholarly report 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 or CPR E 308*

An introduction to fundamental concepts in the design and implementation of computer communication in both wired and wireless networks, their protocols, and applications. Layered network architecture in the Internet, applications, transport, network, and data link layers and their protocols, Socket API, software-defined 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 308; 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. Written report and oral presentation required for graduate credit.

**COM S 490: Independent Study**

Cr. arr. Repeatable, maximum of 9 credits. F.S.SS.

*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. F.S.

*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 507: Applied Formal Methods**

(Dual-listed with COM S 407). (Cross-listed with AER E). Cr. 3. S.

*Prereq: AER E 361 for AER E majors. COM S 311 for COM S majors. AER E 361 or COM S 311, or an equivalent course, plus instructor permission for other majors.*

Introduction to the fundamentals of formal methods, a set of mathematically rigorous techniques for the formal specification, validation, and verification of safety- and security-critical systems. Tools, techniques, and applications of formal methods with an emphasis on real-world use-cases such as enabling autonomous operation. Build experience in writing mathematically analyzable specifications from English operational concepts for real cyberphysical systems, such as aircraft and spacecraft. Review capabilities and limitations of formal methods in the design, verification, and system health management of today's complex systems.

**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 elicitation, requirements analysis fundamentals, requirements specification and communication, and requirements evaluation. Modeling of functional and nonfunctional requirements, traceability, and requirements change management. Case studies and software projects.

**COM S 510: Distributed Development of Software**

(Dual-listed with COM S 410). (3-0) Cr. 3.

*Prereq: COM S 309; COM S 327 or CPR E 288; for graduate credit: graduate standing or permission of instructor*

Teams of students develop software applications in a modern software engineering environment. Importance, effective processes pertaining to team organization, management and communication, and cultural issues of distributed development. Graduate credit requires in-depth study of concepts and oral presentations.

**COM S 511: 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. Network flows and linear programming. Randomized algorithms. NP-completeness. Approximation algorithms. Fixed-parameter algorithms.

**COM S 512: Formal Methods in Software Engineering**

(Dual-listed with COM S 412). (3-0) Cr. 3.

*Prereq: COM S 311; STAT 305 or STAT 330 or STAT 341; 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 513: Foundations and Applications of Program Analysis**

(Dual-listed with COM S 413). (Cross-listed with CPR E). (3-0) Cr. 3.

*Prereq: COM S 327 or CPR E 288; COM S 342*

Algorithms and tools for automatically reasoning about code and program executions. Theory and foundations related to control flow analysis, dataflow analysis, abstract interpretation, and symbolic execution. Applications of program analysis to bug detection, test input generation, debugging, program repair, specification inference and trustworthy AI engineering. Concepts, algorithms, tools, benchmarks, methodologies for solving problems using program analysis and for preparing research in program analysis.

**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*

Interdisciplinary course designed for students interested in assistive technology, pervasive computing, mobile computing and principles of universal and inclusive design for end users, in particular, the elderly population. Students work in semester-long projects as interdisciplinary teams to apply knowledge obtained from lectures and mutual presentations. Research report and oral presentation required for graduate credit.

**COM S 515: Software System Safety**

(Dual-listed with COM S 415). (3-0) Cr. 3.

*Prereq: COM S 309 or COM S 311; for graduate credit: graduate standing or permission of instructor*

An introduction to the hazard analysis, safety requirements, design, and testing of software for safety-critical and high-dependability systems. Safety analysis techniques, fault identification and recovery, 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; for graduate credit: graduate standing or permission of instructor*

Introduction to data structures, algorithms, and analysis techniques for computational problems that involve geometry. Convex hulls, line segment intersection, polygon triangulation, 2D linear programming, range queries, point location, arrangements and duality, Voronoi diagrams, Delaunay triangulations, geometric data structures, robot motion planning, visibility graphs. Other selected topics. Programming assignments. Scholarly report required for graduate credit.

**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 527: Concurrent Systems**

(Cross-listed with CPR E). Cr. 3.

*Prereq: COM S 352 or CPR E 308*

Fundamentals and advances in concurrent systems in the context of GPUs, TPUs, multicore and HPC systems with specific focus on parallel programming models. Discussion of high-performance computing, GPGPU, scaling deep neural network training and machine learning algorithms, high-performance deep learning, engineering parallel software and and parallel design patterns.

**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. Computational resources such as time, space, and nonuniformity. Complexity classes, computational intractability and completeness. Selected topics from randomness, algorithmic information theory, and logic.

**COM S 533: Molecular Programming of Nanoscale Devices and Processes**

(Dual-listed with COM S 433). (3-0) Cr. 3.

*Prereq: Minimum of C- in COM S 331 or permission of instructor; for graduate credit: graduate standing or permission of instructor*

Programming, modeling, and analysis of natural and engineered systems at the nanoscale. Topics include chemical reaction networks, strand displacement systems, models of self-assembly, biomolecular origami, and molecular robotics. Emphasis on mathematical methods of describing, simulating, programming, and assessing the computational power of such 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 311 or equivalent; for graduate credit: graduate standing or permission of instructor*

Algorithmic challenges involved in solving computational problems on massive data sets. Probabilistic data structures, Curse of Dimensionality and dimensionality reduction, locality sensitive hashing, similarity measures, matrix decompositions. Optimization problems in massive data analysis. Computational problems that arise in the context of web search, social network analysis, online advertising etc. 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 or COM S 342; COM S 309; ENGL 250; 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 include: lexical, syntactic and semantic analyses, syntax-directed translation, code generation, runtime environment and library support.

**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. Alt. F., offered odd-numbered years.

*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 551: Computational Techniques for Genome Assembly and Analysis**

(3-0) Cr. 3.

*Prereq: COM S 311 and some knowledge of programming*

Introduction to a big data research area in bioinformatics. Focus on applying computational techniques to huge genomic sequence data. These techniques include finding optimal sequence alignments, generating genome assemblies, finding genes in genomic sequences, mapping short sequences onto a genome assembly, finding single-nucleotide and structural variations, building phylogenetic trees from genome sequences, and performing genome-wide association studies.

**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. 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. Programming and research projects.

**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 or CPR E 308; 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; STAT 305 or STAT 330 or STAT 341; ENGL 250; 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.

**COM S 556: Analysis Algorithms for Stochastic Models**

(3-0) Cr. 3.

*Prereq: Graduate standing or permission of instructor*

Advanced techniques to specify and study the correctness and timing properties of complex systems and software, with a particular focus on concurrent and distributed behavior. Petri nets and related formalisms to describe discrete-state systems. Decision diagram algorithms for state-space and reachability graph generation, symbolic model checking, and timing analysis. Markov models for exact and approximate probabilistic verification.

**COM S 557: Computer Graphics and Geometric Modeling**

(Cross-listed with CPR E, M E). (3-0) Cr. 3. Alt. F., offered odd-numbered years.

*Prereq: M E 421 or instructor permission*

Fundamentals of computer graphics technology. Data structures. Parametric curve and surface modeling. Solid model representations. Applications in engineering design, analysis, and manufacturing.

**COM S 559: Security and Privacy in Cloud Computing**

(Cross-listed with CPR E). (3-0) Cr. 3.

*Prereq: COM S 352 or CPR E 308; COM S 486 or CPR E 489 or CPR E 530*

Introduction to cloud computing concepts and systems. Security and privacy threats in cloud computing. Practical techniques for cloud computing security. Theoretical and practical solutions for secure outsourcing of data and computation. Oral presentations and research projects.

**COM S 560: Data-Driven Security and Privacy**

(Cross-listed with CPR E, CYBSC). Cr. 3. Alt. S., offered irregularly.

*Prereq: CPR E 531; COM S 474 or COM S 573*

Examination of applications of machine learning and big data techniques to various security and privacy problems, as well as secure and privacy-preserving machine learning algorithms.

**COM S 561: Database Design, Management, and Research**

(3-0) Cr. 3.

*Prereq: COM S 311 or permission of instructor*

Database design including entity-relationship model, relational data model, and non-relational data models, data dependency, and normalization. Database management including physical storage, access methods, query processing, and transaction management. Database systems of special purposes such as spatial databases, mobile object databases, and multimedia databases. Introduction to current database research such as cloud data management and Internet information retrieval. Significant work on reading and presentation of research publications.

**COM S 567: Bioinformatics Algorithms**

(Cross-listed with BCB, CPR E). (3-0) Cr. 3.

*Prereq: COM S 228; COM S 230; 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: Statistical Bioinformatics**

(Cross-listed with BCB, GDCB, STAT). (3-0) Cr. 3. S.

*Prereq: BCB 567 or (BIOL 315 and one of STAT 430 or STAT 483 or STAT 583), 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: 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: 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 or STAT 483 or STAT 583*

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; STAT 330 or STAT 305 or STAT 341; ENGL 250; for graduate credit: graduate standing or permission of instructor*

Basic principles, techniques, and applications of artificial intelligence. Specification, design, implementation, and applications of intelligent agents. Computational models of intelligent behavior, including problem solving, knowledge representation and reasoning, planning, decision making, learning, perception, and communication. Artificial intelligence programming. Term project and written report for graduate credit.

**COM S 573: Machine Learning**

(3-1) Cr. 3.

*Prereq: Graduate standing or permission of instructor*

Basic principles, techniques, and applications of machine learning. Design, analysis, theoretical foundation, implementation, and applications of learning algorithms. Selected machine learning techniques in supervised learning, unsupervised learning, and reinforcement learning, including Bayesian decision theory, computational learning theory, decision trees, linear models, support vector machines, artificial neural networks, instance-based learning, probabilistic graphical models, ensemble learning, clustering algorithms, dimensionality reduction and feature selection. Selected applications in data mining and pattern recognition.

**COM S 574: Introduction to Machine Learning**

(Dual-listed with COM S 474). (3-1) Cr. 3.

*Prereq: COM S 311; STAT 330 or STAT 305 or STAT 341; MATH 165; ENGL 250; for graduate credit: graduate standing or permission of instructor*

Introduction to concepts, tools and techniques of machine learning for applications. Selected machine learning algorithms in practical data mining tasks such as classification, regression, and clustering, e.g., association rules, decision trees, linear models, Bayesian learning, support vector machines, artificial neural networks, instance-based learning, probabilistic graphical models, ensemble learning, and clustering algorithms. Selected applications in data mining and pattern recognition.

**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 576: Motion Strategy Algorithms and Applications**

(Dual-listed with COM S 476). Cr. 3.

*Prereq: COM S 311, ENGL 250*

Recent techniques for developing algorithms that automatically generate continuous motions while satisfying geometric constraints. Applications in areas such as robotics and graphical animation. Basic path planning. Kinematics, configuration space, and topological issues. Collision detection. Randomized planning. Nonholonomic systems. Optimal decisions and motion strategies. Coordination of Multiple Bodies. Representing and overcoming uncertainties. Visibility-based motion strategies. Implementation of software that computes motion strategies. Written reports.

**COM S 577: Foundations of Robotics and Computer Vision**

(Dual-listed with COM S 477). (3-0) Cr. 3.

*Prereq: COM S 228; COM S 230 or CPR E 310; MATH 207 or MATH 317, or consent of the instructor; for graduate credit: graduate standing or permission of instructor*

Selected topics in applied mathematics, algorithms, and geometry that have found applications in areas such as robotics and computer vision, as well as computer graphics, geometric modeling, visualization, human-computer interaction and speech recognition, to name a few. Homogeneous coordinates and transformations, perspective projection, rotations in space, quaternions, roots of polynomials and polynomial systems, solution of linear and nonlinear equations, parametric and algebraic curves, curvature, torsion, Frenet formulas, surfaces, fundamental forms, principal curvatures, Gaussian and mean curvatures, geodesics, approximation, Fourier series and fast Fourier transform, linear programming, simplex method, nonlinear optimization, Lagrange multipliers, data fitting, least squares, calculus of variations. Programming components. Scholarly report required for graduate credit.

**COM S 578: Optimization for Machine Learning**

Cr. 3.

*Prereq: Com S 472, Com S 474, or instructor permission.*

Advances in optimization theory and algorithms with evolving applications for machine learning. Theoretical foundations at the intersection of optimization and machine learning to conduct advanced research in machine learning and related fields. Emphasis on proof techniques for optimization algorithms in machine learning.

**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 reconfigurability.

**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, software-defined networking and network security.

**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 308; 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. Written report and oral presentation required for graduate credit.

**COM S 590: Special Topics**

Cr. arr. Repeatable.

*Prereq: Permission of instructor*

Special Topics in Computer Science.

**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.

Seminar in Computer Science. 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, but are not limited to, graph algorithms, geometric algorithms, approximation algorithms, fixed-parameter algorithms, randomized algorithms and advanced data structures. Content varies by semester.

**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. Fundamental problems including consensus, leader election, mutual exclusion and clock synchronization. Synchronous, asynchronous and partially synchronous distributed systems models. Shared memory and message passing systems. Fault-tolerance and randomization. Wait-free object simulations. Distributed shared memory. Special topics vary from year to year.

**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. Novel models of computation emerging in a rapidly changing field.

**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. Novel models of computation emerging in a rapidly changing field.

**COM S 634: Theory of Games, Knowledge and Uncertainty**

(3-0) Cr. 3.

*Prereq: COM S 230*

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, and distributed file systems. Research project.

**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 665: Advanced Topics in Software Engineering**

Cr. 3. Repeatable, maximum of 6 credits. Alt. F., offered irregularly. Alt. S., offered irregularly.

*Prereq: COM S 511*

Advanced topics on software repository analysis, data mining and software engineering, software engineering for context-aware and situation-aware computing, distributed development, product lines, safety, security, and reliability, and traceability. Content varies by semester. Maximum 6 credits of COM S 665 may apply toward graduation.

**COM S 665A: Advanced Topics in Software Engineering: Foundations**

Cr. 3. Repeatable, maximum of 6 credits. Alt. F., offered irregularly. Alt. S., offered irregularly.

*Prereq: COM S 511*

Advanced topics on theoretical and technical foundations in Software Engineering. Content varies by semester. Maximum 6 credits of COM S 665 may apply toward graduation.

**COM S 665B: Advanced Topics in Software Engineering: Empirical**

Cr. 3. Repeatable, maximum of 6 credits.

*Prereq: COM S 511*

Advanced topics on empirical studies on human factors and other software engineering topics. Content varies by semester. Maximum 6 credits of COM S 665 may apply toward graduation.

**COM S 672: 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*

Selected topics in probabilistic graphical models, causal inference, semantic web, information retrieval, natural language processing, knowledge representation and reasoning, deep learning, embedding, distributed learning, incremental learning, multi-task learning, multi-strategy learning, multi-relational learning, modeling the internet and the web, automated scientific discovery, neural and cognitive modeling. Advanced applications of artificial intelligence in bioinformatics, distributed systems, natural language, multimedia data, decision making, robotics, and more.

**COM S 673: 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*

Advanced topics in machine learning. Selected topics in computational learning theory, Bayesian and information theoretic models (ML, MAP, MDL, MML), probabilistic graphical models, statistical relational learning, reinforcement learning, and deep learning.

**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*

Research. Offered on a satisfactory-fail basis only.