Computer Science

Prerequisite courses must be passed with a minimum grade of C.

LOWER DIVISION

CS 100. Critical Thinking with Computers (3). Apply critical thinking skills studying human and computer parallels, computer technology and methodology, and program development. [GE.]

CS 111. Computer Science Foundations 1 (4). Introductory programming covering problem decomposition, control structures, simple data structures, testing, and documentation. Students design and implement a number of programs. [Prereq: MATH 115 (C).]

CS 112. Computer Science Foundations 2 (4). Object-oriented programming, focusing on classes, instances, methods, encapsulation, inheritance, overloading, multiple inheritance, and exception handling. [Prereq: CS 111. Weekly: 3 hrs lect, 2 hrs lab.]

CS 211. Data Structures (4). Introduction to classic data structures and algorithms. Performance comparisons, bit-O notation, trade-offs, arrays, linked lists, recursion, sorting, stacks, queues, trees, graphs, and hash tables. [Prereq: CS 112 and MATH 253.]

CS 212. Algorithms (4). Introduction to algorithmic thinking. Recurrences and solution techniques, fundamental algorithms including graph algorithms, algorithm design techniques, balanced trees, performance trade-offs. [Prereq: CS 211, STAT 108 (C), and MATH 105 or MATH 109.]

CS 232. Python Programming (3). Introduction to the Python language. Idiomatic language features such as lists, dictionaries, tuples, and sets. Use of Python classes and modules to accomplish complex tasks. [Prereq: CS 111.]

CS 233. Computer Organization (3). Principles of computer architecture from a layered point of view, including data representation, machine language execution, addressing modes, and symbolic assembly language. Fundamental notions of operating systems, interfacing, and communication are also introduced. [Prereq: CIS 132 or IA. Weekly: 2 hrs lect, 2 hrs lab.]

CS 234. Computer Architecture (3). A study of the design of computers. Topics include the design of combinatorial and sequential circuits, design methodology of a basic computer, central processor organization, microprogramming, memory organization, input-output organization, and arithmetic processor design. [Prereq: CS 233.]

CS 235. Java Programming (3). Object orientation; event handling; abstract windowing toolkit applets; applications; Java database connectivity; applications programming interface and Java doc. [Prereq: CS 112. Service fee.]

CS 236. Algorithms (3). Introduction to key algorithmic concepts and constructs. Algorithmic development, tracing, and analysis. Algorithm construction and analysis in both non-executable contexts and within programming environments. [Prereq: MATH 253.]

CS 237. Bioinformatics Programming (3). Introductory course on using software tools to solve biological problems. Students collaboratively model genomic and/or proteomic data with scripting and statistical languages. [Prereq: CS 111 and BIOL 105.]

CS 243. Architecture (4). Introduction to computer architecture including assembly language, computer arithmetic, performance measures, datapath, control, pipelining, and memory/storage design. [Prereq: CS 112 and MATH 253.]

CS 274. Operating Systems (4). Introduction to operating systems with an emphasis on process synchronization and control. Synchronization, kernel structure, scheduling, deadlock, virtual and physical memory, file and I/O. [Prereq: CS 211 and CS 243.]

CS 279. Introduction to Linux (4). Introduces the UNIX/Linux family of operating systems. Basic commands, utilities, system structures, scripting and tools are explored. Elements of system administration are presented. [Prereq: CS 111.]

CS 280. Selected Topics in Computing (1-3). Special topics in computer science. [Courses with this number have only freshman/sophomore prerequisites, excluding CS 212 and CS 243. Rep.]

CS 280L. Selected Topics in Computing (1-2). Special topics in computer science. [Courses with this number have only freshman/sophomore prerequisites, excluding CS 212 and CS 243. Rep.]

UPPER DIVISION

CS 309. Computers & Social Change (3). How computers influence societal systems. Issues: privacy, employment, politics, social interaction, and risk. Group discussion and writing on selected issues. [GE. CWT.]

CS 310. Database for Non-Majors (4). Concepts/applications for non-computer science majors.

CS 315. Database Design & Implementation (3). Design/implementation concepts for relational model. Enterprise and entity-relationship modeling. Schema development: normalization; SQL data definition and data manipulation language; user-defined types, rules, and triggers to support the schema. Features to support integrity, ease of use, and control: concurrency, locking, distribution, performance. [Prereq: CIS 260 or CS 233, and MATH 253 recommended. Weekly: 2 hrs lect, 2 hrs lab.]

CS 318. Programming Database Applications (3). 4th generation language tools. Ad hoc interaction with database using SQL. Program SQL scripts; design applications using forms and menus; program an application using form and menu structures; program with a report generator; access the database from a procedural language. [Prereq: CIS 315/CS 315 and MATH 253. Weekly: 2 hrs lect, 2 hrs lab.]

CS 325. Database Design (4). Introduction to database design and implementation. Relational model, entity-relationship model and diagrams, converting a model to a schema, elementary Structured Query Language (SQL), normalization. [Prereq: CS 112.]

CS 328. Web-Apps Using Databases (4). Building applications atop databases. N-tiered architecture; database tier: stored procedures/functions; presentation tier: web GUIs; application tier: controlling web-to-database interactions. [Prereq: CS 325.]

CS 334. Operating Systems and Architecture (3). An in-depth treatment of computer architecture, technology choices, and the operating system interface with the hardware, the application, and the system user. [Prereq: CS 233 or IA. Weekly. Rep.]

CS 335. Programming Languages: Principles and Paradigms (3). An in-depth treatment of programming languages, including their history, data types, data control, sequence control, run-time storage, language translation, and semantics. Paradigms include procedural, functional logic, and object-oriented programming. [Prereq: CS 233 or IA. Rep.]

CS 346. Telecommunications & Networks (4). Introduction to the fundamentals of telecommunication and to the structure, implementation, and theoretical underpinnings of computer networking. [Prereq: CS 243 and STAT 108.]

CS 372. Telecommunications (3). Data communications principles and applications; administering and managing communications systems. Protocols, networks, communication hardware, design, performance analysis. [Prereq: CS 233 or IA. Weekly: 2 hrs lect, 2 hrs lab.]

CS 373. Network Design & Implementation (3). Comprehensively examine network design standards, communication protocols, configuration and management methods, security, and traffic analysis. Practical lab activities with tools and equipment. [Prereq: CIS 372/CS 372 recommended.]

CS 435. Software Engineering (3). Introduction to software engineering principles, including discussion of development methodologies, requirements, analysis, project planning, software design, construction, management, and quality assurance. [Prereq: CS 334 and CS 335.]

CS 436. Theory of Computation (3). A study of formal models of computation, such as finite state automata, pushdown automata, and Turing machines. Elements of formal languages to be examined include regular expressions, context-free languages, recursively-enumerable languages, undecidability, and NP-completeness. [Prereq: CS 236; CS 335 recommended.]

CS 444. Robotics (4). A project-based introduction to robotic systems and software that controls them, including gearing, mechanics, AI control systems, and problem solving with robots. [Prereq: CS 211 and STAT 108.]

CS 449. Computer Security (4). Introduction to central concepts of computer security on networked systems. Topics include threats, cryptography, authentication, operating systems in security, legal and privacy issues. [Prereq: CS 346.]

activ activity; (C) may be concurrent; coreq corequisite(s); CR/NC mandatory credit/no credit; CWT communication & ways of thinking; DA dept approval

CS 458. Software Engineering (4). Introduction to software engineering principles and methodologies in the context of a semester-long software team project. [Prereq: CS 274 and CS 328.]

CS 461. Computational Models (4). An introduction to the Chomsky hierarchy, automata, Church-Turing Thesis, computability, NP-Completeness, and information theory. [Prereq: CS 212, MATH 253, and MATH 105 or MATH 109.]

CS 475. Geographic Information Systems (3). GIS applications. Vector- and raster-based systems. Layering techniques, representation methods, analytical operations, information management/integration. [Weekly: 2 hrs lect, 2 hrs lab.]

CS 480. Advanced Topics in Computing (1-3). Advanced topics in computer science. [Courses with this number must have as a prerequisite at least CS 211. Rep.]

CS 480L. Advanced Topics in Computing (1-2). Advanced topics in computer science. [Courses with this number must have as a prerequisite at least CS 211. Rep.]

CS 482. Internship (1-4). Supervised experience in business, governmental, or service agencies, matching theory with practice. [CR/NC. Prereq: IA. Weekly: 3 hrs per unit of credit.]

CS 492. Systems Design & Implementation (3). Apply computer programming and implementation concepts to comprehensive group project. Use management planning and scheduling tools; practice assessing and reporting progress; develop, test, quality assure software; develop documentation. CIS majors only. [Prereq: CS 318/CIS 318, CS 350/CIS 350, CS 372/CIS 372, CS 450/CIS 450. All prereqs must be completed with C or above. Weekly: 2 hrs lect, 2 hrs lab.]

CS 499/CIS 499. Directed Study (1-4). Individual study on selected topics. Open to advanced students with consent of faculty sponsor and DA. [Rep by topic for a maximum of 12 units; multiple enrollments in term.]