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. Introduces 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(s): MATH 111 or equivalent. Corequisite(s): CS 194L.

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(s): MATH 111. Corequisite(s): CS 211L.

CS 238. Assembly Language Programming (3).
Introduces 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(s): 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(s): 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(s): CS 211. Corequisite(s): CS 311L.

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(s): MATH 242 or equivalent with a grade of 2.000 or better.

CS 322. Discrete Structures II (3).
Cross-listed as MATH 322. Continuation of Discrete Structures I. Includes relations, graphs, trees, Boolean algebra and automata. Prerequisite(s): MATH 321 or CS 321.

CS 338. FPGA-Based System Design (4).
3 Classroom hours; 2 Lab hours. Introduces digital design concepts using field programmable gate arrays (FPGAs). Includes programmable logic devices, FPGA architecture, interconnect, digital design challenges, digital design process, and integrated circuit fabrication process. Presents digital design flow using FPGAs, and other technologies associated with field programmable gate arrays. Introduces the concept of Verilog programming. Uses CAD tool for circuit simulation. Prerequisite(s): CS 194 and CS 211. Corequisite(s): CS 338L.

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(s): 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. Prerequisite(s): CS 194, 211.

CS 400. Data Structures (4).
3 Classroom hours; 2 Lab hours. Introduces basic data structures and covers their implementations using classes in C++. Includes lists, stacks, queues, binary trees and hash tables. Prerequisite(s): CS 311 with a C- grade or better. Corequisite(s): CS 400L.

CS 400L. Data Structures Lab I (0).
Applies concepts from CS 400 in a lab setting. Corequisite(s): CS 400.

CS 410. Programming Paradigms (3).
Overview of different programing paradigms, including their philosophies, uses and relative advantages/disadvantages. Covers the procedural/imperative, functional, logic and object-oriented paradigms. Includes programming assignments in the functional and logic paradigms. Prerequisite(s): CS 311.

CS 444. Linux Essentials (3).
Fundamentals of the Unix/Linux operating system. Topics include Linux file systems, essential commands, best security practices, and introduces shell programming. Prerequisite(s): CS 211.

CS 481. Cooperative Education (1-3).
Provides a field placement that integrates theory with a planned and supervised professional experience designed to complement and enhance the student's academic program. Individualized programs must be formulated in consultation with, and approved by, appropriate faculty sponsors. Repeatable for credit. Prerequisite(s): departmental consent.

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(s): 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. Prerequisite(s): 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(s): departmental consent.

CS 498. Individual Projects (1-3).
Repeatable for a total of 6 credit hours. Prerequisite(s): 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. Prerequisite(s): CS 311, MATH 322.
Fundamental principles of modern operating systems. CPU management including processes, threads, scheduling, synchronization, resource allocation and deadlocks. Memory management including paging and virtual memory. Storage management and file systems. Prerequisite(s): 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. Prerequisite(s): CS 322, 400; STAT 460 or IME 254.

CS 580. Introduction to Software Engineering (3).
Introduces the processes, methods and tools used in software development and maintenance. Topics include software development life cycle and processes, configuration management, requirements gathering, OOA/D with UML, cohesion and coupling, and unit testing. Prerequisite(s): CS 311.

CS 581. Programming for Computing (3).
Fast-paced course in computer programming in Python. Emphasizes problem solving and object-oriented programming principles. Topics include language syntax, built-in data structures, functions, classes, modules and exception handling. Not for BSCE, BSCS or BSEE credit. Prerequisite(s): MATH 242.

Fast-paced course that introduces basic data structures, algorithms, and introduces analysis of space and time complexity. Topics includes sorting algorithms, greedy algorithms, linked lists, stacks, queues, binary search trees, asymptotic notation, substitution and recursion tree method, master theorem, and order statistics. Not for BSCE, BSCS or BSEE credit. Prerequisite(s): CS 581.

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. Prerequisite(s): CS 238, 394. Corequisite(s): CS 594L.

CS 656. Introduction to Cybersecurity (3).
Provides fundamental concepts in cybersecurity including cryptography, security in networks, operating systems and databases. Topics in intrusion detection, security administration, and legal and ethical issues are also discussed. Prerequisite(s): CS 664 or CS 764.

CS 664. Computer Networks (3).
Introductory course on computer networking. Introduces concepts and protocols in various network layers with emphasis on the internet. Topics covered include: physical layer (wired and wireless), medium access control and data link layers, packet switching and routing (IP), routing protocols, transport layer (TCP, UDP), congestion and flow control, basic network security, and network applications. Prerequisite(s): undergraduate students: IME 254 and CS 311; graduate students: object oriented programming and statistics/probability knowledge.

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. Prerequisite(s): 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(s): 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. Course introduces 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. Explores 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. Prerequisite(s): CS 300, MATH 511, IME 254.

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 bootstrap loading, (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. Prerequisite(s): CS 394 and a desire to learn more about both computer architecture and security.

CS 697AP. Applied Parallel Computing (3).
This course is to teach how to program parallel computers to efficiently analyze challenging problems with enormous datasets. Two distinct approaches will be introduced which can be used to solve problems in all manner of domains including data analytics and machine learning. The first approach to be studied will be embarrassingly parallel in nature while the second approach will leverage fine-grain parallelism. Prerequisite(s): CS 394 or Instructor’s consent.

CS 697AQ. Web Programming (3).
Hands-on introduction to web programming. Prepares students to create webpages and develop web applications that integrate with a backend database. Topics covered include client-side technologies that run in the web browser (HTML, CSS and JavaScript), and server-side technologies that run on the server (Node.js or PHP and SQL). A strong programming background is preferred for successful completion of several practical exercises contained in the course. Prerequisite(s): CS 311.

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. Prerequisite(s): 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(s): 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(s): CS 560.

Introductory 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 topics 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(s): MATH 243. Corequisite(s): CS 664.

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(s): CS 664.

CS 746. Perspectives on Data Science (3).
Covers the perspectives and fundamentals of data science. Topics include data collection, pre-processing, transformation, exploratory data analysis, visualization, predictive modeling, descriptive modeling, clustering, regression and classification and data science project life cycle. This course is limited to engineering students and students in other colleges majoring in data science/analytics related programs. Prerequisite(s): IME 254 and CS 211 for undergraduate students; instructor's consent for graduate students.

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

CS 764. Routing and Switching I (4).
3 Classroom hours; 2 Lab hours. 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(s): CS 664.

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. Prerequisite(s): CS 664; the department also highly recommends CS 656, but it is not required.

CS 771. Artificial Intelligence (3).
Introduces 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(s): CS 560.

CS 780. Advanced Software Engineering (3).
Systematically learns and practices all workflows of the Unified Process. Discusses advanced topics in software development, maintenance and evolution. Topics include software design patterns, architecture and architectural styles, frameworks, refactoring, and static and dynamic analyses. Includes a group project. Prerequisite(s): CS 580.

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 for credit, but may not be used to satisfy degree requirements. Prerequisite(s): departmental consent and graduate GPA of 3.00 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. Prerequisite(s): CS 211, 394.

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