

COMPUTER SCIENCE MINOR

Computer Science

Department of Integrated Engineering
College of Science, Engineering & Technology
131 Trafton Science Center N • 507-389-2744
Email: integrated.engineering@mnsu.edu
Websites: cset.mnsu.edu/ie and cset.mnsu.edu/cs

Chair: Dean Kelley

Faculty: Rebecca Bates, Dean Kelley

The field of computer science spans a wide range of topics from theoretical and algorithmic foundations to cutting-edge development in computer hardware and software. A computer science minor prepares students to apply the tools and theory of computer science to whatever their major field of study is. Applications in biology, physics, chemistry, engineering, cognitive science and the social sciences can benefit from a deeper understanding of computer science.

Academic Map/Degree Plan at www.mnsu.edu/programs/#All

POLICIES/INFORMATION

GPA Policy. A GPA of 2.5 or higher in courses required for the minor is required for graduation with the minor.

Grading Policy. All coursework applied towards the minor must be taken for a letter grade except for course offered only as P/N. A minimum grade of "C-" is required in all courses which are to be applied towards a minor. In addition, a minimum grade of "C-" is required for all prerequisite courses. Grades of "D" are not accepted by the department.

Incomplete Policy. An incomplete grade for a course will generally be given only under two conditions. The first condition is illness—a doctor's written recommendation must be supplied. The second condition arises when a death in the student's family has caused the student to be away from the campus for an extended period of time. The student must have a satisfactory grade ("C" or better) in the course at the time of the onset of the condition.

Residency. At least 50 percent of the computer science credits required for a minor from this department must be earned from the Computer Science program at Minnesota State Mankato when using transfer credits. Students receiving a computer science minor must take at least 15 credits of Computer Science courses, which may include CS 201W, CS 293, CS 493, and CS 495. These classes may allow a student to fulfill the residency requirement but do not meet other requirements of the minor.

COMPUTER SCIENCE MINOR

Minor Core

MATH 121 Calculus I (4)

(choose 1 option)

CS Option

CS 110 Computer Science I (4)
CS 111 Computer Science II (4)
CS 305 Algorithmic Structures (4)

EE/CE Option

This option is recommended for students majoring in electrical or computer engineering.

EE 106 Introduction to Electrical/Computer Engineering I (3)
EE 107 Introduction to Electrical/Computer Engineering II (3)
CS 111 Computer Science II (4)
CS 305 Algorithmic Structures (4)

Minor Electives (choose 6-7 credits)

(choose 2 classes from the following)

CS 230 Introduction to Intelligent Systems (4)
CS 350 Network Architectures (3)
CS 430 Artificial Intelligence (3)
CS 460 Operating Systems: Design and Implementation (3)

COURSE DESCRIPTIONS

CS 105 (3) Computer Science Foundations

This course provides fundamental conceptual, mathematical, and logical tools for students wishing to major in Computer Science. Topics include hardware concepts, number systems, computer arithmetic, counting, sets and functions, logic, simple induction, etc. Co-requisite: Math 112
Fall

CS 110 (4) Computer Science I

Students will learn programming skills in object-oriented C++. Students will design algorithms and learn how to write, compile, run and debug programs that include selection and repetition structures, functions, and arrays. Study skills and professional development will be addressed.

Prerequisite: MATH 112 (College Algebra)

Fall, Spring

CS 111 (4) Computer Science II

Continues the exploration of introductory Computer Science begun in CS 110. Focus is on developing basic knowledge of algorithms, programming skills and problem solving techniques. Topics include recursion, sorting, linked lists, stacks and queues. Prerequisite: CS 110 or EE 107. MATH 113 or MATH 115 or MATH 121

Fall, Spring

CS 171 (2) Introduction to C++ Programming

This course provides an introduction to programming using C++. Emphasis on structured programming concepts, with a brief discussion of object-oriented programming. Control structures, expressions, input/output, arrays and functions. Prerequisite: MATH 113 or MATH 115

Fall, Spring

CS 201W (4) Artificial Intelligence & Science Fiction

Course will explore the interplay between science fiction (1950s-present) and the development of artificial intelligence. Turing tests, agents, senses, problem solving, game playing, information retrieval, machine translation robotics, and ethical issues. Variable

WI, GE-6, GE-9

CS 209 (2) C++ for Java Programmers

C++ syntax for students who already know Java. Specific topics: data types, operators, functions, arrays, string operations, pointers, structures, classes, constructors, destructors, pointers as class members, static classes, "this" pointer, operator functions, data type conversions, inheritance, polymorphism, and dynamic binding. Prerequisite: Consent

Variable

CS 210 (4) Data Structures

Investigates efficient data structuring techniques to support a variety of operations in different problem scenarios. Topics include binary trees, binary search trees, multiway search trees, hashing and hash tables, priority queues, and algorithm analysis for best, worst and average cases.

Prerequisite: CS 111 and MATH 121

Fall, Spring

CS 220 (3) Machine Structures and Programming

This course introduces students to assembly language programming and basic machine structures. Topics include number systems; basic central processing unit (CPU) organization, instruction formats, addressing modes and their use with a variety of data structures; and parameter passing techniques.

Prerequisite: CS 110 and EE 106

Fall, Spring

CS 221 (1) Machine Structures and Programming Lab

This laboratory course complements CS 220, offering students hands-on programming experience to reinforce assembly language programming concepts. Topics include number systems; instruction formats, addressing modes and their use; and parameter passing techniques including the use of a stack frame.

Co-requisite: CS 220

Fall, Spring

CS 230 (4) Introduction to Intelligent Systems

Fundamentals of data mining and knowledge discovery. Methods include decision tree algorithms, association rule generators, neural networks, and web-based mining. Rule-based systems and intelligent agents are introduced. Students learn how to apply data-mining tools to real-world problems.

Prerequisite: CS 110

Fall

CS 271 (3) Introduction to Graphical Programming

An introduction to graphical programming environments. Topics include data and data types, repetition, selection, data acquisition, data dependency, efficiency, modular program construction, array processing, debugging, and visualization.

Prerequisite: EET 113, MATH 121

Fall, Spring

CS 293 (1) MAX Scholar Seminar

This class provides MAX scholars with an opportunity to explore a set of topics related to achieving success in academic, professional and personal realms. Speakers will include faculty, graduate students, visiting researchers and industry members as well as student participants.

Prerequisite: Recipient of a MAX scholarship or instructor consent

Fall, Spring

CS 294 (1-3) Workshop

Workshop topics will be announced. Workshops on different topics may be taken for credit.

Prerequisite: Consent of instructor

Variable

CS 295 (1) Computer Science Seminar

Provides students interested in a computer science major or minor an opportunity to explore topics not normally covered in the curriculum. Speakers will include faculty, graduate students, undergraduate students admitted to the Computer Science major, visiting researchers and industry members.

Fall, Spring

CS 296 (1-2) Introduction to Selected Topics

Special topics not covered in other 100 or 200-level courses. May be repeated for each new topic.

Variable

CS 300 (4) Large-Scale Software Development

A team-based capstone experience for the mid-point of the CS program. Students are introduced to principles and methodologies of large-scale software development and engineering by working on a full life-cycle software project solving a substantial problem using multiple CS concepts.

Prerequisite: CS 210 and CS 220

Spring

CS 305 (4) Algorithmic Structures

Study of the core algorithm design and analysis techniques of computer science and the data structures which support them with attention to the applicability to specific problem types and comparison metrics.

Prerequisite: CS 111, MATH 121

Fall

CS 310 (3) Algorithm Analysis

Algorithm design and analysis is central to much of computer science. This course exposes students to fundamental algorithm design and analysis techniques. Topics include many of the basic topic areas of computer science: searching, sorting, numeric computation, data representation, communication.

Prerequisite: CS 210

Fall

CS 315 (4) Introduction to Cryptographic Methods

An introduction to methods, algorithms, and tools of cryptography. We will study the algorithmic and mathematical aspects of cryptographic methods and protocols. We will experiment with how they can be used to solve particular data and communication security problems.

Prerequisite: CS 305 or permission of instructor.

Variable

CS 320 (3) Computer Architecture

This course presents historical and current concepts and implementations of computer organization. Topics include instruction set design, digital storage, performance metrics, processor datapath and control, pipelining, memory hierarchy, busses and I/O interfacing, and parallel processors.

Prerequisite: CS 111 and CS 220, or EE 334

Spring

CS 330 (4) Introduction to Neural Computation

This course provides an introduction to the theory and practice of neural computation. The goal is to familiarize students with the major models, techniques, and problems of neural network computation and to provide hands-on experience using these things. Topics include neural network models, supervised and unsupervised learning, associative memory models, and data representation.

Prerequisite: CS 230

Co-requisite: Permission of the Instructor

Spring

CS 340 (3) Concepts of Database Management Systems

This course covers the fundamentals of database management focusing on the relational data model. Topics include database organization, file organization, query processing, concurrency control, recovery, data integrity, optimization and view implementation.

Prerequisite: CS 210 and CS 320

Fall

CS 350 (3) Network Architectures

An introduction to data communications and networks. The field encompasses local area networks, wide area networks, and wireless communication. Topics include digital signals, transmission techniques, error detection and correction, OSI model, TCP/IP model, network topologies, network protocols, and communications hardware.

Prerequisite: CS 305 or EE 234

Spring

CS 351 (1) Network Architectures Lab

A laboratory in conjunction with CS 350.

Prerequisite: CS 305 or EE 234. Permission of instructor

Co-requisite: CS 350

Fall

CS 360 (3) Systems Programming

This course focuses on machine level I/O and operating system file processing. Structure of systems programs including assemblers, linkers, and object-oriented utilities and interfaces. Students will gain experience in writing utility programs and extensions to an operating system.

Prerequisite: CS 111 or EE 107, and CS 320

Fall

CS 361 (3) Windows Programming

This course introduces the student to Windows programming in C++ using the Application Programming Interface. Windows programs are created in a visual development environment which includes editing and code generating facilities. Hands-on programming skills are developed in the lab.

Prerequisite: CS 210

Variable

CS 365 (3) Graphics and Game Programming I

The course introduces the student to graphics and game programming. Graphics programming topics addressed include modeling, rendering, and animation of vector-based components and bitmaps. Programs are created using a current graphics and game development environment.

Prerequisite: CS 210, CS 220, MATH 121

Alt-Fall

CS 370 (3) Concepts of Programming Languages

Fundamental concepts of programming languages, including principles of language design, language constructs, and comparison of major languages. Topics: formal methods of examining syntax and semantics of languages and lexical analysis of language components and constructs, and propositional and predicate calculi.

Prerequisite: CS 210

Fall

CS 380 (3) Analysis and Design of Software Systems

Students are introduced to techniques used in analysis and design of software systems. Traditional techniques are reviewed and current methodologies for both object-oriented and procedural systems are studied. Standard notations used to document software requirements and designs are presented.

Prerequisite: CS 300

Spring

CS 400 (3) Software Design and Architecture

Current processes, methods and tools related to formal methods for modeling and designing software systems. Topics include software architectures, methodologies, model representations, component-based design, patterns, frameworks, CASE-based designs, and case studies.

Prerequisite: CS 300 and MATH 121

Variable

CS 410 (3) Formal Languages/Abstract Machines

This course studies the theoretical underpinnings of modern computer science, focusing on three main models of computation: DFA, PDA, and Turing Machines. Students determine model capabilities and limitations: what is and is not computable by each of them.

Prerequisite: CS 310 and MATH 375

Fall

CS 415 (3) High Performance Computing

High Performance Computing techniques used to address problems in computational science. Topics include application areas and basic concepts of parallel computing, hardware design of modern HPC platforms and parallel programming models, methods of measuring and characterizing serial and parallel performance.

Prerequisite: CS 310, CS 350, and MATH 247

Variable

CS 420 (3) Advanced Computer Architecture

This course addresses advanced topics in computer architecture including a major emphasis on measuring and improving computer performance. Topics include advances in pipelining and analysis and optimization of storage systems and networks, multiprocessor challenges and trends.

Prerequisite: CS 320 and MATH 375

Variable

CS 425 (3) Real-time and Embedded Systems

This course provides an overview of embedded and real-time systems including design principles, methodologies, design tools and problem solving techniques. Students design and build a real-time operation system with a microprocessor to host real-time service data processing using sensor/actuator devices.

Prerequisite: CS 210 and CS 230

Variable

CS 430 (3) Artificial Intelligence

Basic introductory concepts and a history of the field of Artificial Intelligence (AI) are covered. Emphasis is placed on the knowledge representation and reasoning strategies used for AI problem solving. Solutions are found using the LISP programming language.

Prerequisite: CS 230 or CS 305

Alt-Fall

CS 431 (3) Computational Linguistics

Computational linguistics topics covered include regular expressions, finite state automata, information theory, context free grammars, hidden Markov models and Viterbi algorithms. Students will work on problems within the field including parsing, machine translation, speech recognition, information extraction and parsing.

Prerequisite: CS 210 or CS 230

Alt-Fall

CS 433 (3) Data Mining and Machine Learning

A blend of computer science, information science, and statistics for storing, accessing, modeling, and understanding large data sets. Topics include fundamental data mining algorithms: decision trees, classification, regression, association rules, statistical models, neural networks, and support vector machines.

Prerequisite: CS 210 and STAT 354

Alt-Spring

CS 452 (3) Network Protocol Internals

As an advanced coverage of data communication, this course explores principles, protocols and performance evaluation techniques of advanced networking technologies. Topics include error detection and recovery, flow control, routing, data throughput, and performance analysis of existing and emerging Internet protocols.

Prerequisite: CS 350 and STAT 354

Variable

CS 454 (3) Mobile and Wireless Networks

Emerging mobile and wireless data networks technologies covered include standard wireless protocols (e.g., Bluetooth, IEEE 802.11, RFID, and WAP), and development of mobile and wireless applications (e.g., J2ME, WML, Brew). Includes research, design, and implementation of a wireless, mobile application.

Prerequisite: CS 320 and CS 350

Variable

CS 455 (4) Computational Geometry

This course studies the problems, methods, and algorithms of computational geometry. We will focus on the core problems and categories of the discipline: static problems, geometric query problems, and dynamic problems. Some additional attention will be given to numerical geometric problems (e.g., parametric surfaces).

Prerequisite: CS 305 and Math 247 or permission of instructor.

Fall (Even), Spring (Odd Years)

CS 460 (3) Operating Systems: Design & Implementation

This course studies historical and current concepts and implementations of computer operating systems. Basic operating systems topics include processes, interprocess communication, interprocess synchronization, deadlock, memory allocation, segmentation, paging, resource allocation, scheduling, file systems, storage, devices, protection, security, and privacy.

Prerequisite: CS 305 or EE 395

Spring

CS 461 (1) Operating Systems Lab

A laboratory in conjunction with CS 460.

Prerequisite: CS 305, EE 395. Permission of instructor

Co-requisite: CS 460

Spring

CS 465 (3) Graphics and Game Programming II

The second of a two-course sequence on graphics and game programming. The course concentrates on 3D graphics including modeling, rendering, and animation for computer games and graphic simulations. Programs are created using a current graphics and game development environment.

Prerequisite: CS 365, MATH 375

Variable

CS 470 (3) Compilers

This course offers an introduction to specification and implementation of modern compilers. Topics include lexical scanning, parsing, type checking, code generation and translation, optimization, and compile-time and run-time support for modern programming languages. Students build a working compiler.

Prerequisite: CS 370

Variable

CS 480 (3) Advanced Programming Practices

This course covers advanced programming for general-purpose software development. Topics include tools and processes appropriate for employing object-oriented designs and programming within a significant software development environment and advanced data structures and algorithms, graphical user interfaces, and software development processes.

Prerequisite: CS 300 and CS 380

Variable

CS 481 (3) Software Engineering

Building upon the introduction provided in CS 300, provides a formal presentation of software engineering concepts. Additional topics include alternative design methods, software metrics, software project management, reuse and re-engineering.

Prerequisite: CS 300, CS 380 and MATH 121

Variable

CS 482 (3) Software Verification

Provides an introduction to software quality assurance with focus on software testing processes, methods, techniques and tools. Topics include formal verification and validation techniques; black box and white box testing; integration, regression, performance, stress, and acceptance testing of software.

Prerequisite: CS 300, CS 380 and MATH 354

Variable

CS 490W (4) Senior Capstone

Students gain experience working with a team to solve a substantial problem in the field of computer science using concepts that span several topic areas in computer science. Class time focuses primarily on project design and implementation.

Prerequisite: Senior standing and successful completion of all core requirements.

Spring

WVI

CS 493 (1) MAX Scholar Seminar

This class is for MAX scholars and covers topics related to achieving success in academic, professional and personal realms. Speakers will include faculty, graduate students, visiting researchers and industry members. Students will mentor lower division scholars and do presentations.

Prerequisite: Recipient of a MAX scholarship or instructor consent

Fall, Spring

CS 494 (1-3) Workshop

Workshop topics will be announced. Workshops on different topics may be taken for credit.

Variable

Prerequisite: Consent of Instructor

CS 495 (1) Computer Science Seminar

Provides Computer Science majors or minors an opportunity to explore topics not normally covered in the curriculum. Speakers will include faculty, graduate students, undergraduate students admitted to the Computer Science major, visiting researchers and industry members. This class may be repeated for credit.

Prerequisite: Admitted to major

Fall, Spring

CS 496 (1-4) Selected Topics in Computer Science

Special topics not covered in other courses. May be repeated for credit on each new topic.

Prerequisite: Consent

Variable

CS 497 (1-6) Internship

This course is designed to provide students with an opportunity to utilize their training in a real-world environment. Participants work under the guidance and direction of a full-time staff member. (At most 4 hours towards the CS major.)

Prerequisite: Permanent admission to the CS major, CS 300, consent.

CS 498W (4) Senior Thesis

Advanced study and research required. Topic of the senior thesis determined jointly by the student and the faculty advisor.

Prerequisite: Senior standing and consent

Fall, Spring

WI

CS 499 (1-4) Individual Study

Problems in the field of computer science are studied on an individual basis under the guidance of a faculty mentor.

Prerequisite: Consent

Fall, Spring