Computer Science (CS)

CS 1100. Computer Science and Its Applications. (4 Hours)
Introduces students to the field of computer science and the patterns of thinking that enable them to become intelligent users of software tools in a problem-solving setting. Examines several important software applications so that students may develop the skills necessary to use computers effectively in their own disciplines.

Corequisite(s): CS 1101
Attribute(s): NUpath Analyzing/Using Data

CS 1101. Lab for CS 1100. (1 Hour)
Accompanies CS 1100. Involves experiments and problem solving across multiple disciplines using computer science techniques and tools.

Corequisite(s): CS 1100

CS 1200. First Year Seminar. (1 Hour)
Seeks to support students in their transition to Northeastern and in their holistic development as they become responsible members of the college and university communities. Incorporates large group discussion, small group activities, and self-reflection in order to facilitate connections with faculty, staff, and peers; promote utilization of appropriate campus resources; and assist with academic and personal goal setting.

CS 1210. Professional Development for Khoury Co-op. (1 Hour)
Continues the preparation of students for careers in the computing and information fields by discussing co-op and co-op processes. Offers students an opportunity to prepare a professional resumé; practice proper interviewing techniques; explore current job opportunities; learn how to engage in the job and referral process; and to understand co-op policies, procedures, and expectations. Discusses professional behavior and ethical issues in the workplace.

Prerequisite(s): CS 2100 with a minimum grade of D- or CS 2510 with a minimum grade of D- or DS 2500 with a minimum grade of D-

CS 1800. Discrete Structures. (4 Hours)
Introduces the mathematical structures and methods that form the foundation of computer science. Studies structures such as sets, tuples, sequences, lists, trees, and graphs. Discusses functions, relations, ordering, and equivalence relations. Examines inductive and recursive definitions of structures and functions. Discusses principles of proof such as truth tables, inductive proof, and basic logic. Also covers the counting techniques and arguments needed to estimate the size of sets, the growth of functions, and the space-time complexity of algorithms.

Corequisite(s): CS 1802
Attribute(s): NUpath Formal/Quant Reasoning

CS 1802. Seminar for CS 1800. (1 Hour)
Accompanies CS 1800. Illustrates topics from the lecture course through discussions, quizzes, and homework assignments.

Corequisite(s): CS 1800

CS 1990. Elective. (1-4 Hours)
Offers elective credit for courses taken at other academic institutions. May be repeated without limit.

CS 2000. Introduction to Program Design and Implementation. (4 Hours)
Introduces computer science and data science to students with no prior programming experience. Starts by building programs with numbers, text, and images. Explores real, complex datasets both interactively and through coding by doing data analysis, constructing graphs, and computing interesting results. Offers students an opportunity to build a solid foundation and practice coding with a popular industrial coding language with a professional programmer’s interface to the code. Covers skills needed to identify and respond to ethical challenges that arise in the program design process.

Corequisite(s): CS 2001
Attribute(s): NUpath Formal/Quant Reasoning, NUpath Natural/Designed World

CS 2001. Lab for CS 2000. (1 Hour)
Accompanies CS 2000. Covers topics from the course through various experiments.

Corequisite(s): CS 2000

CS 2100. Program Design and Implementation 1. (4 Hours)
Builds on prior introductory programming experience (from inside or outside Northeastern University) to examine the fundamentals of program design and implementation. Studies design of data- and object-oriented programs including common patterns; use of data structures; and underlying principles such as abstraction, encapsulation, inheritance, and interfaces. Introduces common software engineering practices such as test-driven development, version control, development environments, and good programming habits. Students practice using these design principles by writing medium-sized applications and using data science libraries. Covers skills needed to identify and respond to ethical challenges that arise in the program design process.

Prerequisite(s): CS 2000 with a minimum grade of D- or CS 2500 with a minimum grade of D-
Corequisite(s): CS 2101
Attribute(s): NUpath Analyzing/Using Data, NUpath Natural/Designed World

CS 2101. Lab for CS 2100. (1 Hour)
Accompanies CS 2100. Covers topics from the course through various experiments.

Corequisite(s): CS 2100

CS 2386. Game Programming 1. (4 Hours)
Introduces the different subsystems used to create a 3D game: rendering, animation, collision, physics, audio, trigger systems, game logic, and simple game artificial intelligence. Offers students an opportunity to learn the inner workings of game engines and how to use multiple libraries such as physics and graphics libraries to develop a game, while developing proficiency in a high-level game programming language. Discusses graphics pipeline, scene management, level design, and behavior scripting.

Prerequisite(s): ARTG 2260 with a minimum grade of D- or ARTG 2262 with a minimum grade of D- or CS 2000 with a minimum grade of D- or CS 2100 with a minimum grade of D- or CS 2500 with a minimum grade of D-

CS 2484. Principles of Human-Computer Interaction. (4 Hours)
Introduces the foundational principles of human-computer interaction. Presents an overview of the field, focusing on methods for designing, prototyping, and evaluating user interfaces for computing applications. Emphasizes a human-centered design approach, exploring topics such as human capabilities and social factors that influence interface use. Covers design methods, best practices for ensuring usability and accessibility, and usability evaluation techniques.

CS 2501. Lab for CS 2500. (1 Hour)
Accompanies CS 2500. Covers topics from the course through various experiments.

Corequisite(s): CS 2500

CS 2800. Logic and Computation. (4 Hours)
Introduces formal logic and its connections to computer and information science. Offers an opportunity to learn to translate statements about the behavior of computer programs into logical claims and to gain the ability to prove such assertions both by hand and using automated tools. Considers approaches to proving termination, correctness, and safety for programs. Discusses notations used in logic, propositional and first order logic, logical inference, mathematical induction, and structural induction. Introduces the use of logic for modeling the range of artifacts and phenomena that arise in computer and information science.

Prerequisite(s): (CS 1800 with a minimum grade of D- or MATH 1365 with a minimum grade of D- or MATH 2310 with a minimum grade of D-);
(CS 2100 with a minimum grade of D- or CS 2500 with a minimum grade of D-)

CS 2963. Topics. (1,2 Hours)
Offers undergraduate students an opportunity to learn about timely issues, develop new skills, or explore areas of broad interest in an immersive, short-course format. Content and instructors vary by offering. May be repeated three times.

CS 2990. Elective. (1-4 Hours)
Offers elective credit for courses taken at other academic institutions. May be repeated without limit.

CS 2991. Research in Computer Science. (1-4 Hours)
Offers an opportunity to conduct introductory-level research or creative endeavors under faculty supervision. May be repeated three times.

CS 2992. Research. (0 Hours)
Offers an opportunity to document student contributions to research projects or creative endeavors.

CS 3000. Algorithms and Data. (4 Hours)
Introduces the basic principles and techniques for the design, analysis, and implementation of efficient algorithms and data representations. Discusses asymptotic analysis and formal methods for establishing the correctness of algorithms. Considers divide-and-conquer algorithms, graph traversal algorithms, and optimization techniques. Introduces information theory and covers the fundamental structures for representing data. Examines flat and hierarchical representations, dynamic data representations, and data compression. Concludes with a discussion of the relationship of the topics in this course to complexity theory and the notion of the hardness of problems.

Prerequisite(s): ((CS 2100 with a minimum grade of D- or CS 2510 with a minimum grade of D- or DS 2500 with a minimum grade of D-); (CS 1800 with a minimum grade of D- or EECE 2160 with a minimum grade of D-))
Corequisite(s): CS 3001
Attribute(s): NUpath Formal/Quant Reasoning

CS 3001. Recitation for CS 3000. (0 Hours)
Accompanies CS 3000. Provides students with additional opportunities to ask questions and engage with course material.

Corequisite(s): CS 3000

CS 3100. Program Design and Implementation 2. (4 Hours)
Builds on foundations introduced in CS 2100 to examine program design at increasing scales of complexity. Reviews abstraction, encapsulation, inheritance, and interfaces in statically typed object-oriented languages. Presents a comparative approach to software design patterns and paradigms, including object-oriented and functional programming. Offers students an opportunity to obtain a deeper understanding of the principles of program design including interface design, test-driven development, graphical design notations, reusable software components, and open-source ecosystems. Illustrates the impact of design-time decisions on software correctness including accessibility, changeability, performance, reusability, and privacy. Students collaborate throughout the semester to design and implement a large software project.

Prerequisite(s): CS 2100 with a minimum grade of D- or CS 2510 with a minimum grade of D- or EECE 2560 with a minimum grade of D-
Corequisite(s): CS 3101
Attribute(s): NUpath Analyzing/Using Data, NUpath Natural/Designed World

CS 3101. Lab for CS 3100. (1 Hour)
Accompanies CS 3100. Covers topics from the course and provides students with additional opportunities to engage with course material.

Corequisite(s): CS 3100

CS 3200. Introduction to Databases. (4 Hours)
Studies the design of a database for use in a relational database management system. The entity-relationship model and normalization are used in problems. Relational algebra and then the SQL (structured query language) are presented. Advanced topics include triggers, stored procedures, indexing, elementary query optimization, and fundamentals of concurrency and recovery. Students implement a database schema and short application programs on one or more commercial relational database management systems.

Prerequisite(s): CS 2000 with a minimum grade of D- or CS 2100 with a minimum grade of D- or CS 2500 with a minimum grade of D- or DS 2000 with a minimum grade of D- or DS 2500 with a minimum grade of D- or EECE 2560 with a minimum grade of D-
Attribute(s): NUpath Analyzing/Using Data

CS 3386. Game Programming 2. (4 Hours)
Covers advanced concepts and techniques for developing high-performance, professional-quality 3D games using a modern game engine such as Unity3D. Building on foundational game programming skills, provides an opportunity to dive deeper into topics such as advanced physics systems, dynamic asset management, game data storage solutions, networking architectures and APIs for multiplayer games, advanced rendering techniques, and performance optimization. Offers opportunity to implement advanced game mechanics, persistent data storage, integrate telemetry for player analytics, and develop custom tools to streamline game development workflows. Presents skills necessary to tackle complex game development challenges and deliver polished, optimized games for various platforms.

Prerequisite(s): (CS 2100 with a minimum grade of D- or CS 2510 with a minimum grade of D-); (CS 2386 with a minimum grade of D- or CS 3540 with a minimum grade of D-)

CS 3484. GUI Programming. (4 Hours)
Focuses on the implementation of usable graphical user interfaces using object-oriented design principles. Demonstrates how to build interactive and responsive applications while focusing on the usability of GUIs. Covers fundamental concepts of GUI programming including event-driven programming, layout management, and state handling. Demonstrates how to develop proficiency in a modern GUI framework.

Prerequisite(s): CS 2100 with a minimum grade of D- or CS 2510 with a minimum grade of D-
Attribute(s): NUpath Writing Intensive

CS 3520. Programming in C++. (4 Hours)
Examines how to program in C++ in a robust and safe manner. Reviews basics, including scoping, typing, and primitive data structures. Discusses data types (primitive, array, structure, class, string); addressing/parameter mechanisms (value, pointer, reference); stacks; queues; linked lists; binary trees; hash tables; and the design of classes and class inheritance, emphasizing single inheritance. Considers the instantiation of objects, the trade-offs of stack vs. heap allocation, and the design of constructors and destructors. Emphasizes the need for a strategy for dynamic memory management. Addresses function and operator overloading; templates, the Standard Template Library (STL), and the STL components (containers, generic algorithms, iterators, adaptors, allocators, function objects); streams; exception handling; and system calls for processes and threads.

Prerequisite(s): CS 2100 with a minimum grade of D- or CS 2510 with a minimum grade of D- or DS 2500 with a minimum grade of D-

CS 3620. Building Extensible Systems. (4 Hours)
Deals with the design of extensible software systems, which enable clients to add functionality both statically as well as dynamically. Examples of such systems are operating systems, game servers, and Web browsers. Describes the classic systems built on C-like languages with unsafe, manual memory control and the more recent systems built on Java-like languages with safe, automated memory management. Introduces the Rust programming language, which combines the efficiency of C with safe manual memory control via type specifications and compiler constraints. Offers students an opportunity to build systems using all three settings but focuses on the Rust approach. Students also have an opportunity to evaluate their work via essays and memos.

Prerequisite(s): (CS 2100 with a minimum grade of D- or CS 2510 with a minimum grade of D-); (ENGW 1111 with a minimum grade of C or ENGW 1102 with a minimum grade of C or ENGL 1111 with a minimum grade of C or ENGL 1102 with a minimum grade of C)
Attribute(s): NUpath Writing Intensive

CS 3650. Computer Systems. (4 Hours)
Introduces the basic design of computing systems, computer operating systems, and assembly language using a RISC architecture. Describes caches and virtual memory. Covers the interface between assembly language and high-level languages, including call frames and pointers. Covers the use of system calls and systems programming to show the interaction with the operating system. Covers the basic structures of an operating system, including application interfaces, processes, threads, synchronization, interprocess communication, deadlock, memory management, file systems, and input/output control.

Prerequisite(s): CS 2510 with a minimum grade of D- or CS 3100 with a minimum grade of D- or EECE 2560 with a minimum grade of D-

CS 3700. Networks and Distributed Systems. (4 Hours)
Introduces the fundamentals of computer networks, including network architectures, network topologies, network protocols, layering concepts (for example, ISO/OSI, TCP/IP reference models), communication paradigms (point-to-point vs. multicast/broadcast, connectionless vs. connection oriented), and networking APIs (sockets). Also covers the construction of distributed programs, with an emphasis on high-level protocols and distributed state sharing. Topics include design patterns, transactions, performance trade-offs, security implications, and reliability. Uses examples from real networks (TCP/IP, Ethernet, 802.11) and distributed systems (Web, BitTorrent, DNS) to reinforce concepts.

Prerequisite(s): CS 2510 with a minimum grade of D-

CS 3800. Theory of Computation. (4 Hours)
Introduces the theory behind computers and computing aimed at answering the question, “What are the capabilities and limitations of computers?” Covers automata theory, computability, and complexity. The automata theory portion includes finite automata, regular expressions, nondeterminism, nonregular languages, context-free languages, pushdown automata, and noncontext-free languages. The computability portion includes Turing machines, the Church-Turing thesis, decidable languages, and the Halting theorem. The complexity portion includes big-O and small-o notation, the classes P and NP, the P vs. NP question, and NP-completeness.

Prerequisite(s): CS 2100 with a minimum grade of D- or CS 2510 with a minimum grade of D- or EECE 2160 with a minimum grade of D-

CS 3950. Introduction to Computer Science Research. (2 Hours)
Introduces students to research in the fields of computer science, information science, data science, and cybersecurity. Explores how the scientific method is applied to these fields and covers the breadth of subareas of specialty that exist. Offers students an opportunity to practice how to locate and read scientific literature in different subareas. Also offers students an overview of graduate education in these fields.

CS 3990. Elective. (1-4 Hours)
Offers elective credit for courses taken at other academic institutions. May be repeated without limit.

CS 4050. Artificial Intelligence and Society. (4 Hours)
Examines the societal impact of artificial intelligence technologies and prominent strategies for aligning these impacts with social and ethical values. Offers multidisciplinary readings to provide conceptual lenses for understanding these technologies in their contexts of use.

CS 4097. Mixed Reality. (4 Hours)
Seeks to provide a strong foundation in the fundamentals of virtual and augmented reality, broadly defined as mixed reality (XR). These technologies have recently witnessed a resurgence of interest. Offers students an opportunity to obtain hands-on experience developing XR applications by diving into this burgeoning area of research and practice in computer science. Synthesizes theoretical and practice knowledge from various disciplines, including computer graphics, 3D interfaces, human-computer interaction, tracking systems, and perceptual psychology.

Prerequisite(s): CS 3540 with a minimum grade of D-

CS 4100. Artificial Intelligence. (4 Hours)
Introduces the fundamental problems, theories, and algorithms of the artificial intelligence field. Includes heuristic search; knowledge representation using predicate calculus; automated deduction and its applications; planning; and machine learning. Additional topics include game playing; uncertain reasoning and expert systems; natural language processing; logic for common-sense reasoning; ontologies; and multiagent systems.

Prerequisite(s): CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D- or DS 3500 with a minimum grade of D-
Attribute(s): NUpath Capstone Experience, NUpath Writing Intensive

CS 4120. Natural Language Processing. (4 Hours)
Introduces the computational modeling of human language; the ongoing effort to create computer programs that can communicate with people in natural language; and current applications of the natural language field, such as automated document classification, intelligent query processing, and information extraction. Topics include computational models of grammar and automatic parsing, statistical language models and the analysis of large text corpora, natural language semantics and programs that understand language, models of discourse structure, and language use by intelligent agents. Course work includes formal and mathematical analysis of language models and implementation of working programs that analyze and interpret natural language text. Knowledge of statistics is helpful.

Prerequisite(s): CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D- or DS 3500 with a minimum grade of D-

CS 4130. Engineering LLM-Integrated Systems. (4 Hours)
Studies the software engineering foundations for systems that integrate large language models (LLMs). Examines how LLM-integrated systems turn natural language instructions into actions. Offers opportunities to build systems with natural and fluid interfaces, integrate them with existing software, rigorously test their behavior, and understand their failure modes and limitations.

Prerequisite(s): CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D- or DS 3500 with a minimum grade of D-

CS 4150. Game Artificial Intelligence. (4 Hours)
Offers an overview of classical and modern approaches to artificial intelligence in digital games. Focuses on the creation of believable agents and environments with the goal of providing a fun and engaging experience to a player. Covers player modeling, procedural content generation, behavior trees, interactive narrative, decision-making systems, cognitive modeling, and path planning. Explores different approaches for behavior generation, including learning and rule-based systems. Requires students to complete several individual assignments in these areas to apply the concepts covered in class. Students choose a group final project to explore one aspect of artificial intelligence for games in further depth. Offers students an opportunity to learn team management and communication. Students who do not meet course prerequisites may seek permission of instructor.

Prerequisite(s): (CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D-); CS 3520 with a minimum grade of D-
Attribute(s): NUpath Capstone Experience, NUpath Writing Intensive

CS 4180. Reinforcement Learning. (4 Hours)
Introduces reinforcement learning and the Markov decision process (MDP) framework. Covers methods for planning and learning in MDPs such as dynamic programming, model-based methods, and model-free methods. Examines commonly used representations including deep-learning representations. Students are expected to have a working knowledge of probability, to complete programming assignments, and to complete a course project that applies some form of reinforcement learning to a problem of interest.

Prerequisite(s): CS 3000 with a minimum grade of D- ; (ECON 2350 with a minimum grade of D- or CS 2810 with a minimum grade of D- or DS 3000 with a minimum grade of D- or ENVR 2500 with a minimum grade of D- or MATH 3081 with a minimum grade of D- or PSYC 2320 with a minimum grade of D-); (CS 2810 with a minimum grade of D- or DS 3000 with a minimum grade of D- or MATH 2331 with a minimum grade of D-)

CS 4220. Information Retrieval. (4 Hours)
Introduces information retrieval (IR) systems and different approaches to IR. Topics covered include evaluation of IR systems; retrieval, language, and indexing models; file organization; compression; relevance feedback; clustering; distributed retrieval and metasearch; probabilistic approaches to IR; Web retrieval; filtering, collaborative filtering, and recommendation systems; cross-language IR; multimedia IR; and machine learning for IR.

Prerequisite(s): (CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D- or DS 3500 with a minimum grade of D-); (CS 2810 with a minimum grade of D- or DS 3000 with a minimum grade of D- or ECON 2350 with a minimum grade of D- or MATH 2280 with a minimum grade of D- or MATH 3081 with a minimum grade of D- or MGSC 2301 with a minimum grade of D- or PSYC 2320 with a minimum grade of D-)

CS 4300. Computer Graphics. (4 Hours)
Charts a path through every major aspect of computer graphics with varying degrees of emphasis. Discusses hardware issues: size and speed; lines, polygons, and regions; modeling, or objects and their relations; viewing, or what can be seen (visibility and perspective); rendering, or how it looks (properties of surfaces, light, and color); transformations, or moving, placing, distorting, and animating and interaction, or drawing, selecting, and transforming.

Prerequisite(s): (CS 2510 with a minimum grade of D- or CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D- or EECE 2560 with a minimum grade of D-); (CS 2810 with a minimum grade of D- or DS 3000 with a minimum grade of D- or MATH 1260 with a minimum grade of D- or MATH 2331 with a minimum grade of D- or MATH 2341 with a minimum grade of D-)
Attribute(s): NUpath Capstone Experience, NUpath Writing Intensive

CS 4350. Empirical Research Methods. (4 Hours)
Evaluates and conducts empirical research, focusing on students' use of empirical methods to study the effectiveness and organizational/social impact of information systems and technologies. Empirical research involves a number of broad steps including identifying problems; developing specific hypotheses; collecting data relevant to the hypotheses; analyzing the data; and considering alternative explanations for the empirical findings. Some of the most commonly used research techniques, such as surveys, experiments, and ethnographic methods, are discussed. Additional topics include the ethics of data collection and experimentation in behavioral science. Although the course focuses primarily on the relationship between formulating research questions and implementing the appropriate methods to answer them, students can expect to apply the statistical techniques learned in the course prerequisites.

Prerequisite(s): CS 2810 with a minimum grade of D- or DS 3000 with a minimum grade of D- or ECON 2350 with a minimum grade of D- or ENVR 2500 with a minimum grade of D- or MATH 2280 with a minimum grade of D- or MATH 3081 with a minimum grade of D- or MGSC 2301 with a minimum grade of D- or PHTH 2210 with a minimum grade of D- or PSYC 2320 with a minimum grade of D-

CS 4360. Non-Interactive Computer Graphics. (4 Hours)
Introduces computer graphics algorithms and concepts primarily focusing on offline rendering techniques. Consists of a lecture component and in-class laboratory to study common image synthesis algorithms and techniques to generate images used in games and 3D animated movies. Culminates with a final project in which students complete in groups or individually a renderer for generating high quality images. Students with an interest in a career as a graphics, rendering, or high performance computer engineer may consider taking this course.

Prerequisite(s): (CS 2810 with a minimum grade of D- or DS 3000 with a minimum grade of D- or MATH 2331 with a minimum grade of D-); (CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D-)

CS 4400. Programming Languages. (4 Hours)
Introduces a systematic approach to understanding the behavior of programming languages. Covers interpreters; static and dynamic scope; environments; binding and assignment; functions and recursion; parameter-passing and method dispatch; objects, classes, inheritance, and polymorphism; type rules and type checking; and concurrency.

Prerequisite(s): (CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D-); (CS 3000 with a minimum grade of D- or CS 4800 with a minimum grade of D-)

CS 4410. Compilers. (4 Hours)
Studies the construction of compilers and integrates material from earlier courses on programming languages, automata theory, computer architecture, and software design. Examines syntax trees; static semantics; type checking; typical machine architectures and their software structures; code generation; lexical analysis; and parsing techniques. Uses a hands-on approach with a substantial term project.

Prerequisite(s): CS 4400 with a minimum grade of D- or CS 5400 with a minimum grade of C- or CS 7400 with a minimum grade of C-
Attribute(s): NUpath Capstone Experience, NUpath Writing Intensive

CS 4500. Software Development. (4 Hours)
Considers software development as a systematic process involving specification, design, documentation, implementation, testing, and maintenance. Examines software process models; methods for software specification; modularity, abstraction, and software reuse; and issues of software quality. Students, possibly working in groups, design, document, implement, test, and modify software projects.

Prerequisite(s): (CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D-); (ENGW 1102 with a minimum grade of C or ENGW 1111 with a minimum grade of C)
Attribute(s): NUpath Writing Intensive

CS 4520. Mobile Application Development. (4 Hours)
Focuses on mobile application development on a mobile phone or related platform. Discusses memory management; user interface building, including both MVC principles and specific tools; touch events; data handling, including core data, SQL, XML, and JSON; network techniques and URL loading; and, finally, specifics such as GPS and motion sensing that may be dependent on the particular mobile platform. Students are expected to work on a project that produces a professional-quality mobile application. The instructor chooses a modern mobile platform to be used in the course.

Prerequisite(s): CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D-

CS 4530. Fundamentals of Software Engineering. (4 Hours)
Covers the fundamentals of software engineering, including software development life cycle models (e.g., waterfall, spiral,agile); requirements analysis; user-centered design; software design principles and patterns; testing (functional testing, structural testing, testing strategies); code refactoring and debugging; software architecture and design; and integration and deployment. Includes a course project in which some of the software engineering methods (from requirements analysis to testing) are applied in a team-based setting.

Prerequisite(s): CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D-
Attribute(s): NUpath Capstone Experience, NUpath Writing Intensive

CS 4535. Professional Practicum Capstone. (4 Hours)
Offers students an opportunity to expand skills in real-world application development and team collaboration within an experiential learning structure. Involves students in industry-specific projects designed to integrate into an industry partner's intellectual property portfolio. Projects focus on software design and development and also include written components that align with the partner's standards and reference relevant prior work. Students engage in a structured process of milestones and feedback cycles with peers; instructors; and, as appropriate, industry partners to refine project outcomes.

Attribute(s): NUpath Capstone Experience, NUpath Integration Experience, NUpath Writing Intensive

CS 4550. Web Development. (4 Hours)
Discusses Web development for sites that are dynamic, data driven, and interactive. Focuses on the software development issues of integrating multiple languages, assorted data technologies, and Web interaction. Considers ASP.NET, C#, HTTP, HTML, CSS, XML, XSLT, JavaScript, AJAX, RSS/Atom, SQL, and Web services. Requires each student to deploy individually designed Web experiments that illustrate the Web technologies and at least one major integrative Web site project. Students may work as a team with the permission of the instructor. Each student or team must also create extensive documentation of their goals, plans, design decisions, accomplishments, and user guidelines. All source files must be open and be automatically served by a sources server.

Prerequisite(s): CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D- or DS 3500 with a minimum grade of D-
Attribute(s): NUpath Capstone Experience, NUpath Writing Intensive

CS 4610. Robotic Science and Systems. (4 Hours)
Introduces autonomous mobile robots, with a focus on algorithms and software development, including closed-loop control, robot software architecture, wheeled locomotion and navigation, tactile and basic visual sensing, obstacle detection and avoidance, and grasping and manipulation of objects. Offers students an opportunity to progressively construct mobile robots from a predesigned electromechanical kit. The robots are controlled wirelessly by software of the students' own design, built within a provided robotics software framework. The course culminates in a grand challenge competition using all features of the robots.

Prerequisite(s): CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D-

CS 4700. Network Fundamentals. (4 Hours)
Introduces the fundamental concepts of network protocols and network architectures. Presents the different harmonizing functions needed for the communication and effective operation of computer networks. Provides in-depth coverage of data link control, medium access control, routing, end-to-end transport protocols, congestion and flow control, multicasting, naming, auto configuration, quality of service, and network management. Studies the abstract mechanisms and algorithms as implemented in real-world Internet protocols. Also covers the most common application protocols (e-mail, Web, and ftp).

Prerequisite(s): CS 3650 with a minimum grade of D- or CS 5600 with a minimum grade of D- or CS 5600 with a minimum grade of C- (Graduate)

CS 4730. Distributed Systems. (4 Hours)
Introduces distributed systems, covering fundamental concepts and showing how they are applied to build reliable distributed services. Examines several existing distributed applications, such as file systems, databases, lock services, digital currencies, smart contracts, and machine learning, and how these applications must coordinate to function and overcome failures, network partitions, or compromised parties. Distributed systems, such as databases, cloud services, and blockchains, are omnipresent in the services and applications that serve society on a daily basis.

Prerequisite(s): CS 3650 with a minimum grade of D-

CS 4805. Fundamentals of Complexity Theory. (4 Hours)
Reviews basic material such as automata, Turing machines, (un)decidability, time complexity, P vs. NP, and NP-completeness. Studies core topics in computational complexity, including time and space complexity, polynomial hierarchy, circuit complexity, probabilistic computation, interactive proofs, and hardness of approximation. Optional topics may include Gödel’s incompleteness theorem, Kolgomorov complexity, cryptography, quantum computing, communication complexity, lower bounds, or pseudorandomness.

Prerequisite(s): CS 3800 with a minimum grade of D-

CS 4810. Advanced Algorithms. (4 Hours)
Builds on CS 3000. Presents an advanced study of computer algorithms. Covers basic algorithmic paradigms (e.g., greedy, divide-and-conquer, and dynamic programming); graph algorithms; optimization; computational Intractability (e.g., NP-completeness, PSPACE-completeness); randomized algorithms; and approximation algorithms.

Prerequisite(s): CS 3000 with a minimum grade of D-

CS 4820. Computer-Aided Reasoning. (4 Hours)
Covers fundamental concepts, techniques, and algorithms in computer-aided reasoning, including propositional logic, variants of the DPLL algorithm for satisfiability checking, first-order logic, unification, tableaux, resolution, Horn clauses, congruence closure, rewriting, Knuth-Bendix completion, decision procedures, Satisfiability Modulo Theories, recursion, induction, termination, Presburger arithmetic, quantifier elimination, and interactive theorem proving. Offers students an opportunity to develop and implement a reasoning engine in a sequence of projects over the course of the semester. Also covers how to formalize and reason about computational systems using a modern interactive theorem prover.

Prerequisite(s): CS 2800 with a minimum grade of D- ; CS 3000 with a minimum grade of D-
Attribute(s): NUpath Capstone Experience

CS 4830. System Specification, Verification, and Synthesis. (4 Hours)
Covers the fundamental topics in formal modeling and specification (transition systems, temporal logic, regular and omega-regular languages, safety and liveness properties, etc.); computer-aided verification (state-space exploration, model checking, bounded-model checking, binary-decision diagrams, symbolic model checking, etc.); compositionality and assume-guarantee reasoning; contracts; and component-based design. Also covers fundamental topics in computer-aided synthesis of correct-by-construction systems, starting from high-level formal specifications or from example scenarios. Designing large and complex systems (digital circuits, embedded control systems such as automated vehicles, computerized healthcare devices such as pacemakers, cyber-physical systems such as automated intersections, etc.) and their software cannot be done by hand. Instead, designers use computer-aided techniques that allow them to build system models and verify correctness of the design before the real system is actually built.

Prerequisite(s): CS 3000 with a minimum grade of D-

CS 4850. Building Game Engines. (4 Hours)
Discusses the components of game engines and strategies for their software implementation. Includes graphics management algorithms (animation, scene graph, level of detail); basic artificial intelligence algorithms (search, decision making, sensing); and related algorithmic issues (networking, threading, input processing). Explores the use of data-driven software design. Offers students an opportunity to use a rendering engine and to build and integrate several software components to create a complete game engine. Requires students to work on several individual assignments to apply the algorithms and then develop a project in a team. Offers students an opportunity to learn team/project management; work division; team communication; and the software development cycle of implementation, testing, critique, and further iteration. Students who do not meet course prerequisites may seek permission of instructor.

Prerequisite(s): CS 3520 with a minimum grade of D- ; CS 3540 with a minimum grade of D-

CS 4950. Computer Science Research Seminar. (1 Hour)
Offers students an in-depth look at research in a particular subarea of computer science, information science, data science, or cybersecurity. The particular subarea varies from semester to semester. Exposes students to current research topics, often via guest faculty members. Offers students an opportunity to practice reading and discussing scientific literature, presenting scientific work, and distilling the key ideas and contributions of papers through required weekly paper summaries. May be repeated once.

Prerequisite(s): CS 3950 with a minimum grade of D-

CS 4955. Computer Science Teaching Seminar. (1 Hour)
Introduces techniques and frameworks to prepare undergraduate students to become more effective teaching assistants in the field of computer science. Students analyze and reflect on literature, case studies, and real examples of teaching computer science. Offers students an opportunity to participate within in-class activities to learn presentation skills, to practice speaking to different audience sizes, and to learn how to work with different types of audiences. Culminates with a final capstone project in which students prepare and present a lecture on a topic in computer science. Successful students are prepared for careers in teaching, presenting technical content when pursuing graduate studies, and for presenting technical information in industry.

Prerequisite(s): CS 2500 with a minimum grade of D-

CS 4970. Junior/Senior Honors Project 1. (4 Hours)
Focuses on in-depth project in which a student conducts research or produces a product related to the student's major field. Combined with Junior/Senior Project 2 or college-defined equivalent for 8 credit honors in the discipline project.

CS 4971. Junior/Senior Honors Project 2. (4 Hours)
Focuses on second semester of in-depth project in which a student conducts research or produces a product related to the student's major field.

Prerequisite(s): CS 4970 with a minimum grade of D-

CS 4973. Topics in Computer Science. (4 Hours)
Offers a lecture course in computer science on a topic not regularly taught in a formal course. Topics may vary from offering to offering. May be repeated up to three times.

Prerequisite(s): CS 3000 with a minimum grade of D- ; (CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D- or DS 3500 with a minimum grade of D-)

CS 4990. Elective. (1-4 Hours)
Offers elective credit for courses taken at other academic institutions. May be repeated without limit.

CS 4991. Research. (4,8 Hours)
Offers an opportunity to conduct research under faculty supervision. May be repeated up to three times.

Prerequisite(s): (CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D-); CS 3800 with a minimum grade of D-
Attribute(s): NUpath Capstone Experience, NUpath Integration Experience, NUpath Writing Intensive

CS 4992. Directed Study. (1-6 Hours)
Focuses on student examining standard computer science material in fresh ways or new computer science material that is not covered in formal courses. May be repeated up to three times.

Prerequisite(s): (CS 3100 with a minimum grade of D- or CS 3500 with a minimum grade of D-); CS 3800 with a minimum grade of D-

CS 4998. Research. (0 Hours)
Offers an opportunity to document student contributions to research projects or creative endeavors.

CS 5001. Intensive Foundations of Computer Science. (4 Hours)
Introduces systematic problem solving through programming. Offers students an opportunity to learn how to analyze a problem, how to divide and organize the problem into appropriate components, how to describe the problem in a computer language, how to analyze and understand the behavior of their programs, and how to test that their programs are working correctly. Additionally, introduces a method of program design called object-oriented programming and various ways to organize data, including a discussion of their advantages and disadvantages. To practice the course concepts, students undertake assignments ranging from small, highly specified programming tasks to larger open-ended problems where students design and code their own solutions.

Corequisite(s): CS 5003

CS 5002. Discrete Structures. (4 Hours)
Introduces the mathematical structures and methods that form the foundation of computer science. Studies structures such as sets, tuples, sequences, lists, trees, and graphs. Discusses functions, relations, ordering, and equivalence relations. Examines inductive and recursive definitions of structures and functions. Covers principles of proof such as truth tables, inductive proof, and basic logic and the counting techniques and arguments needed to estimate the size of sets, the growth of functions, and the space-time complexity of algorithms. Also, discusses data structures such as arrays, stacks, queues, lists, and the algorithms that manipulate them.

CS 5003. Recitation for CS 5001. (0 Hours)
Provides a small-group discussion format to cover material in CS 5001.

Corequisite(s): CS 5001

CS 5004. Object-Oriented Design. (4 Hours)
Presents a comparative approach to object-oriented programming and design. Discusses the concepts of object, class, metaclass, message, method, inheritance, and genericity. Reviews forms of polymorphism in object-oriented languages. Contrasts the use of inheritance and composition as dual techniques for software reuse—forwarding vs. delegation and subclassing vs. subtyping. Offers students an opportunity to obtain a deeper understanding of the principles of object-oriented programming and design, including software components, object-oriented design patterns, and the use of graphical design notations such as UML (unified modeling language). Illustrates basic concepts in object-oriented design with case studies in application frameworks and by writing programs in Java.

Prerequisite(s): (CS 5001 with a minimum grade of C- or CS 5001 with a minimum grade of C-); (CS 5002 with a minimum grade of C- or CS 5002 with a minimum grade of C-)
Corequisite(s): CS 5005

CS 5005. Recitation for CS 5004. (0 Hours)
Provides small-group discussion format to cover material in CS 5004.

Corequisite(s): CS 5004

CS 5008. Data Structures, Algorithms, and Their Applications within Computer Systems. (4 Hours)
Presents an integrated approach to the study of data structures, algorithms, and their application within systems topics. Introduces a variety of fundamental algorithmic techniques (divide-and-conquer, dynamic programming, graph algorithms) and systems topics (models of computation, computer architecture, compilation, system software, networking). Demonstrates the integration of topics through programming assignments in the C language that implement fundamental data structures (lists, queues, trees, maps, graphs) and algorithms as they are applied in computer systems. Additional breadth topics include programming applications that expose students to primitives of different subsystems using threads and sockets.

Prerequisite(s): (CS 5001 with a minimum grade of C- or CS 5001 with a minimum grade of C-); (CS 5002 with a minimum grade of C- or CS 5002 with a minimum grade of C-)
Corequisite(s): CS 5009

CS 5009. Recitation for CS 5008. (0 Hours)
Provides small-group active learning format to augment material in CS 5008.

Corequisite(s): CS 5008

CS 5010. Programming Design Paradigm. (4 Hours)
Introduces modern program design paradigms. Starts with functional program design, introducing the notion of a design recipe. The latter consists of two parts: a task organization (ranging from the description of data to the creation of a test suite) and a data-oriented approach to the organization of programs (ranging from atomic data to self-referential data definitions and functions as data). The course then progresses to object-oriented design, explaining how it generalizes and contrasts with functional design. In addition to studying program design, students also have an opportunity to practice pair-programming and public code review techniques, as found in industry today.

Corequisite(s): CS 5011

CS 5011. Recitation for CS 5010. (0 Hours)
Provides small-group discussion format to cover material in CS 5010.

Corequisite(s): CS 5010

CS 5047. Exploring AI Trends and Tools. (4 Hours)
Explores key ideas in artificial intelligence while delving into trending developments in the field. Examines AI tools and frameworks to enable collaboration effectively and efficiently across technical and nontechnical stakeholders. Analyzes topics such as AI-enabled perception, representation, reasoning, and learning as contextualized by societal impact. Offers students an opportunity to obtain the knowledge and skills necessary to confidently navigate the ever-evolving landscape of AI.

CS 5097. Mixed Reality. (4 Hours)
Seeks to provide a strong foundation in the fundamentals of virtual and augmented reality, broadly defined as mixed reality (XR), and hands-on experience developing XR applications. Offers students an opportunity to dive into this burgeoning area of research and practice in computer science. Given the complex nature of XR environments, this course synthesizes theoretical and practice knowledge from various disciplines, including computer graphics, 3D interfaces, human-computer interaction, tracking systems, and perceptual psychology (to name a few). XR technologies have recently witnessed a resurgence of interest.

CS 5100. Foundations of Artificial Intelligence. (4 Hours)
Introduces the fundamental problems, theories, and algorithms of the artificial intelligence field. Topics include heuristic search and game trees, knowledge representation using predicate calculus, automated deduction and its applications, problem solving and planning, and introduction to machine learning. Required course work includes the creation of working programs that solve problems, reason logically, and/or improve their own performance using techniques presented in the course.

Attribute(s): NUpath Capstone Experience, NUpath Writing Intensive

CS 5130. Applied Programming and Data Processing for AI. (4 Hours)
Presents an in-depth exploration of programming paradigms, mathematical foundations, and computational techniques essential for scientific computing and data-driven applications. Integrates linear algebra, Python programming principles, algorithmic design, and data science methodologies to offer students an opportunity to develop the technical expertise necessary for handling complex numerical computations and large-scale data processing. Strongly emphasizes Pythonic programming, object-oriented and functional programming, data preprocessing, visualization, and numerical stability, with the aim of developing the skills needed to write clean, efficient, and maintainable code for scientific and engineering applications.

Corequisite(s): CS 5131

CS 5131. Recitation for CS 5130. (0 Hours)
Provides small-group discussion format to cover material in CS 5130.

Corequisite(s): CS 5130

CS 5150. Game Artificial Intelligence. (4 Hours)
Offers an overview of classical and modern approaches to artificial intelligence in digital games. Focuses on the creation of believable agents and environments with the goal of providing a fun and engaging experience to a player. Covers player modeling, procedural content generation, behavior trees, interactive narrative, decision-making systems, cognitive modeling, and path planning. Explores different approaches for behavior generation, including learning and rule-based systems. Requires students to complete several individual assignments in these areas to apply the concepts covered in class. Students choose a group final project, which requires a report, to explore one aspect of artificial intelligence for games in further depth. Offers students an opportunity to learn team management and communication. Requires knowledge of algorithms and experience with object-oriented design or functional programming.

Attribute(s): NUpath Capstone Experience, NUpath Writing Intensive

CS 5170. Artificial Intelligence for Human-Computer Interaction. (4 Hours)
Offers an overview of the wide range of AI techniques that exploit knowledge of the domain and humans to facilitate interaction between humans and systems, mediate human-human interaction, leverage humans to improve system performance, and promote beneficial outcomes at the social and individual level. Topics can include AI/human computation, plan and activity recognition, smart sensing/homes, active learning, preference elicitation, intelligent/adaptive user interfaces, and mixed human-agent simulations. Studies how to design and develop intelligent interaction technologies while also critically assessing their social and ethical impact.

Prerequisite(s): CS 5100 with a minimum grade of C- ; CS 5800 with a minimum grade of C- ; CS 6140 with a minimum grade of C-

CS 5180. Reinforcement Learning and Sequential Decision Making. (4 Hours)
Introduces reinforcement learning and the underlying computational frameworks and the Markov decision process framework. Covers a variety of reinforcement learning algorithms, including model-based, model-free, value function, policy gradient, actor-critic, and Monte Carlo methods. Examines commonly used representations including deep learning representations and approaches to partially observable problems. Students are expected to have a working knowledge of probability and linear algebra, to complete programming assignments, and to complete a course project that applies some form of reinforcement learning to a problem of interest.

CS 5200. Database Management Systems. (4 Hours)
Introduces relational database management systems as a class of software systems. Prepares students to be sophisticated users of database management systems. Covers design theory, query language, and performance/tuning issues. Topics include relational algebra, SQL, stored procedures, user-defined functions, cursors, embedded SQL programs, client-server interfaces, entity-relationship diagrams, normalization, B-trees, concurrency, transactions, database security, constraints, object-relational DBMSs, and specialized engines such as spatial, text, XML conversion, and time series. Includes exercises using a commercial relational or object-relational database management system.

Attribute(s): NUpath Analyzing/Using Data

CS 5310. Computer Graphics. (4 Hours)
Introduces the fundamentals of two-dimensional and three-dimensional computer graphics, with an emphasis on approaches for obtaining realistic images. Covers two-dimensional algorithms for drawing lines and curves, anti-aliasing, filling, and clipping. Studies rendering of three-dimensional scenes composed of spheres, polygons, quadric surfaces, and bi-cubic surfaces using ray-tracing and radiosity. Includes techniques for adding texture to surfaces using texture and bump maps, noise, and turbulence. Requires knowledge of linear algebra.

Prerequisite(s): MATH 2331 with a minimum grade of D- or graduate program admission
Attribute(s): NUpath Capstone Experience, NUpath Writing Intensive

CS 5330. Pattern Recognition and Computer Vision. (4 Hours)
Introduces fundamental techniques for low-level and high-level computer vision. Examines image formation, early processing, boundary detection, image segmentation, texture analysis, shape from shading, photometric stereo, motion analysis via optic flow, object modeling, shape description, and object recognition (classification). Discusses models of human vision (gestalt effects, texture perception, subjective contours, visual illusions, apparent motion, mental rotations, and cyclopean vision). Requires knowledge of linear algebra.

Prerequisite(s): MATH 2331 with a minimum grade of D- or graduate program admission

CS 5335. Robotic Science and Systems. (4 Hours)
Introduces autonomous mobile robots with a focus on algorithms and software development, including closed-loop control, robot software architecture, wheeled locomotion and navigation, tactile and basic visual sensing, obstacle detection and avoidance, and grasping and manipulation of objects. Offers students an opportunity to progressively construct mobile robots from a predesigned electromechanical kit. The robots are controlled wirelessly by software of the students' own design, built within a provided robotics software framework. Culminates in a project that connects the algorithms and hardware developed in the course with a selected topic in the current robotics research literature.

CS 5340. Computer/Human Interaction. (4 Hours)
Covers the principles of human-computer interaction and the design and evaluation of user interfaces. Topics include an overview of human information processing subsystems (perception, memory, attention, and problem solving); how the properties of these systems affect the design of user interfaces; the principles, guidelines, and specification languages for designing good user interfaces, with emphasis on tool kits and libraries of standard graphical user interface objects; and a variety of interface evaluation methodologies that can be used to measure the usability of software. Other topics may include World Wide Web design principles and tools, computer-supported cooperative work, multimodal and "next generation" interfaces, speech and natural language interfaces, and virtual reality interfaces. Course work includes both the creation and implementation of original user interface designs, and the evaluation of user interfaces created by others. Requires knowledge of C programming language/UNIX.

CS 5350. Computational Geometry. (4 Hours)
Surveys practical techniques for representing geometric objects in two and three dimensions, for computing their motions and interactions, and for human interfaces to manipulate them. These techniques are useful not only in graphics but also in robotics, computer vision, game design, geographic information systems, computer-aided design and manufacturing, spatial reasoning and planning, physical simulation, biomechanics, and the implementation of many types of human-computer interface. Requires undergraduate background in algorithms.

Prerequisite(s): CS 5800 with a minimum grade of C- or CS 7800 with a minimum grade of C-

CS 5360. Noninteractive Computer Graphics. (4 Hours)
Introduces computer graphics algorithms and concepts. Focuses on offline rendering techniques. Consists of a lecture component and in-class laboratory offering instruction about common image synthesis algorithms and techniques to generate images used in games and 3D-animated movies. Culminates in a final individual or group project to complete a renderer for generating high-quality images. This course is appropriate for students interested in a career as a graphics, rendering, or high-performance computer engineer.

CS 5400. Principles of Programming Language. (4 Hours)
Studies the basic components of programming languages, specification of syntax and semantics, and description and implementation of programming language features. Discusses examples from a variety of languages.

Prerequisite(s): CS 5010 with a minimum grade of D- or CS 5004 with a minimum grade of B- or CS 5010 with a minimum grade of C- (Graduate)

CS 5500. Foundations of Software Engineering. (4 Hours)
Covers the foundations of software engineering, including software development life cycle models (e.g., waterfall, spiral, agile); requirements analysis; user-centered design; software design principles and patterns; testing (functional testing, structural testing, testing strategies); code refactoring and debugging; software architecture and design; and integration and deployment. Includes a course project where some of the software engineering methods (from requirements analysis to testing) are applied in a team-based setting. Requires admission to MS program or completion of all transition courses.

Prerequisite(s): CS 5010 with a minimum grade of D or CS 5004 with a minimum grade of C or CS 5010 with a minimum grade of C
Attribute(s): NUpath Writing Intensive

CS 5520. Mobile Application Development. (4 Hours)
Focuses on mobile application development on a mobile phone or related platform. Discusses memory management; user interface building, including both MVC principles and specific tools; touch events; data handling, including core data, SQL, XML, and JSON; network techniques and URL loading; and, finally, specifics such as GPS and motion sensing that may be dependent on the particular mobile platform. Students are expected to work on a project that produces a professional-quality mobile application and to demonstrate the application that they have developed. The instructor chooses a modern mobile platform to be used in the course.

CS 5540. Game Programming. (4 Hours)
Covers the skills needed to develop easily scalable and modifiable scripts that can be used to implement various game mechanics common to most game genres. Programming is an integral part of the digital game design and development life cycle. Designed as a foundational game programming course covering numerous aspects of game programming. At least one semester of programming experience in a high-level language is required.

CS 5600. Computer Systems. (4 Hours)
Studies the structure, components, design, implementation, and internal operation of computer systems, focusing mainly on the operating system level. Reviews computer hardware and architecture including the arithmetic and logic unit, and the control unit. Covers current operating system components and construction techniques including the memory and memory controller, I/O device management, device drivers, memory management, file system structures, and the user interface. Introduces distributed operating systems. Discusses issues arising from concurrency and distribution, such as scheduling of concurrent processes, interprocess communication and synchronization, resource sharing and allocation, and deadlock management and resolution. Includes examples from real operating systems. Exposes students to the system concepts through programming exercises. Requires admission to MS program or completion of all transition courses.

CS 5610. Web Development. (4 Hours)
Discusses Web development for sites that are dynamic, data driven, and interactive. Focuses on the software development issues of integrating multiple languages, assorted data technologies, and Web interaction. Considers ASP.NET, C#, HTTP, HTML, CSS, XML, XSLT, JavaScript, AJAX, RSS/Atom, SQL, and Web services. Each student must deploy individually designed Web experiments that illustrate the Web technologies and at least one major integrative Web site project. Students may work in teams with the permission of the instructor. Each student or team must also create extensive documentation of their goals, plans, design decisions, accomplishments, and user guidelines. All source files must be open and be automatically served by a sources server.

Attribute(s): NUpath Capstone Experience, NUpath Writing Intensive

CS 5700. Fundamentals of Computer Networking. (4 Hours)
Studies network protocols, focusing on modeling and analysis, and architectures. Introduces modeling concepts, emphasizing queuing theory, including Little's theorem, M/M/1, M/M/m, M/D/1, and M/G/1 queuing systems. Discusses performance evaluation of computer networks including performance metrics, evaluation tools and methodology, simulation techniques, and limitations. Presents the different harmonizing functions needed for communication and efficient operation of computer networks and discusses examples of Ethernet, FDDI, and wireless networks. Covers link layer protocols including HDLC, PPP, and SLIP; packet framing; spanning tree and learning bridges, error detection techniques, and automatic repeat request algorithms; sliding window and reliable/ordered services; and queuing disciplines including FQ and WFQ. Introduces flow control schemes, such as window flow control and leaky bucket rate control schemes, and discusses congestion control and fairness. Requires knowledge of probability theory.

CS 5800. Algorithms. (4 Hours)
Presents the mathematical techniques used for the design and analysis of computer algorithms. Focuses on algorithmic design paradigms and techniques for analyzing the correctness, time, and space complexity of algorithms. Topics may include asymptotic notation, recurrences, loop invariants, Hoare triples, sorting and searching, advanced data structures, lower bounds, hashing, greedy algorithms, dynamic programming, graph algorithms, and NP-completeness.

Attribute(s): NUpath Formal/Quant Reasoning

CS 5850. Building Game Engines. (4 Hours)
Discusses the components of game engines and strategies for their software implementation. Includes graphics management algorithms (animation, scene graph, level of detail); basic artificial intelligence algorithms (search, decision making, sensing); and related algorithmic issues (networking, threading, input processing). Explores the use of data-driven software design. Offers students an opportunity to use a rendering engine and to build and integrate several software components to create a complete game engine. Requires students to work on individual assignments and then develop a project in a team, which requires a report. Offers students an opportunity to learn team/project management; work division; team communication; and the software development cycle of implementation, testing, critique, and further iteration. Requires knowledge of computer graphics, differential calculus, operating systems concepts, and algorithms.

CS 5933. Advanced Computer Science Topics for Teachers. (4 Hours)
Offers learners intending to be certified as K–12 computer science educators an advanced programming course. Covers GUI development and event-driven programming, an introduction to modeling and simulation, basics of computer networking and security, as well as topics related to the state of the art in computer science.

CS 5934. Introduction to Inclusive Computer Science Teaching. (4 Hours)
Introduces teaching computer science (CS) in grades K–12. Focuses on integrating CS concepts with other topic areas. Includes working with multilingual learners—examining student identities with respect to CS and exploring strategies for creating an inclusive CS classroom. Offers students an opportunity to design and practice implementing lesson plans for CS curricula at all grade levels, to explore current research in CS education, and to build a portfolio of teaching resources.

CS 5963. Topics. (1,2 Hours)
Offers students an opportunity to learn about timely issues, develop new skills, or explore areas of broad interest in an immersive, short-course format. Content and instructors vary by offering.

CS 5964. Projects for Professionals. (0 Hours)
Offers students an applied project setting in which to apply their curricular learning. Working with a sponsor, students refine an applied research topic, perform research, develop recommendations that are shared with a partner sponsor, and create a plan for implementing their recommendations. Seeks to benefit students with a curriculum that supports the development of key business communication skills, project and client management skills, and frameworks for business analysis. Offers students an opportunity to learn from sponsor feedback, review 'lessons learned,' and incorporate suggestions from this review to improve and further develop their career development and professional plan. May be repeated twice.

CS 5965. Engaging with Industry Partners for Rising Professionals. (0 Hours)
Offers students an enhanced applied project setting in which to apply their curricular learning. Working with a partner sponsor, students refine an applied research topic, perform research, develop recommendations that are shared with the partner sponsor, and create a plan for implementing their recommendations. Curriculum supports students as they develop key business communication skills, project and client management skills, and frameworks for business analysis. Offers students an opportunity to learn from sponsor feedback, review lessons learned, and incorporate suggestions to improve and further hone their career development and professional plan. Career development opportunities through skill-building workshops, panels, and interview preparation are available. Partner-student interactions, including a culminating project presentation, allow partners to assess student potential for co-op, internship, or other employment opportunities with the partner. May be repeated two times.

CS 5976. Directed Study. (2-4 Hours)
Focuses on student examining standard computer science material in fresh ways or new computer science material that is not covered in formal courses. May be repeated up to three times.

CS 6120. Natural Language Processing. (4 Hours)
Provides an introduction to the computational modeling of human language, the ongoing effort to create computer programs that can communicate with people in natural language, and current applications of the natural language field, such as automated document classification, intelligent query processing, and information extraction. Topics include computational models of grammar and automatic parsing, statistical language models and the analysis of large text corpuses, natural language semantics and programs that understand language, models of discourse structure, and language use by intelligent agents. Course work includes formal and mathematical analysis of language models, and implementation of working programs that analyze and interpret natural language text.

CS 6130. Affective Computing. (4 Hours)
Studies affective computing—computing that relates to, arises from, or influences emotions. Offers an overview of the theory of human emotion (how it arises from and influences cognition, the body, and the social environment) and computational techniques for modeling human emotion processes as well as for recognizing and synthesizing emotional behavior. Discusses how these can be applied to application design. Offers students an opportunity to gain a strong background in the theory and practice of human-centered computing as it relates to games, immersive environments, and pedagogical applications. Brings together students from different disciplines to work together and learn from each other. CS 6130 and PSYC 6130 are cross-listed.

CS 6140. Machine Learning. (4 Hours)
Provides a broad look at a variety of techniques used in machine learning and data mining, and also examines issues associated with their use. Topics include algorithms for supervised learning including decision tree induction, artificial neural networks, instance-based learning, probabilistic methods, and support vector machines; unsupervised learning; and reinforcement learning. Also covers computational learning theory and other methods for analyzing and measuring the performance of learning algorithms. Course work includes a programming term project.

Prerequisite(s): CS 5800 with a minimum grade of C- or CS 7800 with a minimum grade of C-

CS 6170. AI Capstone. (4 Hours)
Offers students a culminating experience through one or more projects to apply key concepts learned throughout their program in the core and elective courses. Emphasizes exploiting domain knowledge to facilitate interaction between humans and systems; developing intelligent interaction technologies; and critically assessing their social and ethical impacts.

Prerequisite(s): CS 5100 with a minimum grade of C- ; CS 5130 with a minimum grade of C- ; CS 5800 with a minimum grade of C- ; (CS 6140 with a minimum grade of C- or EECE 5644 with a minimum grade of C-)

CS 6180. Foundations for Generative AI. (4 Hours)
Offers students an opportunity to obtain both the theoretical foundation and practical skills to understand, analyze, and apply generative AI techniques effectively. Begins with foundational concepts in AI and machine learning; progresses to advanced natural language processing, transformer models, and large language models; and concludes with cutting-edge techniques like prompt engineering, domain-specific adaptations, and retrieval-augmented generation.

Prerequisite(s): CS 5100 with a minimum grade of C- or CS 5180 with a minimum grade of C- or CS 6120 with a minimum grade of C- or CS 6140 with a minimum grade of C- or CS 7140 with a minimum grade of C- or CS 7150 with a minimum grade of C-

CS 6200. Information Retrieval. (4 Hours)
Provides an introduction to information retrieval systems and different approaches to information retrieval. Topics covered include evaluation of information retrieval systems; retrieval, language, and indexing models; file organization; compression; relevance feedback; clustering; distributed retrieval and metasearch; probabilistic approaches to information retrieval; Web retrieval; filtering, collaborative filtering, and recommendation systems; cross-language IR; multimedia IR; and machine learning for information retrieval.

CS 6220. Data Mining Techniques. (4 Hours)
Covers various aspects of data mining, including classification, prediction, ensemble methods, association rules, sequence mining, and cluster analysis. The class project involves hands-on practice of mining useful knowledge from a large data set.

Prerequisite(s): CS 5800 with a minimum grade of C- or CS 7800 with a minimum grade of C-

CS 6240. Large-Scale Parallel Data Processing. (4 Hours)
Covers big-data analysis techniques that scale out with increasing number of compute nodes, e.g., for cloud computing. Emphasizes approaches for problem and data partitioning that distribute work effectively, while keeping total cost for computation and data transfer low. Studies and analyzes deterministic and random algorithms from a variety of domains, including graphs, data mining, linear algebra, and information retrieval in terms of their cost, scalability, and robustness against skew. Course work emphasizes hands-on programming experience with modern state-of-the-art big-data processing technology. Students who do not meet course prerequisites may seek permission of instructor.

Prerequisite(s): CS 5800 with a minimum grade of C- or CS 7800 with a minimum grade of C- or EECE 7205 with a minimum grade of C-

CS 6350. Empirical Research Methods. (4 Hours)
Presents an overview of methods for conducting empirical research within computer science. These methods help provide objective answers to questions about the usability, effectiveness, and acceptability of systems. The course covers the basics of the scientific method, building from a survey of objective measures to the fundamentals of hypothesis testing using relatively simple research designs, and on to more advanced research designs and statistical methods. The course also includes a significant amount of fieldwork, spanning the design, conduct, and presentation of small empirical studies.

CS 6410. Compilers. (4 Hours)
Expects each student to write a small compiler. Topics include parser generation, abstract syntax trees, symbol tables, type checking, generation of intermediate code, simple code improvement, register allocation, run-time structures, and code generation.

Prerequisite(s): CS 5400 with a minimum grade of C- or CS 7400 with a minimum grade of C-
Attribute(s): NUpath Capstone Experience, NUpath Writing Intensive

CS 6510. Advanced Software Development. (4 Hours)
Designed to integrate academic concepts and practical experience of software design by having students work as part of a programming team, with an option to lead a subteam. Offers students an opportunity to study, in-depth, some aspects of the development process. The goal is to have students participate in a large-scale project, taking time to reflect and analyze the work and the process, rather than concentrating exclusively on the final product. Students who do not meet course prerequisites may seek permission of instructor.

Prerequisite(s): (CS 5004 with a minimum grade of B- or CS 5010 with a minimum grade of C-); CS 5500 with a minimum grade of C-

CS 6620. Fundamentals of Cloud Computing. (4 Hours)
Covers fundamentals of cloud computing, including virtualization and containers, distributed file systems and object stores, infrastructure as a service platforms, open source cloud platforms, key big data platforms, and topics in data center scale systems. Combines classroom material delivered via lectures, readings from literature, student presentations, and a semester-long software project.

CS 6640. Operating Systems Implementation. (4 Hours)
Introduces concepts of OS kernels. Discusses the design and implementation of today's OS kernel components, such as process abstraction, virtual memory module, and file systems. Offers students an opportunity to obtain hands-on experience in building these components step-by-step through a series of assignments on a state-of-the-art hardware platform and to implement their own OS kernels that can run on bare-metal machines.

Prerequisite(s): CS 3650 with a minimum grade of D- or CS 5600 with a minimum grade of C-

CS 6650. Building Scalable Distributed Systems. (4 Hours)
Covers the essential elements of distributed, concurrent systems and builds upon that knowledge with engineering principles and practical experience with state-of-the-art technologies and methods for building scalable systems. Scalability is an essential quality of internet-facing systems and requires specialized skills and knowledge to build systems that scale at low cost.

CS 6760. Privacy, Security, and Usability. (4 Hours)
Challenges conventional wisdom and encourages students to discover ways that security, privacy, and usability can be made synergistic in system design. Usability and security are widely seen as two antagonistic design goals for complex computer systems. Topics include computer forensics, network forensics, user interface design, backups, logging, economic factors affecting adoption of security technology, trust management, and related public policy. Uses case studies such as PGP, S/MIME, and SSL. Introduces basic cryptography and hash function as it is needed. Course work includes analysis of papers, problem sets, and a substantial term project.

CS 6954. Co-op Work Experience - Half-Time. (0 Hours)
Provides eligible students with an opportunity for work experience. May be repeated without limit.

CS 6955. Co-op Work Experience Abroad - Half-Time. (0 Hours)
Provides eligible students with an opportunity for work experience abroad. May be repeated once.

CS 6962. Elective. (1-4 Hours)
Offers elective credit for courses taken at other academic institutions. May be repeated without limit.

CS 6964. Co-op Work Experience. (0 Hours)
Provides eligible students with an opportunity for work experience. May be repeated without limit.

CS 6965. Co-op Work Experience Abroad. (0 Hours)
Provides eligible students with an opportunity for work experience abroad. May be repeated without limit.

CS 6966. Practicum. (2-4 Hours)
Provides eligible students with an opportunity for practical experience.

CS 6983. Topics in Computer Science. (4 Hours)
Offers various topics in computer science. May be repeated once.

CS 7140. Advanced Machine Learning. (4 Hours)
Covers topics in advanced machine learning. Presents materials in the current machine learning literature. Focuses on graphical models, latent variable models, Bayesian inference, and nonparametric Bayesian methods. Seeks to prepare students to do research in machine learning. Expects students to read conference and journal articles, present these articles, and write an individual research paper. CS 7140 and EECE 7397 are cross-listed.

Prerequisite(s): CS 6140 with a minimum grade of C- or EECE 7204 with a minimum grade of C- or EECE 7313 with a minimum grade of C-

CS 7150. Deep Learning. (4 Hours)
Introduces deep learning, including the statistical learning framework, empirical risk minimization, loss function selection, fully connected layers, convolutional layers, pooling layers, batch normalization, multilayer perceptrons, convolutional neural networks, autoencoders, U-nets, residual networks, gradient descent, stochastic gradient descent, backpropagation, autograd, visualization of neural network features, robustness and adversarial examples, interpretability, continual learning, and applications in computer vision and natural language processing. Assumes students already have a basic knowledge of machine learning, optimization, linear algebra, and statistics.

Prerequisite(s): CS 6140 with a minimum grade of C- or DS 5220 with a minimum grade of C-

CS 7170. Seminar in Artificial Intelligence. (2-4 Hours)
Offers students an opportunity to read and present various survey and research papers in artificial intelligence. May be repeated up to five times for a maximum of 24 semester hours. Faculty supervisor and topics vary from semester to semester.

Prerequisite(s): CS 5100 with a minimum grade of C- or CS 5100 with a minimum grade of D-

CS 7180. Special Topics in Artificial Intelligence. (4 Hours)
Offers various topics on artificial intelligence. May be repeated up to two times.

CS 7200. Statistical Methods for Computer Science. (4 Hours)
Introduces concepts in applied statistics. Covers frequentist and Bayesian characterization of uncertainty for continuous and categorical data, principles of experimental design, and methods of causal inference. Discusses the methodological foundations, as well as issues of practical implementation and use.

CS 7240. Principles of Scalable Data Management: Theory, Algorithms, and Database Systems. (4 Hours)
Covers the algorithms, core principles, and foundational concepts for managing data at scale. Topics include data models, query languages, query execution and optimization, complexity of query execution and query resilience, data stream processing, parallel data processing, transactions, linear vs. relational algebra, factorizations, and uncertainty in logic. Requires standard CS knowledge of algorithms and hardness (e.g., a typical undergraduate class based on a standard algorithms textbook such as Ericson; Cormen, Leiserson, Rivest,and Stein; or Dasgupta, Papadimitriou, and Vazirani). Offers students an opportunity to gain hands-on experience through smaller assignments and a project. The project is flexible to allow students to explore scalable data management and analysis aspects related to their PhD research.

CS 7250. Information Visualization: Theory and Applications. (4 Hours)
Covers foundational as well as contemporary topics of interest in data visualization to enable the effective representation of data across disciplines, including examples drawn from computer science, physical sciences, biomedical sciences, humanities, and economics. Topics include data visualization theory and methodology, visualization design and evaluation, visual perception and cognition, interaction principles, and data encoding and representation techniques. Students who do not meet course restrictions may seek permission of instructor.

CS 7260. Visualization for Network Science. (4 Hours)
Covers the principles of information visualization in the specific context of network science. Introduces visual encoding of data and our understanding of human vision and perception; interaction principles including filtering, pivoting, aggregation; and both quantitative and human subjects evaluation techniques. Covers visualization techniques for several network types, including multivariate networks with attributes for entities and relationships, evolving and dynamic networks that change over time, heterogeneous networks with multiple types of entities, and geospatial networks. Offers students an opportunity to learn about the design of layout algorithms for node-link and matrix visualizations.

CS 7268. Verifiable Machine Learning. (4 Hours)
Introduces verifiable machine learning, focusing on the limitations of current learning-based methods regarding safety and robustness. Explores adversarial robustness, uncertainty quantification, safety verification, reachability analysis, and safe/robust reinforcement learning. Examines neural network relaxations and their applications in robotics, control, image classification, medical diagnosis, and language modeling.

CS 7270. Seminar in Database Systems. (2-4 Hours)
Gives students the opportunity to read and present various survey and research papers in database systems. Faculty supervisor and topics vary from semester to semester. May be repeated once.

Prerequisite(s): CS 5200 with a minimum grade of C- or CS 5200 with a minimum grade of D-

CS 7280. Special Topics in Database Management. (4 Hours)
Offers various topics. Possible areas include object-oriented database systems and distributed database systems. May be repeated up to two times.

CS 7290. Special Topics in Data Science. (4 Hours)
Offers special topics in data science, including machine learning, statistics, data mining, parallel and distributed data analysis, database systems, information retrieval, knowledge representation, information visualization, natural language processing, computational biology and bioinformatics, computational social science, digital humanities, health informatics, business, and predictive analytics. May be repeated once for up to 8 total credits.

CS 7295. Special Topics in Data Visualization. (4 Hours)
Offers various topics in data visualization. May be repeated once.

CS 7300. Empirical Research Methods for Human Computer Interaction. (4 Hours)
Introduces concepts related to human subject studies for human computer interaction and personal health informatics. Discusses the methodological foundations, as well as issues of practical implementation and use. Provides an overview of the most common experimental designs used in these fields, and how studies should be proposed, conducted, analyzed, and documented for publication. Covers qualitative methods, including interviewing and focus groups, along with approaches to inductive qualitative data analysis. Methods discussed in the course are useful in any area of science that involves the application of the scientific method to situations involving people interacting with computational artifacts or systems, including observation, hypothesis formation, measurement, and descriptive and inferential statistical analyses.

CS 7332. Machine Learning with Graphs. (4 Hours)
Covers a number of advanced topics in machine learning and data mining on graphs, including vertex classification, graph clustering, link prediction and analysis, graph distance functions, graph embedding and representation learning, deep learning for graphs, anomaly detection, graph summarization, network inference, adversarial learning on networks, and notions of fairness in social networks. Seeks to familiarize students with state-of-the-art descriptive and predictive algorithms on graphs. Requires a foundational understanding of calculus and linear algebra, probability, machine learning or data mining, algorithms, and programming skills.

Prerequisite(s): PHYS 5116 with a minimum grade of C

CS 7340. Theory and Methods in Human Computer Interaction. (4 Hours)
Covers the foundations of human abilities, computational artifacts, design, and evaluation. Human computer interaction concerns the design and evaluation of software based on a deep understanding of how humans interact with computers, devices, and sensors. The field merges theories from psychology and computer science, using methods from AI and design. Introduces cognitive, perceptual, and affective theories and theories of individual differences that allow us to design and develop better computer software and systems. Also covers research methods for designing and evaluating computer software systems. Topics discussed in the context of next-generation interaction modalities include sensors, haptics, wearables, and performative interfaces. Students who do not meet course restrictions may seek permission of instructor.

CS 7375. Seminar in Human-Computer Interaction. (4 Hours)
Offers students an opportunity to read and present various survey and research papers in human-computer interaction, computer-supported collaborative work, data visualization, social computing, design, critical computing, accessibility, game design, health informatics, personal health informatics, human-robot interaction, or HCI in security. May be repeated once.

CS 7380. Special Topics in Graphics/Image Processing. (4 Hours)
Offers various topics on graphics/image processing. May be repeated up to two times.

CS 7390. Special Topics in Human-Centered Computing. (4 Hours)
Offers various topics in human-centered computing. May be repeated once.

CS 7400. Intensive Principles of Programming Languages. (4 Hours)
Studies the basic components of programming languages, specification of syntax and semantics, and description and implementation of programming language features. Discusses examples from a variety of languages.

CS 7430. Formal Specification, Verification, and Synthesis. (4 Hours)
Covers software and system modeling (how to formally describe the behavior of software and systems); specification (how to formally state the properties that the system should have); verification (how to check whether—and ultimately prove—that a system satisfies its specification); and synthesis (how to automatically generate software and systems that are "correct-by-construction").

CS 7470. Seminar in Programming Languages. (2-4 Hours)
Gives students the opportunity to read and present various survey and research papers in programming languages. Faculty supervisor and topics vary from semester to semester. May be repeated three times for a maximum of 16 total semester hours.

Prerequisite(s): CS 5400 with a minimum grade of C- or CS 5400 with a minimum grade of D- or CS 7400 with a minimum grade of C-

CS 7480. Special Topics in Programming Language. (4 Hours)
Offers various topics in programming language. May be repeated up to two times.

CS 7485. Special Topics in Formal Methods. (4 Hours)
Offers various topics in formal methods. May be repeated without limit.

CS 7575. Seminar in Software Engineering. (2-4 Hours)
Gives students the opportunity to read and present various survey and research papers in software engineering. Faculty supervisor and topics vary from semester to semester. May be repeated for credit for PhD students.

Prerequisite(s): CS 6520 with a minimum grade of C-

CS 7580. Special Topics in Software Engineering. (4 Hours)
Offers various topics on software engineering. May be repeated up to two times.

CS 7600. Intensive Computer Systems. (4 Hours)
Studies the structure, components, design, implementation, and internal operation of computer systems, focusing on the operating system level. Reviews computer hardware and architecture including the arithmetic and logic unit, and the control unit. Covers current operating system components and construction techniques including the memory and memory controller, I/O device management, device drivers, memory management, file system structures, and the user interface. Discusses distributed operating systems, real-time systems, and addresses concurrent processes, scheduling, interprocess communication, and synchronization. Discusses relevant distributed algorithms. Also covers design and analysis techniques for desirable properties in computer systems including functional correctness (in the absence of faults), performance and throughput, fault-tolerance and reliability, real-time response, security, and quality of service. Draws examples from real operating systems. Emphasizes abstraction, while programming exercises are used to facilitate the understanding of concepts.

CS 7610. Foundations of Distributed Systems. (4 Hours)
Covers foundational concepts in the design and implementation of efficient and reliable distributed computing systems. Covers internet communication protocols, fault-tolerant computing, synchronization protocols, synchronous and asynchronous computing, dynamic group communication systems, load balancing, Byzantine models, distributed hash tables, distributed file systems, and application of foundational concepts to modern distributed systems in the field. Requires knowledge of operating systems; e.g., an undergraduate course in Systems and Networks, Computer Systems, or Networks and Distributed systems.

CS 7670. Seminar in Computer Systems. (2-4 Hours)
Gives students the opportunity to read and present various survey and research papers in computer systems. Faculty supervisor and topics vary from semester to semester. May be repeated three times for a maximum of 16 total semester hours.

Prerequisite(s): CS 5600 with a minimum grade of C- or CS 5600 with a minimum grade of D- or CS 7600 with a minimum grade of C-

CS 7675. Master's Research. (4 Hours)
Exposes students to research in the fields of computer sciences. Explores how the scientific method is applied to these fields and covers the breadth of subareas of specialty that exist. Offers students an opportunity to practice how to locate and read scientific literature in different subareas. May be repeated once.

CS 7680. Special Topics in Computer Systems. (4 Hours)
Offers various topics on computer systems. May be repeated up to two times.

CS 7770. Seminar in Computer Networks. (2-4 Hours)
Gives students the opportunity to read and present various survey and research papers in computer networks. Faculty supervisor and topics vary from semester to semester. May be repeated for credit for PhD students.

Prerequisite(s): CS 5700 with a minimum grade of C- or CS 5700 with a minimum grade of D-

CS 7775. Seminar in Computer Security. (2-4 Hours)
Gives students the opportunity to read and present various survey and research papers in cryptography and computer security. Faculty supervisor and topics vary from semester to semester. May be repeated for credit for PhD students.

CS 7800. Advanced Algorithms. (4 Hours)
Presents advanced mathematical techniques for designing and analyzing computer algorithms. Reviews some of the material covered in CS 5800 and then covers advanced topics. Emphasizes theoretical underpinnings of techniques used to solve problems arising in diverse domains. Topics include asymptotic analysis, advanced data structures, dynamic programming, greedy algorithms and matroid theory, amortized analysis, randomization, string matching, algebraic algorithms, and approximation algorithms. Introduces Turing machines, P and NP classes, polynomial-time reducibility, and NP completeness.

CS 7805. Complexity Theory. (4 Hours)
Covers core topics in computational complexity, including NP-completeness, time and space complexity, polynomial hierarchy, circuit complexity, probabilistic computation, interactive proofs, and hardness of approximation. Moves to more advanced topics that may include lower bounds, pseudorandomness, cryptography, and communication complexity.

Prerequisite(s): CS 7800 with a minimum grade of C-

CS 7810. Foundations of Cryptography. (4 Hours)
Offers students at the PhD level an accelerated introduction to cryptography and quickly progresses to advanced topics that are at the forefront of current research. Cryptography is the science of protecting information against adversarial eavesdropping and tampering. Examines what kind of security properties can be achieved by relying solely on probability and information theory, without restricting the adversary's computational power. Studies the complexity-theoretic basis of modern cryptography and the connection between computational hardness and pseudo-randomness. Explores, as the main component of the course, how to take a few well-studied problems in number theory and algebra and use them to build powerful cryptosystems with advanced functionality and security properties. Requires prior completion of an undergraduate course in the theory of computation.

CS 7840. Foundations and Applications of Information Theory. (4 Hours)
Studies information theory and selected applications for data management, machine learning, and information retrieval. Information theory examines the transmission, processing, extraction, and utilization of information. Topics include entropy; mutual information; cross-entropy; data processing theorem; information inequalities; Cox's theorem; maximum entropy solutions; and applications such as data normalization, decision trees, maximum likelihood, logistic regression, and VC dimensions. Requires prior study of standard computer science algorithms. Offers hands-on experience through a flexible project, allowing students to explore information theory in aspects related to their PhD research.

CS 7870. Seminar in Theoretical Computer Science. (2-4 Hours)
Gives students the opportunity to read and present various survey and research papers in theoretical computer science. Faculty supervisor and topics vary from semester to semester. May be repeated three times for a maximum of 16 total semester hours.

Prerequisite(s): CS 5800 with a minimum grade of C- or CS 5800 with a minimum grade of D- or CS 7800 with a minimum grade of C-

CS 7880. Special Topics in Theoretical Computer Science. (4 Hours)
Covers various topics including advanced cryptography, approximation algorithms, complexity theory, computational algebra, distributed computing, formal verification, network algorithms, online computation, parallel computing, and randomness and computation. May be repeated up to two times.

CS 7930. Effective Scientific Writing in Computer Science. (2 Hours)
Introduces the principles and practice of technical communication in computer science and related fields. Emphasizes concepts such as genre, structure, argument, and audience expectations. Focuses on the story of scientific research and how it manifests in written genre like the abstract, research statement, and conference paper, with a portion of the course focused on presentation design and delivery.

CS 7962. Elective. (2-4 Hours)
Offers elective credit for courses taken at other academic institutions. May be repeated without limit.

CS 7976. Directed Study. (2-4 Hours)
Focuses on student examining standard computer science material in fresh ways or new computer science material that is not covered in formal courses. May be repeated without limit.

CS 7980. Research Capstone. (4 Hours)
Offers students a culminating experience to demonstrate proficiency in key concepts learned throughout their programs in the core and elective courses. Designed to reinforce concepts in ethics and basic concepts in research, beyond an emphasis on the technical principles learned throughout the program.

Prerequisite(s): (CS 5004 with a minimum grade of C- or CS 5010 with a minimum grade of C-); CS 5800 with a minimum grade of C-

CS 7986. Research. (0 Hours)
Offers students an opportunity to conduct full-time research under faculty supervision.

CS 7990. Thesis. (4 Hours)
Offers selected work with the agreement of a project supervisor. May be repeated once.

CS 7996. Thesis Continuation - Half-Time. (0 Hours)
Offers continued thesis work conducted under the supervision of a departmental faculty.

Prerequisite(s): CS 7990 with a minimum grade of C-

CS 8674. Master’s Project. (4 Hours)
Offers selected work with the agreement of a project supervisor. May be repeated once.

CS 8948. Research Work Experience - Half-Time. (0 Hours)
Provides an opportunity for all doctoral students to engage in industry research in the area of their dissertation. Doctoral students register for this course before starting their off-campus internship. May be repeated up to five times.

CS 8949. Research Work Experience. (0 Hours)
Provides an opportunity for all doctoral students to engage in industry research in the area of their dissertation. Doctoral students register for this course before starting their off-campus internships. May be repeated without limit.

CS 8982. Readings. (1-8 Hours)
Offers selected readings under the supervision of a faculty member. May be repeated without limit.

CS 8986. Research. (0 Hours)
Offers an opportunity to conduct full-time research under faculty supervision. May be repeated without limit.

CS 9000. PhD Candidacy Achieved. (0 Hours)
Indicates successful completion of the doctoral comprehensive exam.

CS 9990. Dissertation Term 1. (0 Hours)
Offers selected work with the agreement of a thesis supervisor.

Prerequisite(s): CS 9000 with a minimum grade of S

CS 9991. Dissertation Term 2. (0 Hours)
Offers dissertation supervision by members of the department.

Prerequisite(s): CS 9990 with a minimum grade of S

CS 9996. Dissertation Continuation. (0 Hours)
Continues work with the agreement of a thesis supervisor.

Prerequisite(s): CS 9991 with a minimum grade of S or Dissertation Check with a score of REQ