COMPUTER SCIENCE (CSCI-GA)

CSCI-GA 1133 PAC I (4 Credits)
Typically offered Fall
An accelerated introduction to the fundamental concepts of computer science for students who lack a formal background in the field. Topics include algorithm design and program develop- ment; data types; control structures; subprograms and parameter passing; recursion; data structures; searching and sorting; dynamic storage allocation and pointers; abstract data types, such as stacks, queues, lists, and tree structures; generic packages; and an introduction to the principles of object-oriented programming. Packages are emphasized as a means to develop skills in effective software design and development. Students should expect an average of 12-16 hours of programming and related course work per week.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 1144 PAC II (4 Credits)
Typically offered Spring
Builds directly on the foundation developed in PAC I and extends this two ways: down, to the level of machine architecture, and up, to the higher levels of programming abstraction, using Java and object-oriented programming techniques.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 1170 Fundamental Algorithms (3 Credits)
Typically offered Fall, Spring, and Summer terms
Reviews a number of important algorithms, with emphasis on correctness and efficiency. The topics covered include solution of recurrence equations, sorting algorithms, selection, binary search trees and balanced-tree strategies, tree traversal, partitioning, graphs, spanning trees, shortest paths, connectivity, depth-first and breadth-first search, dynamic programming, and divide-and-conquer techniques.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 1180 Mathematical Techniques for Cs Applications (3 Credits)
Typically offered Fall
An introduction to theory, computational techniques, and applications of linear algebra, probability and statistics. These three areas of continuous mathematics are critical in many parts of computer science, including machine learning, scientific computing, computer vision, computational biology, natural language processing, and computer graphics. The course teaches a specialized language for mathematical computation, such as Matlab, and discusses how the language can be used for computation and for graphical output. No prior knowledge of linear algebra, probability, or statistics is assumed.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2110 Programming Languages (3 Credits)
Typically offered Fall, Spring, and Summer terms
Discusses the design, use, and implementation of imperative, object-oriented, and functional programming languages. The topics covered include scoping, type systems, control structures, functions, modules, object orientation, exception handling, and concurrency. A variety of languages are studied, including C++, Java, Ada, Lisp, and ML, and concepts are reinforced by programming exercises.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2112 Scientific Computing (3 Credits)
Typically offered Fall and Spring
A practical introduction to scientific computing covering theory and basic algorithms together with use of visualization tools and principles behind reliable, efficient, and accurate software. Students program in C/C++ or MatLab. Specific topics include IEEE arithmetic, conditioning and error analysis, classical numerical analysis (finite difference and integration formulas, etc.), numerical linear algebra, optimization and nonlinear equations, ordinary differential equations, and basic Monte Carlo.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2130 Compiler Construction (3 Credits)
Typically offered Fall and Spring
This is a capstone course based on compilers and modern programming languages. The topics covered include structure of one-pass and multiple-pass compilers; symbol table management; lexical analysis; traditional and automated parsing techniques, including recursive descent and LR parsing; syntax-directed translation and semantic analysis; run-time storage management; intermediate code generation; introduction to optimization; and code generation. The course includes a special compiler-related capstone project, which ties together concepts of algorithms, theory (formal languages), programming languages, software engineering, computer architecture, and other subjects covered in the M.S. curriculum. This project requires a substantial semester-long programming effort, such as construction of a language compilation or translation system that includes lexical and syntactic analyzers, a type checker, and a code generator..
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2233 Computer Systems Design (3 Credits)
Typically offered occasionally
Gives students whose interest is in software an introduction to hardware and the logical design of digital computers. Topics include design of basic logic modules and arithmetic units; fixed and microprogrammable control structures; computer architecture; memory organization; and input-output organization.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2243 High Performance Computer Arch (3 Credits)
Typically offered occasionally
Measures of architecture quality. Memory system techniques: cache-memory design techniques, models of program behavior, cache and virtual memory structures. Pipeline computers, vector processors, and array processors. Multiprocessors, synchronization, cache coherence. Parallelization techniques, efficient parallel software.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2245 UNIX Tools (3 Credits)
Typically offered not typically offered
Brief history of the UNIX operating system: basic utilities (mail, editors); shells; windowing systems; shell programming using UNIX tools (awk, set, grep, tar); networking tools; news readers; etiquette and Internet databases and facilities; C programming tools; UNIX-based systems programming; desktop publishing tools; visualization systems; symbolic algebra tools; and system administration.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2246 Open Source Tools (3 Credits)
Typically offered Fall
This course covers a brief history and philosophy of open source software, followed by an in-depth look at open source tools intended for developers. In particular, we will present an overview of the Linux operating system, command line tools (find, grep, sed), programming tools (GIT, Eclipse, DTrace), web and database tools (Apache, MySQL), and system administration tools. We will also cover scripting languages such as shell and Python.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2250 Operating Systems (3 Credits)
Typically offered Fall, Spring, and Summer terms
The topics covered include a review of linkers and loaders and the high-level design of key operating systems concepts such as process scheduling and synchronization; deadlocks and their prevention; memory management, including (demand) paging and segmentation; and I/O and file systems, with examples from Unix/Linux and Windows. Programming assignments may require C, C++, Java, or C#.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2262 Data Comm & Networks (3 Credits)
Typically offered Fall and Spring
This course teaches the design and implementation techniques essential for engineering robust networks. Topics include networking principles, Transmission Control Protocol/Internet Protocol, naming and addressing (Domain Name System), data encoding/decoding techniques, link layer protocols, routing protocols, transport layer services, congestion control, quality of service, network services, programmable routers and overlay networks.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2270 Computer Graphics (3 Credits)
Typically offered Fall and Spring
Problems and objectives of computer graphics. Vector, curve, and character generation. Interactive display devices. Construction of hierarchical image list. Graphic data structures and graphics languages. Hidden-line problems; windowing, shading, and perspective projection. Curved surface generation display.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2271 Computer Vision (3 Credits)
Typically offered Fall and Spring
Basic techniques of computer vision and image processing. General algorithms for image understanding problems. Study of binary image processing, edge detection, feature extraction, motion estimation, color processing, stereo vision, and elementary object recognition. Mathematical, signal processing, and image processing tools. Relation of computer vision algorithms to the human visual system.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2274 Adv Computer Graphics (3 Credits)
Typically offered Fall and Spring
This is a projects course based on computer graphic tools. Projects may be in one of the following areas: computer animation, gaming, geometric modeling, motion capture, computational photography, physically-based simulation, scientific visualization, user interfaces, etc. Not all areas are available every semester; the choice of areas is determined by the instructor. Typically, instructors require several of these elements: project proposal, literature review, interim report, project demo, final report.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2280 User Interfaces (3 Credits)
Typically offered not typically offered
Review of some of the basic principles and history of user interfaces. Building an interactive window system from the ground up, starting with a generic portable graphics base. Examination of future and emerging (nontraditional) user interfaces, including virtual reality and immersive environments.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2340 Elem of Discrete Math (3 Credits)
Typically offered Summer term
Introduction to the central mathematical concepts that arise in computer science. Emphasis is on proof and abstraction. Topics include proof techniques; combinatorics; sets, functions, and relations; discrete structures; order of magnitude analysis; formal logic; formal languages and automata.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2390 Logic in Computer SCI (3 Credits)
Typically offered occasionally
A beginning graduate-level course in mathematical logic with motivation provided by applications in computer science. There are no formal prerequisites, but the pace of the class requires that students can cope with a significant level of mathematical sophistication. Topics include propositional and first-order logic; soundness, completeness, and compactness of first-order logic; first-order theories; undecidability and Godel's incompleteness theorem; and an introduction to other logics such as second-order and temporal logic.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2433 Database Systems (3 Credits)
Typically offered Fall and Spring
Database system architecture. Modeling an application and logical database design. The relational model and relational data definition and data manipulation languages. Design of relational databases and normalization theory. Physical database design. Concurrency and recovery. Query processing and optimization.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2434 Advanced Topics Database Systems (3 Credits)
Typically offered Fall
This is a capstone course emphasizing large-scale database systems. This course studies the internals of database systems as an introduction to research and as a basis for rational performance tuning. Topics include concurrency control, fault tolerance, operating system interactions, query processing, and principles of tuning. Database capstone projects involve topics such as design, concurrency control, interactions, and tuning. These projects include some or all of the following elements: formation of a small team, project proposal, literature review, interim report, project presentation, and final report.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2436 Realtime and Big Data Analytics (3 Credits)
This course introduces architectures and technologies at the foundation of the Big Data movement. These technologies facilitate scalable management and processing of vast quantities of data collected through realtime and near realtime sensing. We explore tools enabling the acquisition of data in the social domain and the fusion of those data when in flight and at rest using Hadoop and Hadoop-related tools. The material covered in this course aligns with the prevailing state of the art in Big Data technologies, which continues to be a rapidly evolving landscape as new technologies emerge and existing ones evolve and mature. Students are required to complete weekly reading and/or programming assignments and demonstrate mastery of course topics by designing, developing, and demonstrating an analytics project of their choosing. Class time will be set aside for project proposal and final demo. This is an introductory Hadoop course intended for students who have no experience with the Apache Hadoop ecosystem of tools and frameworks
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2437 Big Data Application Development (3 Credits)
This is an introductory course on Spark programming, Spark architecture, Spark SQL, Spark Streaming, and integration of Spark with the Hadoop ecosystem for developing Big Data analytics applications. The course project can be completed with Scala or Python, and Spark. This course covers technologies that integrate well with Spark in the creation of Big Data analytics applications. Students are required to complete weekly reading and programming assignments and demonstrate mastery of course topics by developing a final project using Spark.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2440 Software Engineering (3 Credits)
Typically offered Spring
Presents modern software engineering techniques. Examines the software life cycle, including software specification, design, implementation, testing, and maintenance. Object-oriented design methods.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2560 Artificial Intelligence (3 Credits)
Typically offered Fall and Spring
There are many cognitive tasks that people do easily and almost unconsciously but that have proven extremely difficult to program on a computer. Artificial intelligence is the problem of developing computer systems that can carry out these tasks. This course covers problem solving and state space search; automated reasoning; probabilistic reasoning; planning; and knowledge representation
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2565 Machine Learning (3 Credits)
Typically offered all terms
This course covers a wide variety of topics in machine learning, pattern recognition, statistical modeling, and neural computation. The course covers the mathematical methods and theoretical aspects but primarily focuses on algorithmic and practical issues.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2566 Foundations of Machine Learning (3 Credits)
Typically offered Fall
This course introduces the fundamental concepts and methods of machine learning, including the description and analysis of several modern algorithms, their theoretical basis, and the illustration of their applications. Many of the algorithms described have been successfully used in text and speech processing, bioinformatics, and other areas in real-world products and services. The main topics covered are probability and general bounds; PAC model; VC dimension; perceptron, Winnow; support vector machines (SVMs); kernel methods; decision trees; boosting; regression problems and algorithms; ranking problems and algorithms; halving algorithm, weighted majority algorithm, mistake bounds; learning automata, Angluin-type algorithms; and reinforcement learning, Markov decision processes (MDPs)
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2572 Deep Learning (3 Credits)
Typically offered all terms
This course seeks to move close to Artificial Intelligence by learning multiple levels of representation and abstraction useful in making sense of data including images, sound, and text. This course provides an introduction to deep learning and an overview of state-of-the art topics. The pre-requisites include: DS-GA 1001 Intro to Data Science or a graduate-level machine learning course.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2580 Web Search Engines (3 Credits)
Typically offered Fall and Spring
Discusses the design of general Discusses the design of general and specialized Web search engines and the extraction of information from the results of Web search engines. Topics include Web crawlers, database design, query language, relevance ranking, document similarity and clustering, the ?invisible? Web, specialized search engines, evaluation, natural language processing, data mining applied to the Web, and multimedia retrieval.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2590 Natural Lang Processing (3 Credits)
Typically offered Spring
Survey of the techniques used for processing natural language. Syntactic analysis: major syntactic structures of English; alternative formalisms for natural language grammar; parsing algorithms; analyzing coordinate conjunction; parsing with graded acceptability. Semantic analysis: meaning representations; analysis of quantificational structure; semantic constraints; anaphora resolution; analysis of sentence fragments. Analysis of discourse and dialog. Text generation. Students get some experience using a natural language parser and a natural language query interface. Brief weekly written assignments and a term project involving a mixture of library research and programming (mostly in LISP).
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2591 Adv Natural Lang Process (3 Credits)
Typically offered occasionally
One of the roadblocks to improving the performance of natural language systems is the difficulty of acquiring large amounts of knowledge about the properties of language: which words can meaningfully combine in linguistic structures and how words are semantically related. The recent availability of very large machine-readable corpora has sparked increased interest in acquiring this information automatically from text, using a combination of symbolic and statistical analysis. This course reviews some of the recent work in this area, including the following topics: statistical models of language; entropy and perplexity; n-gram word models: acquisition and smoothing, part-of-speech models; finite state models: hidden Markov models, acquisition procedures; probabilistic context-free grammars: acquisition procedures; semantic models: word-concurrence, word classes; applications in information retrieval, speech recognition, and machine translation.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2620 Networks and Mobile Systems (3 Credits)
Typically offered Spring
This is a graduate level course on computer networks and mobile systems. This is a capstone class which will be programming and network analysis intensive. In this class, we will teach the design and implementation techniques essential for engineering both robust networks and mobile systems. The class will cover two sets of topics. The first part of the class will focus on core networking concepts pertaining to the Internet including Internet architecture, routing protocols, congestion control, software defined networks, overlay, P2P and data center networking. The second part will cover concepts in wireless and mobile systems where we will cover topics pertaining to wireless MAC protocols, cellular networks, mobile cloud computing, mobile sensing and Internet of things. Prerequisites: A course in undergraduate networks and/or operating systems; programming experience in C/C++ or Java is helpful for the final project.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2621 Distributed Systems (3 Credits)
Large-scale distributed systems lie at the core of application domains such as cloud computing, internet of things, large multiplayer games, etc. These application domains make use of systems such as distributed databases (e.g., Google's Spanner, Amazon's Dynamo, etc.), large scale analytics frameworks (e.g., Hadoop, Spark and TensorFlow), distributed locking systems (e.g., Paxos, etcd, etc.), and others. In this class we will look at how to construct these distributed systems, in particular looking at why this is more complex than building applications running on a single machine, and present abstraction and design techniques for building distributed systems. We will focus on a solving a variety of common problems in these systems including consensus, consistency, naming, fault tolerance, etc. This is a graduate-level course, but undergraduates are welcome! The course itself will consist of a series of lectures and will require reading research papers. This class satisfies the Ph.D. breadth requirement in Systems and also serves as a M.S. capstone course. Prerequisites: 1. Undergraduate Operating Systems or Computer Systems Organization; 2. programming experience; 3. mathematical maturity or the willingness to work through simple proofs.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2630 Foundations of Networks and Mobile Systems (3 Credits)
Typically offered Fall
The class will begin with introductory concepts of network protocols across different layers of the network stack including routing, transport, naming, addressing and connect them to the core building blocks of the Internet. The second part of the class will focus on networking concepts in the evolution of Webbased systems, providing an introduction to datacenter networks, clouds, middle boxes (proxies, firewalls) and next generation networks. The third part of the class will focus on the fundamental concepts in wireless networks, cellular networks and mobile devices with a specific focus on mobile programming and applications, 802.11 evolution, mobile services and the evolution of cellular networks.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2631 Distributed Computing (3 Credits)
Typically offered not typically offered
Concepts underlying distributed systems: synchronization, communication, fault tolerance, and performance. Examined from three points of view: (1) problems, appropriate assumptions, and algorithmic solutions; (2) linguistic constructs; and (3) some typical systems.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2810 Design and Innovation (3 Credits)
Typically offered Summer term
This is an intensive idea-incubation, computing-centric design class where students will be exposed to a spectrum of tech challenges, latest and future technology trends (using case studies) and will need to iteratively propose and refine bold computing centric ideas for real problems. Students will also initiate the process of translating their ideas to initial prototypes. Right from the beginning, students in the class will form small teams where each team will work on a single project idea that solves an important real world problem. Students will be provided exposure to basic tools and platforms that can be leveraged by individual teams in their project design and implementation. In addition to idea incubation, the class will provide a beginner's background to tech startups including: product development cycle, how to pitch your idea, product market fit, fundraising and venture capital, customer discovery. Teams are expected to constantly interact with other teams to discuss and exchange ideas.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2820 DevOps and Agile Methodologies (3 Credits)
Typically offered Fall
This course uses a project based learning approach towards the study of DevOps as a cultural change in Information Technology organizations, and the supporting development tools and automation technologies required to implement it successfully. Students study the principles of DevOps, and as part of an agile development team, each student is involved in planning, designing, building, testing, and deploying one or more cloud native microservices into a Platform as a Service cloud by utilizing a DevOps Pipeline that they will create.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2830 Lean Launch Pad (3 Credits)
Typically offered Fall
This course is based on Steve Blank's Lean LaunchPad curriculum. This course provides real world, hands on learning on what it's like to actually start a high tech company. This is a practical class – essentially a lab, not a theory or "book" class. The goal, within the constraints of a classroom and a limited amount of time, is to create an entrepreneurial experience for students with all of the pressures and demands of the real world in an early stage start up. Students will talk to customers, partners, competitors, as they encounter the chaos and uncertainty of how a startup actually works. They will work in teams learning how to turn a great idea into a great company. They will learn how to use a business model to brainstorm each part of a company and customer development to get out of the classroom to see whether anyone would want/use their product. Finally, based on the customer and market feedback gathered, students will use agile development to rapidly iterate their products to build something customers would actually use and buy. Each day will be a new adventure outside the classroom as students test each part of their business model and then share the hard earned knowledge with the rest of the class.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2840 Entrepreneurship Capstone (3 Credits)
Typically offered Spring
The Entrepreneurship Capstone class is a 15week bootcamp class on entrepreneurship. This class aims to translate experiences of students on idea incubation, prototype development and Lean Launchpad towards entrepreneurship. Students will learn how to pitch their ideas to VCs, focus on business development and translate their prototypes to an evolving MVP.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2930 Advanced Topics in Applied Mat (3 Credits)
Typically offered occasionally
This course will be an introduction to several numerical methods known as "fast analysis-based algorithms," including fast multiple methods, butterfly algorithms, hierarchical matrix compression and fast direct solvers. These algorithms can be used to solve many of the PDE governing classical mathematical physics, namely electromagnetics, acoustics, heat diffusion, and fluid dynamics. There are additional applications in signal processing and data analysis. Methods from potential theory, applied analysis, functional analysis, numerical linear algebra, complex analysis, and asymptotic analysis are central to the construction of almost all such algorithms.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 2965 Heuristic Problm Solving (3 Credits)
Typically offered Fall
This course revolves around several problems new to computer science (derived from games or puzzles in columns for Dr. Dobb's Journal, Scientific American, and elsewhere). The idea is to train students to face a new problem, read relevant literature, and come up with a solution. The solution entails winning a contest against other solutions. The winner receives candy. The best solutions become part of an evolving "Omniheurist" Web site that is expected to get many visitors over the years. The course is for highly motivated, mathematically adept students. It is open to supported Ph.D. students and well-qualified master's students. Class size has been around 10 in the past, and instructor and students have all gotten to know one another very well. Algorithmic and programming knowledge is the main prerequisite. It also helps to be familiar with a rapid prototyping language such as Matlab, Mathematica, K, or Python, or to be completely fluent in some other language. Prerequisites: CSCI-GA 1170 and an ability to prototype algorithms rapidly.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 3033 Spec Top Computer SCI: (3 Credits)
Typically offered Fall, Spring, and Summer terms
Topics vary by semester
Grading: GSAS Graded
Repeatable for additional credit: Yes

CSCI-GA 3110 Honors Programming Lang (4 Credits)
Typically offered Spring
In-depth examination of the four major categories of programming languages: imperative, object-oriented, functional, and logic languages. The specific languages covered include Ada, C++, LISP, ML, Prolog, and SETL. Fundamental issues of programming languages, such as type systems, scoping, concurrency, modularization, control flow, and semantics, are discussed.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 3130 Honors Compilers (4 Credits)
Typically offered occasionally
Lexical scanning and scanner generation from regular expressions; LL, LR, and universal parser generation from context-free grammars; syntax-directed translation and attribute grammars; type and general semantic analysis; code generation, peephole optimization, and register allocation; and global program analysis and optimization. Provides experience using a variety of advanced language systems and experimental system prototypes.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 3140 Abstract Interpretation (3 Credits)
Prerequisites: Some facility with logic, sets, linear algebra, graphs, formal languages, mathematical proofs, and programming languages is expected. Many numerical or symbolic problems are very hard or even undecidable meaning that no computer algorithm can solve them exactly in reasonable time and memory space. This course introduces the latest techniques in abstract interpretation, a powerful framework for automatically calculating approximate solutions of such difficult problems. The main emphasis will be on software verification and automatic analysis with a wide range of applications from compilers to social networks.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 3205 Applied Cryptography & Network Security (3 Credits)
Typically offered Fall and Spring
This course first introduces the fundamental mathematical cryptographic algorithms, focusing on those that are used in current systems. To the extent feasible, the mathematical properties of the cryptographic algorithms are justified, using elementary mathematical tools. Second, actual security mechanisms and protocols, mainly those employed for network traffic that rely on the previously introduced cryptographic algorithms, are presented. The topics covered will include introduction to basic number- theoretical properties, public/private and symmetric key systems, secure hash functions, digital signature standards, digital certificates, IP security, email security, web security, and stand-alone computer privacy and security tools.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 3210 Intro to Cryptography (3 Credits)
Typically offered Fall and Spring
The primary focus of this course is on definitions and constructions of various cryptographic objects, such as pseudorandom generators, encryption schemes, digital signature schemes, message authentication codes, block ciphers, and others, time permitting. The class tries to understand what security properties are desirable in such objects, how to properly define these properties, and how to design objects that satisfy them. Once a good definition is established for a particular object, the emphasis will be on constructing examples that provably satisfy the definition. Thus, a main prerequisite of this course is mathematical maturity and a certain comfort level with proofs. Secondary topics, covered only briefly, are current cryptographic practice and the history of cryptography and cryptanalysis.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 3220 Advanced Cryptography (3 Credits)
Typically offered occasionally
Basics of computational number theory for cryptography. Identification protocols. Digital signatures. Public-key encryption. Additional selected topics.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 3250 Honors Operating Systems (4 Credits)
Typically offered Fall
Operating-system structure. Processes. Process synchronization. Language mechanisms for concurrency. Deadlocks: modeling, prevention, avoidance, and recovery. Memory management. File-system interface. Secondary storage. Distributed systems: layered system design, managing distributed processes, distributed shared memory, fault-tolerance. CPU scheduling. Queuing and performance: analysis of single M/M/1 queue and others. Protection and security. Advanced security concepts: threat monitoring, encryption, and public keys.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 3520 Honors Analy of Algo (4 Credits)
Typically offered Fall
Design of algorithms and data structures. Review of searching, sorting, and fundamental graph algorithms. In-depth analysis of algorithmic complexity, including advanced topics on recurrence equations and NP-complete problems. Advanced topics on lower bounds, randomized algorithms, amortized algorithms, and data structure design as applied to union-find, pattern matching, polynomial arithmetic, network flow, and matching.
Grading: GSAS Graded
Repeatable for additional credit: No

CSCI-GA 3812 Info Tech Projects (3 Credits)
Typically offered Fall, Spring, and Summer terms
This is a capstone course that connects students directly with real-world information technology problems. The goal of this course is to teach the skills needed for success in real-world information technology via a combination of classroom lectures and practical experience with large projects that have been specified by local "clients." The typical clients are primarily companies, but can also be government agencies or nonprofit organizations. Each project lasts for the entire semester and is designed to involve the full software project life cycle. Examples of such projects are development of software to solve a business problem, including specifying requirements, writing and testing prototype code, and writing a final report; and evaluation of commercial software to be purchased to address a business problem, including gathering requirements, designing an architecture to connect the new software with existing systems, and assessing the suitability of available software products.
Grading: GSAS Graded
Repeatable for additional credit: Yes

CSCI-GA 3813 Advanced Lab: (1-12 Credits)
Typically offered Fall, Spring, and Summer terms
Large-scale programming project or research in cooperation with a faculty member or a professional internship.
Grading: GSAS Graded
Repeatable for additional credit: Yes

CSCI-GA 3840 Master's Thesis Research (1-6 Credits)
Typically offered Fall, Spring, and Summer terms
Requires approval of a faculty adviser and the Director of Graduate Studies for the M.S. programs. The thesis must satisfy the following criteria: It must be original research or design/implementation. The work must be equivalent to that of two regular MS courses. The thesis must result in a high-quality document of 30-50 pages or more.
Grading: GSAS Graded
Repeatable for additional credit: Yes

CSCI-GA 3850 PhD Research Seminar (1 Credit)
Typically offered occasionally
Graduate seminars serve as loosely structured forums for exploring research topics from broad areas of computer science. They are designed to foster dialogue by bringing together faculty and students from a given area and to encourage the exchange of ideas. As such, they bridge the gap between more structured course offerings and informal research meetings.
Grading: GSAS Pass/Fail
Repeatable for additional credit: Yes

CSCI-GA 3860 PhD Thesis Research (1-12 Credits)
Typically offered Fall, Spring, and Summer terms
This course is designed to assist students in the preparation of their thesis or dissertation research proposal
Grading: GSAS Graded
Repeatable for additional credit: Yes

CSCI-GA 3870 Internship in Computer Science (1-3 Credits)
Typically offered Fall, Spring, and Summer terms
Participation in a programming project or research project conducted
outside the university in a governmental, commercial, or academic
setting. Open only to graduate students with permission of the Director of
Graduate Studies (DGS). Students must submit a brief written description
of their work to the DGS before starting the internship and submit a
written summary of their work when it is completed.
Grading: GSAS Graded
Repeatable for additional credit: Yes