

CATALOG DESCRIPTIONS OF COMPUTER SCIENCE COURSES

CPSC 110. Programming I. (3-2). Credit 4. I, II, S Basic concepts, nomenclature and historical perspective of computers and computing; internal representation of data; software design principles and practices; structured programming in Pascal; use of terminals, operation of editors and execution of student-written programs. Prerequisite: None.

CPSC 111. Introduction to Computer Science Concepts and Programming. (3-2). Credit 4. Basic concepts, nomenclature, and historical perspective of computers and computing; problem solving and software design principles, including abstraction, modularity, data representation, documentation, portability, structured and object oriented programming; software engineering concepts including requirements definition, testing, and maintenance considerations; development and execution of student written programs. Prerequisite: Course in Pascal, C, or Java (high school or college) or approval of instructor.

CPSC 181: Introduction to Computing. (1-0). Credit 1. Seminar to introduce freshman level students to the broad field of computing. Achieved through presentations from industry and academia about how computer science concepts are used in research and end products, as well as through general lectures on computing. Includes a major writing component.

CPSC 121: Introduction to Program Design and Concepts. Credit 4. Computer programming and design concepts; use of primitive types, control structures, vectors, strings, structs, classes, functions, file I/O, exceptions, and other programming constructs for problem solving with computers; introduction to program structure, design and development, plus the use of class libraries. Includes the execution of student written programs in C++.

CPSC 203. (COSC 1317, 1417, ENGR 2304) Introduction to Computing. (3-0). Credit 3. I, II, S Algorithms, programs and computers; basic programming and program structure; data representation; computer solution of numerical and non-numerical problems using a high-level programming language, FORTRAN.

CPSC 206. Structured Programming in C. (3-0). Credit 3. Basic concepts, nomenclature and historical perspective of computers and computing; internal representation of data; software design principles and practices; structured programming in C; use of terminals, operation of editors and execution of student-written programs.

CPSC 211. Data Structures and Their Implementations. (3-2). Credit 4. Specification and implementation of basic data structures and abstract data types--linked lists, stacks, queues, trees and tables; performance tradeoffs of different implementations; asymptotic analysis of running time and memory usage; compares and contrasts object-oriented language (typically, Java) and non-object-oriented languages (typically, C); emphasis on adherence to good software engineering principles. Prerequisite: CPSC 111 or approval of instructor.

CPSC 221. Data Structures and Algorithms. (4.0). Credits 4. Specification and implementation of basic abstract data types and their associates algorithms – stacks, queues, lists; sorting and selection; searching; graphs; hashing. Performance tradeoffs of different implementations; asymptotic analysis of running time and memory usage. Includes the execution of student written programs in C++. Prerequisite: CPSC 121; Corequisite: MATH 302.

CPSC 291. Research. Credit 1 to 4. Research conducted under the direction of faculty member in computer science. May be repeated 2 times for credit. Prerequisites: Freshman or sophomore classification and approval of instructor.

CPSC 310. Database Systems. (3-0). Credit 3. I, II, S File structures and access methods; database modelling, design and user interface; components of database management systems; information storage and retrieval, query languages, high level language interface with database systems.

CPSC 311. Analysis of Algorithms. (3-0). Credit 3. I, II, S Design of computer algorithms for numeric and non-numeric problems; relation of data structures to algorithms; analysis of time and space requirements of algorithms; complexity and correctness of algorithms. Prerequisites: CPSC 211; MATH 302.

CPSC 312. Computer Organization. (4.0). Credit 4. Introduces computer systems from a programmer's perspective, providing a complete view of the hardware and interface to system software. The course covers data representations, processor architecture, assembly code, machine level representations of C++ programs, program optimizations, the memory hierarchy, linking exceptional control flow (exceptions, interrupts, processes, and signals), performance measurement, virtual memory and memory management and system-level I/O. These concepts are supported by series of hands-on lab assignments. Prerequisite: CPSC 221

CPSC 313. Introduction to Computer Systems. (4.0). Credit 4. Provide students with a general understanding of what system software support is necessary for an application program to run, both on a single node and over a network. Prepare students to do simple system-level and network programming. How to use (instead of design) system components such as memory, file systems, process control, interprocess communication, and networking. Students will have an understanding of the problems and pitfalls typically encountered in multithreaded and network systems and application design. Prerequisite: CPSC 312.

CPSC 314: Programming Languages. (4.0). Credit 4. The course explores the design-space of programming languages via an in-depth study of two programming languages, on object-oriented (Java), on functional (Haskell). With both languages, the course focuses on idiomatic uses of the languages, and language features characteristic for each language. Programming language concepts covered include different forms of polymorphism, concurrency static typing, higher-order functions and closures, and pattern matching. Prerequisite: CPSC 221.

CPSC 315: Programming Studio. (3.0). Credit 3. Provide an intensive programming experience that integrates core concepts from the preceding computer science course, and also exposes the students to a variety of programming/development tools. Students learn to design and write programs of higher complexity. Introduction to software development (including tools, multi-component coding style). Several team projects incorporating a variety of programming topics. Oral and written reporting required. Prerequisite: CPSC 312, CPSC 314. Corequisite: CPSC 313.

CPSC 321. Computer Architecture. (3-2). Credit 4. I, II, S Basic hardware/software components, assembly language, and functional architecture of computers; syntax and semantics of a typical microprocessor assembly language; instruction sets, construction and execution of an assembly program; the design of I/O modules, memory, control unit and arithmetic unit. Prerequisites: ECEN 220 or 248.

CPSC 332. Programming Language Design. (3-0). Credit 3. I, II Design of high-level languages; criteria for language selection; specification techniques for syntax and semantics; trends in high-level language design and introduction to programming in LISP.

CPSC 410. Operating Systems. (3-0). Credit 3. I, II, S Hardware/software evolution leading to contemporary operating systems; basic operating systems concepts; methods of operating systems design and construction; algorithms for CPU scheduling, memory and general resource allocation; process coordination and management; case studies of several operating systems. Prerequisite: CPSC 321.

CPSC 411. Design and Analysis of Algorithms. (3-0). Credit 3. Study of computer algorithms for numeric and non-numeric problems; design paradigms; analysis of time and space requirements of algorithms; correctness of algorithms; NP-completeness and undecidability of problems. Prerequisites: CPSC 221, CPSC 222, CPSC 315.

CPSC 420. Artificial Intelligence. (3-0). Credit 3. I, II, S Fundamental concepts and techniques of intelligent systems; representation and interpretation of knowledge on a computer; search strategies and control; active research areas and applications such as notational systems; natural language understanding, vision systems, planning algorithms, intelligent agents and expert systems. Prerequisite: CPSC 411.

CPSC 431. Software Engineering. (2-2). Credit 3. I, II, S Application of engineering approach to computer software design and development; life cycle models software requirements and specification; conceptual model design; detailed design; validation and verification; design quality assurance; software design/development environments and project management.

CPSC 433. Formal Languages and Automata. (3-0). Credit 3. I Basic types of abstract languages and their acceptors, the Chomsky hierarchy; solvability and recursive function theory; application of theoretical results to practical problems. Prerequisite: CPSC 311.

CPSC 434. Compiler Design. (3-0). Credit 3. II Programming language translation: functions and general organization of compiler design and interpreters; theoretical and implementation aspects of lexical scanners; parsing of context free languages; code generation and optimization; error recovery. Prerequisite: CPSC 411 or permission of instructor.

CPSC 435. Structured Programming in Ada. (3-0). Credit 3. The Ada programming language; history and motivation; scalar and composite types; type and object attributes; control constructs; subprograms; packages and abstract types; numeric types; I/O; program structure; overloading and visibility; tasking; generics; programming style using Ada, Ada Programming Support Environments; bindings to common utilities, including GKS, SQL.

CPSC 436. Computer-Human Interaction. (3-0). Credit 3. Comprehensive study of the Computer-Human Interaction (CHI) area; includes history and importance of CHI; CHI design theories; modeling of computer users and interfaces; empirical techniques for task analysis and interface design; styles of interaction and future directions of CHI including hypermedia and computer-supported collaborative work.

CPSC 437. Engineering Software Products. (3-0). Credit 3. Links theory and practice in providing hands-on experience in development of growth-oriented new software products; student teams prepare and present a plan for a new software product; skills conducive to new software product success are developed including team building; organizing; planning; integrating and persuading. Prerequisites: Senior classification and approval of instructor.

CPSC 438. Distributed Objects Programming. (3-0). Credit 3. Principles of distributed computing and programming with current paradigms, protocols, and application programming interfaces including Sockets, RMI, CORBA, IDL, Servlets, Web Services; security issues with public/private keys, digital signatures, forms and GUI based applications with multi-tier components, database connectivity and storing/streaming data structured using XML. Prerequisites: CPSC 332 or approval of instructor; junior or senior classification.

CPSC 440. Quantum Algorithms. (3-0). Credit 3. Introduction to the design and analysis of quantum algorithms; basic principles of the quantum circuit model; gives a gentle introduction to basic quantum algorithms; reviews recent results in quantum information processing.

CPSC 441. Computer Graphics. (3-0). Credit 3. I, II, S Principles of interactive computer graphics; systems organization and device technologies for raster and vector displays; 2D and 3D viewing, clipping, segmentation and interaction handling. 3D geometrical transformations, projections and hierarchical data structures for graphics modeling.

CPSC 442. Scientific Programming. (3-0). Credit 3. II Introduction to numerical algorithms fundamental to scientific and engineering applications of computers; elementary discussion of error; algorithms, efficiency; polynomial approximations, quadrature and systems of algebraic and differential equations. Prerequisites: CPSC 120, MATH 308.

CPSC 444. Structures of Interactive Information. (3-0). Credit 3. A systems approach to the programming design, authoring and theory of hypermedia; object-oriented visual and interactive programming; visual design, including color, space, text and layering; the reference as a metadisciplinary structure; collecting and sampling; ontologies, maps and navigation as means of structuring information; create dynamic hypermedia that is expressive and interpretive. Prerequisites: CPSC 315.

CPSC 452. Robotics and Spatial Intelligence. (3-0). Credit 3. II Algorithms for executing spatial tasks; path planning and obstacle avoidance in two and three dimensional robots--configuration space, potential field, free-space decomposition methods; stable grasping and manipulation; dealing with uncertainty; knowledge representation for planning--geometric and symbolic models of the environment; task-level programming; learning. Prerequisite: CPSC 420.

CPSC 462. Microcomputer Systems. (2-2). Credit 3. II Microcomputers as components of systems; VLSI processor and co-processor architectures, addressing and instruction sets; I/O interfaces and supervisory control; VLSI architectures for signal processing; integrating special purpose processors into a system. Prerequisite: CPSC 410 or concurrent enrollment.

CPSC 463. Networks and Distributed Processing. (3-0). Credit 3. I, II Basic hardware/software, architectural components for computer communications; computer networks, switching, routing, protocols and security; multiprocessing and distributed processing; interfacing operating systems and networks; case studies of existing networks and network architectures. Prerequisite: CPSC 410.

CPSC 469. Advanced Computer Architecture. (3-0). Credit 3. Introduction to advanced computer architectures including memory designs, pipeline techniques, and parallel structures such as vector computers and multiprocessors. Prerequisite: CPSC 321 or ECEN 350. Cross-listed with ECEN 469.

CPSC 470. Information Storage and Retrieval. (3-0). Credit 3. Representation of, storage of and access to very large multimedia document collections; fundamental data structures and algorithms of current information storage and retrieval systems.

CPSC 481. Seminar. (0-2). Credit 1. I, II, S Investigation and report by students on topics of current interest in computer science. Prerequisite: Junior or senior classification.

CPSC 482: Senior Capstone Design. (3.0). Credit 3. Project-based course to develop system integration solving real-world problems in Computer Science; significant team software project that integrates advanced concepts across Computer Science specializations; projects require design, documentation and demonstration, as well as design methodology, management process. Prerequisites: Senior classification and at least two courses from one track including CPSC 411.

CPSC 483. Computer Systems Design. (1-6). Credit 3. Engineering design; working as a design-team member, conceptual design methodology, design evaluations, total project planning and management techniques, design optimization, systems manufacturing costs considerations; emphasis placed upon students' activities as design professionals. Prerequisites: CPSC 431 and CPSC 462 and senior classification.

CPSC 485. Directed Studies. Credit 1 to 6. I, II, S Permits work on special project in computer science. Project must be approved by the department. Prerequisite: Senior classification.

CPSC 489. Special Topics in ... Credit 1 to 4. Special topics in computer science that are new or unique that are not covered in existing courses.

CPSC 491. Research. Credit 1 to 4. Research conducted under the direction of faculty member in the computer science department. May be repeated 2 times for credit. Prerequisites: Junior or senior classification and approval of instructor.