CS - Computer Science

For a computer science course to be used as a prerequisite, it must have been passed with a C- or better.

Courses numbered 100 to 299 = lower-division; 300 to 499 = upper-division; 500 to 799 = undergraduate/graduate.

CS 194. Introduction to Digital Design (4).
3 Classroom hours; 2 Lab hours. An introduction to digital design concepts. Includes number systems, Boolean algebra, Karnaugh maps, combinational circuit design, adders, multiplexers, decoders, sequential circuit design, state diagram, flip flops, sequence detectors and test different combinational and sequential circuits. Uses CAD tools for circuit simulation. Prerequisite: MATH 111 or equivalent.

CS 210. Introduction to Computer Science (3).
General education introductory course. Broad introduction to the discipline of computer science. Covers algorithms, computer hardware, operating systems, introduction to high-level language programming, databases, artificial intelligence and other applications, and social issues. Prerequisite: MATH 111.

CS 211. Introduction to Programming (4).
3 Classroom hours; 2 Lab hours. First course in computer programming in a high-level language. Emphasizes analyzing problems, designing solutions and expressing them in the form of a well-structured program using the procedural aspects of C++. Prerequisite: MATH 111.

CS 238. Assembly Language Programming (3).
An introduction to basic concepts of computer organization and operation. Studies machine and assembly language programming concepts that illustrate basic principles and techniques. Laboratory exercises given for experience using personal computers. Prerequisite: CS 211.

CS 281I. Noncredit Internship (0).
Complements and enhances the student’s academic program by providing an opportunity to apply and acquire knowledge in a workplace environment as an intern. Prerequisite: departmental consent.

3 Classroom hours. 2 Lab hours. Concepts of object-oriented programming. Covers data abstractions, classes and objects, methods, inheritance, polymorphism, dynamically-bound method calls and data encapsulation. Includes programming assignments in C++. Prerequisite: CS 211.

CS 321. Discrete Structures I (3).
Cross-listed as MATH 321. Provides a mathematical foundation essential to the entire computer science curriculum. Includes propositional and predicate logic, induction, recursion and counting techniques. Prerequisite: MATH 242 with a grade of 2.00 or better.

CS 322. Discrete Structures II (3).
A continuation of CS 321. Includes relations, graphs, trees, Boolean algebra and automata. Prerequisite: CS 321.

CS 350. Computer Science Workshop (1-5).
Short-term courses with special computer science emphases. Repeatable for credit. No credit toward the major or minor in computer science. Prerequisite: departmental consent.

CS 394. Introduction to Computer Architecture (3).
Introduces multilevel approach to computer systems, with an emphasis on micro architecture and instruction set architecture levels. Also introduces techniques to improve performance such as cache memory and instruction level parallelism. Prerequisites: CS 194, 211.
CS 481I. Noncredit Internship (0).
Complements and enhances the student's academic program by providing an opportunity to apply and acquire knowledge in a workplace environment as an intern. Prerequisite: departmental consent.

CS 481N. Internship (1-3).
Complements and enhances the student's academic program by providing an opportunity to apply and acquire knowledge in a workplace environment as an intern. Graded Cr/NCr. Prerequisite: departmental consent.

CS 497. Special Topics (1-3).
1-3 Classroom hours; 0-2 Lab hours. Special topics of current interest in computer science. Prerequisite: departmental consent.

CS 498. Individual Projects (1-3).
Repeatable for a total of 6 hours of credit. Graded Cr/NCr. Prerequisite: departmental consent.

CS 510. Programming Language Concepts (3).
Theoretical concepts in the design and use of programming languages. Formal syntax, including Backus Normal Form (BNF), Extended Backus-Naur Form (EBNF), and syntax diagrams. Semantics, including declaration, allocation and evaluation, symbol table and runtime environment; data types and type checking, procedure activation and parameter passing, modules and abstract data types. Prerequisites: CS 311, MATH 322.

Fundamental principles of modern operating systems. CPU management including processes, threads, scheduling, synchronization, resource allocation and deadlock. Memory management including paging and virtual memory. Storage management and file systems. Prerequisites: CS 238, 311.

CS 560. Design and Analysis of Algorithms (3).
Design of various algorithms including several sorting algorithms. Analysis of their space and time complexities. Data structures include heaps, hash tables and binary search trees. Prerequisites: CS 322, 400; STAT 460 or IME 254.

CS 594. Microprocessor-Based System Design (4).
3 Classroom hours; 2 Lab hours. Presents knowledge and skills required to design and program microprocessor-based systems. Introduces vendor-supplied special-purpose chips such as interrupt controllers and programmable input/output devices. Laboratory activities give hands-on experience. Prerequisites: CS 238, 394.

CS 644. Advanced Unix Programming (3).
Improves skills in C programming under the Unix environment. Covers file I/O, both buffered and unbuffered, working with the Unix file system, concurrent programming with multiple processes and process control. Also includes the use of signals and concepts of interprocess communication with pipes and FIFOs. Laboratory activities give hands-on experience. Prerequisite: CS 540.

CS 655. Information Delivery on the Internet (3).
Explores the capabilities of providing information on the World Wide Web. Information is typically provided through some sort of website that incorporates static text and the dynamic capabilities of the Web. Learn how to create an interactive website through the use of CGI and Java programming and how to interconnect a website to databases and generate images on the fly. Java portion covers a wide range of Java language and the Applet interface and utilities. Prerequisite: CS 300.

CS 665. Introduction to Database Systems (3).
Fundamental aspects of relational database systems, conceptual database design and entity-relationship modeling; the relational data model and its foundations, relational languages and SQL, functional dependencies and logical database design; views, constraints and triggers. Course includes a group project involving the design and implementation of a relational database and embedded SQL programming. Prerequisites: CS 311, MATH 322.

CS 697. Selected Topics (1-3).
1-3 Classroom hours; 0-2 Lab hours. Selected topics of current interest. Repeatable for credit with departmental consent. Prerequisite: departmental consent.

CS 697AG. Introduction to Intelligent Robotics (3).
The study of intelligent robotics allows robots to gather information from surrounding environments and take actions autonomously. This course will introduce the fundamental principles and methods of manipulation, navigation and perception for intelligent robotics. Topics covered include geometry transformations, kinematics, dynamics, localization, navigation, mapping, motion planning, intelligent processing, smart sensing, decision making, and robotic intelligence. The students will explore the robot concepts and algorithms, such as dexterous manipulation, simultaneous localization and mapping (SLAM), and autonomy, while working with Nao humanoid robots and Sawyer collaborative robots. Prerequisites: CS 300, MATH 511, IME 254.

CS 697AI. Static Program Analysis (3).
Graduate and senior undergraduate level, project-based class in programming languages implementation. Studies techniques and algorithms that can be used to understand programs. Discusses classical and recent research papers during class meetings. Looks into applications of these techniques to detect bugs and security vulnerabilities in Android apps. Students finish a static analysis project and submit a short paper. Prerequisite: CS 300.

CS 697AJ. Computer & Network Security (3).
First course on computer and network security. Introduces security design principles, cryptography, cyber operations, risk analysis, vulnerability models, as well as ethical and legal issues. Students apply basic computer and network defense strategies using software tools. Prerequisite: CS 464.

CS 697AN. Hardware-Based Computer Security (3).
Intended for seniors and graduate students who want to study and explore the role of hardware in improving computer security. Topics covered may include (1) elements of computer security, (2) secure coprocessor, (3) secure bootstrapping, (4) secure memory management, (5) hardware-based authentication, (6) hardware-based virus detection, (7) hardware as a cybersecurity solution, (8) security engineering, (9) managing the development of secure systems, and (10) system evaluation and assurance. Prerequisites: CS 394 and a desire to learn more about both computer architecture and security.

CS 697AO. Computational Study of Nanomaterials (3).
Introduces computational approaches for analyzing nanomaterials to undergraduate and graduate students. Lectures focus on fundamentals of nanomaterials, characterizations of nanoparticles, parallel computing, and computer-based simulation of nanomaterials. Laboratory assignments provide hands-on real-life experiences. Special attention is given to team-based project activities. Prerequisites: CHEM 211, and either CS 211 or ME 325.

CS 715. Compiler Construction (3).
First compiler course for students with a good background in programming languages and sufficient programming experience. Covers compiler design, lexical analysis, parsing techniques, symbol tables, scope analysis, type checking and conversion; run-time organization, code generation and optimization. Project-oriented
course involves implementation of a full compiler for a simplified but nontrivial procedural language. Prerequisites: CS 238, 510.

CS 720. Theoretical Foundations of Computer Science (3).
Provides an advanced level introduction to the theoretical bases of computer science. Computer science theory includes the various models of finite state machines, both deterministic and nondeterministic, and concepts of decidability, computability and formal language theory. Prerequisite: CS 322.

CS 721. Advanced Algorithms and Analysis (3).
Topics include height-balanced trees, graph algorithms, greedy algorithms, dynamic programming, hard problems and approximation algorithms. Prerequisite: CS 560.

Introducory class on applying various mathematical tools to the field of computer networks and related areas. Divided into three phases: phase one covers the fundamentals of probability, statistics and linear algebra required for understanding the core concepts to follow. Phase two covers the core topics of optimization and queuing theory. Phase three briefly covers the advanced topics of game theory and information theory. The depth of coverage is sufficient to allow students to read and understand research papers in computer networking and related areas that use these standard techniques. Ideas are taught through intuition, mathematically correct formalization and detailed numerical examples. Prerequisite: MATH 243. Corequisite: CS 464.

CS 732. Data Communication Networks (3).
Presents a quantitative performance evaluation of telecommunication networks and systems. Includes fundamental digital communications system review; packet communications, queuing theory, OSI, s.25 and SNA layered architectures, stop-and-wait protocol, go-back-N protocol, and high-level data link layer; network layer flow and congestion control, routing, polling and random access, local area networks (LAN); integrated services digital networks (ISDN), and broadband networks. Prerequisite: CS 464.

CS 737. Wireless Networking (3).
Covers topics ranging from physical layer to application layer in the wireless and mobile networking fields. Explores physical layer issues of wireless communications, wireless cellular telephony, ad-hoc networks, mobile IP and multicast, wireless LAN (IEEE 802.11), security, Bluetooth and WAP, etc. Imparts general knowledge about wireless communication technologies and ongoing research activities. Prerequisite: CS 736.

CS 738. Embedded Systems Programming (3).
Studies the requirements and design of embedded software systems. Application of the C programming language in the implementation of embedded systems emphasizing real-time operating systems, interfacing to assembly and high-level languages, control of external devices, task control and interrupt processing. Prerequisite: CS 594.

CS 750. Workshop in Computer Science (1-5).
Short-term courses with special focus on introducing computer science concepts. Repeatable for credit. Prerequisite: departmental consent.

CS 764. Routing and Switching I (4).
3 Classroom hours; 2 Lab hours. An introductory course which studies different hardware technologies, like Ethernet and token ring. Discusses VLSM. Introduces different routing protocols. Includes hands-on experience in the CS department's routing and switching lab. Prerequisite: CS 464 or 736.

CS 766. Information Assurance and Security (3).
Provides basic concepts in information assurance and security including encryption, digital certificates, security in networks, operating systems and databases. Topics in intrusion detection, legal and ethical issues in security administration are also discussed. Prerequisite: CS 464 or 736 or 764.

CS 767. Foundations of Network Security (3).
Presents fundamental concepts in cryptography and network security, and discusses applications and protocols for providing confidentiality, authentication, integrity, and availability in networking services and systems. Includes review of symmetric-key cryptographic schemes such as DES and AES, public-key cryptographic schemes such as RSA and Diffie-Hellman key exchange protocol, cryptographic hash functions such as SHA, message authentication codes such as HMAX digital signature schemes such as El-Gamal and DSS, kerberos and user authentication protocols, transport layer security and TLS, IP layer security and IPSec, and wireless security principles. CS 766 is highly preferred, but not required. Prerequisite: CS 464 or 736.

CS 771. Artificial Intelligence (3).
Introduction to some of the fundamental concepts and techniques underlying artificial intelligence. Topics covered include state spaces, heuristic search, game playing, knowledge representation, and resolution in propositional and first-order predicate logic. Prerequisite: CS 560.

CS 780. Advanced Software Engineering (3).
Discuss advanced topics in software development, maintenance and evolution. Topics include software design patterns, architecture and architectural styles, frameworks, refactorings, and static and dynamic analyses. Includes a group project. Prerequisite: CS 480.

CS 781. Cooperative Education (1-3).
Practical experience in a professional environment to complement and enhance the student's academic program. For master's level CS students. Repeatable, but may not be used to satisfy degree requirements. Graded Cr/NCr. Prerequisites: departmental consent and graduate GPA of 3.000 or above.

CS 794. Multicore Architectures and Programming (3).
3 Classroom hours. Introduces state-of-the-art concepts and techniques to design and program modern computer systems. Particular attention is given to the following areas: multicore architecture, parallel programming and advanced research. Labs give hands-on experience. Prerequisites: CS 211, 394.

CS 797. Special Topics in Computer Science (1-4).
New or special courses presented on sufficient demand. Repeatable for credit. Prerequisite: departmental consent.

CS 798. Individual Projects (1-3).
Allows beginning graduate students and mature undergraduate students to pursue individual projects of current interest in computer science. Graded S/U. Prerequisite: departmental consent.