

Computer Science Department (CS)

College of Science and Health

Department Chair: Steven Senger

221 Wing Technology Center; 608.785.6805

Email: ssenger@uwlax.edu

Department Email: compsci@uwlax.edu

www.cs.uwlax.edu

You don't have to be an expert with computers to be a computer science major! However, if you are a logical thinker and like to solve problems this may be the major for you. The curriculum for a computer science major at UW-La Crosse is both exciting and demanding. What will you do with the knowledge you've learned? You will become a world class problem solver. After learning the fundamental material in computer software, hardware and the theory of computing, you will be finding ways to make life better for people. Software is an integral part of our society in both visible and invisible ways. Take a look at some of the modern conveniences of our life; iPods, cell phones, video games, medical imaging systems, and engines. As you can see, the way we interact with information through software leaves very little that computer science does not touch.

With that said, we think there is no better place to pursue your degree than UW-La Crosse. Small class sizes and accessible professors give students a level of attention not available on larger campuses. We have outstanding faculty who have many different interests, but who all share a love for teaching. A history of talented students, great facilities and undergraduate research opportunities, make for an ideal learning environment. The Department of Computer Science at UW-La Crosse is the second oldest in Wisconsin, only behind Madison, and has a fine reputation within the computing industry during the past 48 years.

The job market for computer scientists has always had plenty of opportunities and the demand continues. The National Bureau of Labor Statistics, in its rolling ten-year horizon Occupational Outlook Handbook, consistently projects the software industry to be among the fastest growing sectors of the economy.

If you would like to find out more about computing, check out the Association for Computing Machinery's (ACM) "Computing, Degrees and Careers. (<http://computingcareers.acm.org>)"

Computer Science retroactive credit policy

Students taking CS 220 Software Design II (4 cr.) and earning a "B" or better, will be given retroactive credit for CS 120 Software Design I (4 cr.) provided that the student's transcript shows no record of prior or concurrent enrollment in CS 120 or CS 220.

Computer Science internship policy

Students majoring in computer science who wish to enroll in CEI 450 Cooperative Education/Internship (1-15 cr.) or CEI 475 University-wide Cooperative Education Internship (3-15 cr.), University Internships, must have completed 15 credits of computer science course work, including CS 220, and must have a minimum GPA of 2.50 in computer science courses to be eligible for departmental approval and supervision. Credits earned in internships do not apply to major or minor requirements.

General education writing emphasis

This department incorporates a significant amount of writing through the required courses instead of identifying particular courses as writing emphasis courses. Students who complete a major in this department will fulfill the general education writing emphasis requirement (<http://catalog.uwlax.edu/undergraduate/generaleducation/#generaleducationrequirementstext>).

2017-18 Faculty/Staff

The following is the department's faculty and staff as of the publication date of this catalog. This list will not be updated again until the next catalog is published in June.

Professor

Kenny Hunt

Kasilingam Periyasamy

Steven Senger

Mao Zheng

Associate Professor

Martin Allen

Thomas Gendreau

Assistant Professor

Samantha Foley

Elliott Forbes

John Maraist

Michael Petullo

Allison Sauppé

Jason Sauppe

Brad Shutters

Administrative Support

Becky Yoshizumi

Majors

- Computer science major - BS (<http://catalog.uwlax.edu/undergraduate/computerscience/computersciencebs>)
- Computer science major - computer engineering technology emphasis - BS (<http://catalog.uwlax.edu/undergraduate/computerscience/computersciencemajorcomputerengineeringtechnologyemphasis>)
- Computer science education (early adolescence-adolescence certification) - BS (<http://catalog.uwlax.edu/undergraduate/computerscience/computerscienceeducation>)
- Dual degree program in computer science and engineering (<http://catalog.uwlax.edu/undergraduate/computerscience/dualdegreeprogramincomputerscienceandengineering>)

- Dual degree program in computer science and master of software engineering (<http://catalog.uwlax.edu/undergraduate/computerscience/dualdegreeprogramincomputerscienceandmasterofsoftwareengineering>)

Minors

- Computer science minor (<http://catalog.uwlax.edu/undergraduate/computerscience/computerscienceminor>)
- Computer science education minor (<http://catalog.uwlax.edu/undergraduate/computerscience/computerscienceeducationminor>)

+ next to a course number indicates a **general education** course

Computer Science Courses

+CS 101 Cr.4

Introduction to Computing

Computers and computer software are an integral part of modern society. This course explores this relationship. Students will examine the computer as a problem-solving tool through the use of database, spreadsheets and small scale programming. Students will examine the computer as a communication tool through the use of word processing and the Internet. Other topics include the history and future of computer technology, computer hardware basics, man/machine relationships, applications of computers in various disciplines, and social/ethical issues. Credits earned in CS 101 cannot be applied to the CS major or minor. Offered Occasionally.

CS 103 Cr.1

Elementary Database Principles and Design

An introduction to the design and implementation of relational databases. Design concepts will include entity-relationship modeling, relational table structure, keys, foreign keys, referential integrity, and data quality. Implementation concepts will emphasize extracting information through queries, reports, and forms. Seven-week course. Prerequisite: course not open to those who have completed CS 101. Offered Occasionally.

CS 104 Cr.1

Elementary Spreadsheet Principles and Design

An introduction to the design and implementation of spreadsheets. Design principles will include cell content, use of functions and formulas, relative and absolute addressing, and formatting. Modern spreadsheet software will be used to implement the spreadsheets, with an emphasis on presenting numeric data in an organized manner. Seven-week course. Offered Occasionally.

+CS 120 Cr.4

Software Design I

An introduction to the fundamentals of software development; including software classes, objects, inheritance, polymorphism, logic, selection control, repetition control, subprograms, parameter passage, and rudimentary software engineering techniques. Students complete numerous programming projects using a modern programming language. Prerequisite: MTH 151 or MTH 175 or math placement test scores at or above MTH 151. Offered Fall, Spring.

CS 202 Cr.3

Introduction to Web Design

This course is an introduction to webpage design and website management. Students will be introduced to browser/server interaction, webpage design, techniques for creating media rich graphical interfaces, and ethical considerations regarding intellectual property rights and security. Students will receive hands-on experience in a variety of standardized web technologies to develop dynamic, functional, and appealing webpages. Prerequisite: CT 100 or CS 120 or familiarity with some programming language. Offered Occasionally.

CS 220 Cr.4

Software Design II

This is a second course in the design of programs. Emphasis is placed on data abstraction and its application in design. Definitions of abstract data types are examined. The following structures are examined as methods for implementing data abstractions: recursion, sets, stacks, queues, strings, and various linked lists. Students will be expected to write several programs using these techniques in a modern programming language. Prerequisite: CS 120; concurrent enrollment in CS 225 recommended. Offered Fall, Spring.

CS 224 Cr.1-3

Introduction to Programming Language

This course presents the syntax and semantics of a particular programming language. Different offerings of the course will present different languages. Students are expected to be fluent in another programming language prior to enrollment. Repeatable for credit - maximum six. (Each repeat must be for a different language.) Prerequisite: a prior computer science course appropriate to the language being offered. Offered Occasionally.

CS 225 Cr.3

Discrete Computational Structures

An introduction to mathematical reasoning and proof with an emphasis on its role in computation and software development. Course topics include mathematical logic, set theory, relations and functions, induction and recursion, equivalence relations, partial orders, lattices and other algebraic structures. Prerequisite: CS 120; MTH 151; concurrent enrollment in CS 220; declared computer science major or minor. Course not open to those who have completed MTH 225. Offered Fall, Spring.

CS 227 Cr.1

Competitive Programming

Programming competitions involve writing code to solve a set of problems, working under a time-limit. Learning to participate successfully in such competitions is an effective way to become a better programmer. This course introduces students to some of the techniques and problem types that are commonly encountered in many programming contests. Students will learn a set of common algorithms and data structures, along with general conceptual tools and rules of thumb, that will prepare them to be able to solve programming problems encountered in competition and job interview settings. The course will consist of code review, along with extensive coding practice. Even experienced programmers will have something to learn from the experience of solving new problems from scratch on a regular basis. Repeatable for credit - maximum three. Prerequisite: CS 220. Concurrent enrollment or completion of CS 340 recommended. Offered Fall, Spring.

CS 270 Cr.3

Introduction to Assembler Programming, C Programming and Computer Organization

This course involves programming in machine, assembler and C languages and an introduction to computer organization. Machine language, assembler language and computer organization topics include basic instruction types, data representation, addressing modes, registers, the ALU, computer memory, and interrupt handling. C languages topics include control structures, data types, arrays, structs, files, pointers, dynamic memory, compiling and linking. Prerequisite: CS 120; CS 225 or MTH 225. Offered Fall, Spring.

CS 272 Cr.3

Digital Circuit Design for Microcontrollers I

An introduction to digital circuit design in the context of embedded microcontroller devices. This will include flip-flops, sequential and combinational logic units (e.g. shift registers, half-adders, encoders * decoders, multiplexers), state machines, analog to digital and digital to analog conversion. Prerequisite: CS 225; MTH 207. Offered Annually.

CS 340 Cr.4

Software Design III: Abstract Data Types

An extensive survey of data structures and associated algorithms. An introduction to algorithm efficiency measures is included as a tool for deciding among alternate algorithms. Topics include: searching and sorting in arrays, hash tables, tree traversal and search algorithms, expression evaluation, functional programming, development of thread-safe data structures and graphs. Prerequisite: CS 220; CS 225 or MTH 225. Offered Fall, Spring.

CS 341 Cr.3

Software Design IV: Software Engineering

A study of methodologies for the development of reliable software systems. Several specification, design, and testing techniques are surveyed with an emphasis on one particular formal specification and formal design technique. Students work in teams, applying these techniques to the development of a medium scale (2000-5000 lines) software product. Prerequisite: CS 340. Offered Fall, Spring.

CS 342 Cr.3

Software Testing Techniques

As the size and complexity of software projects have grown, so has the importance of ensuring program correctness. This course examines the issues of program testing, validation, and verification. Course projects require students to construct test data and to analyze the correctness of several software systems. Prerequisite: CS 340. Offered Spring - Odd Numbered Years.

CS 351 Cr.3

Simulation

Programming computers to play games and imitate activities of systems such as drive-in facilities, checkout lanes, and computer operations. Topics include tests of goodness of fit, random number generators, simulated sampling, queuing theory, analysis of systems to be simulated, construction and validation of simulation programs, and interpretation of results. Prerequisite: CS 220; MTH 207. Offered Occasionally.

CS 353 Cr.3

Analysis of Algorithm Complexity

An in-depth analysis of the computational complexity of a wide range of algorithms for problems of fundamental importance to computer science. Algorithms to be examined include: sorting, pattern matching and various graph algorithms. Prerequisite: CS 340; MTH 207. Offered Fall - Even Numbered Years.

CS 364 Cr.3

Introduction to Database Management Systems

Introduction to the design and organization of database management systems. Topics include the relational data model, relational algebra, SQL query language, database software development, data security, normalization, client/server environments. Prerequisite: CS 220. Offered Fall, Spring.

CS 370 Cr.3

Computer Architecture

A presentation of the logical organization of modern digital computers. Topics include performance evaluation, instruction set design, computer arithmetic, processor control, pipelining, cache memory, memory hierarchy, memory and system buses, and I/O organization. Prerequisite: CS 270. Offered Fall, Spring.

CS 372 Cr.3

Digital Circuit Design for Microcontrollers II

This is the second course in digital circuit design for microcontrollers. This will include inter-component communication using PWM, I2C, SPI and UART. The course will also include the design and fabrication of printed circuit boards (PCBs), rigid and flexible, for surface mount devices. Prerequisite: CS 272. Offered Annually.

CS 395 Cr.1-3

Independent Study

Individualized study of topics in computer science not covered by courses regularly taught in the department. Repeatable for credit - maximum six. Consent of department. Offered Fall, Spring, Summer.

CS 402/502 Cr.3

Web Application Development

This course will give a detailed description of the core concepts and general principles of web application development. The course will cover various protocols, programming languages, scripting languages, data storage and security, layered software architectures, and graphical interface design as they relate to web development. Students will apply these techniques to the development of medium scale web application. Prerequisite: CS 202; CS 340; junior standing. Offered Fall - Odd Numbered Years.

CS 410/510 Cr.3

Free and Open Source Software Development

This course examines all aspects of the Free and Open Source Software movement. The course surveys the various definitions of open source licenses and examples of major free and open source development projects (e.g. the GNU Project, Apache Foundation, Linux). The course also examines the development tools that support developer communities, as well as how web-based applications have created the possibility of international development teams. Students will select and contribute to the software development of an existing open source project. Prerequisite: CS 340; junior standing. Offered Spring - Odd Numbered Years.

CS 418/518 Cr.3

Mobile Application Development

An introduction to the concepts and techniques of application development for mobile devices. The course will examine the design constraints of mobile devices, how mobile applications can leverage external data resources, integration of sensor data and the development environments of the chosen platform (e.g. iOS, Android and others). Repeatable for credit with different topic - maximum six. Prerequisite: CS 340; junior standing. Offered Occasionally.

CS 419/519 Cr.1-3

Topics in Computer Science

A special topics course in Computer Science which will function as a forum for new ideas and testing ground for new courses. Repeatable for credit - maximum six. Prerequisite: junior standing. Consent of instructor. Offered Occasionally.

CS 421/521 Cr.3

Programming Language Concepts

A comparative study of the concepts underlying the design of contemporary high-level programming languages, including imperative, functional, logic and object-oriented paradigms; formal representation of syntax and semantics; control structures; data and procedural abstraction; scope and extent; parallelism and exception handling. Prerequisite: CS 340; CS 225 or MTH 225; junior standing. Offered Fall, Spring.

CS 431/531 Cr.3

Introduction to Robotics

This course is a hands-on introduction to the algorithms and techniques required to write robot control software. Topics include the components of mobile robots and robot manipulators, manipulator kinematics, robot task planning, sensing, sensor fusion, visual servoing and robot control concepts. Prerequisite: CS 340; junior standing. Offered Spring - Odd Numbered Years.

CS 441/541 Cr.3

Operating System Concepts

The study of the structures and algorithms of operating systems. Operating systems are viewed as managers and controllers of resources such as processors, memory, input and output devices and data. Topics include multiprogramming systems, CPU scheduling, memory management and device management. Prerequisite: CS 340; CS 370; junior standing. Offered Fall, Spring.

CS 442/542 Cr.3

Structures of Compilers

An extensive study of all phases of the compilation of high level programming languages. Topics include: scanning, parsing (LL and LR), semantics analysis, symbol table organization and manipulation, internal code generation, storage allocation, optimization and object code generation. Students are required to complete a compiler for a small high-level language. Prerequisite: CS 270; CS 340; junior standing. Offered Spring.

CS 443/543 Cr.3

Topics in Operating Systems

An intermediate course in operating systems extending topics introduced in CS 441. Operating systems concepts are studied in depth. Typically students will study and modify an existing system. Prerequisite: CS 441; junior standing. Offered Fall - Even Numbered Years.

CS 446/546 Cr.3

Object-Oriented Software Development

Introduction to the concepts and principles of object-orientation (OO). Topics include detailed discussion on analysis and design of OO software systems, notations for OO analysis and design, and comparison of OO programming languages. Advanced topics on object-orientation such as OO testing and software reuse will be briefly discussed. Prerequisite: CS 340; junior standing. Offered Spring.

CS 449/549 Cr.3

Advances in Software Engineering

Introduces advanced topics in software engineering. Topics include prototyping models, risk analysis, component-oriented software development, software architectures, software reuse, software metrics and quality analysis. Prerequisite: CS 341; junior standing. Offered Fall - Even Numbered Years.

CS 451/551 Cr.3

User Interface Design

This course focuses on the design and implementation of user interfaces. The topics include characteristics of user interfaces, user profiles, user interface design principles, methods and tools for user interface development, evolution of user interfaces, evaluation of user interfaces, and case studies. Prerequisite: CS 340; junior standing. Offered Fall - Odd Numbered Years.

CS 452/552 Cr.3

Artificial Intelligence and Pattern Recognition

An introduction to the fundamental principles of artificial intelligence. Topics include the biological basis for intelligence, classification of object descriptions and pattern recognition, search strategies and game trees, natural language processing, automatic theorem proving, programming for artificial intelligence and knowledge-based systems. Projects include writing a substantial artificial intelligence application program. Prerequisite: CS 340; junior standing. Offered Fall - Odd Numbered Years.

CS 453/553 Cr.3

Introduction to Theory of Computation

An introduction to the theoretical aspects of computation. The capabilities and limits of several computation models are considered including: partial recursive functions, Turing machines, finite state automata and formal languages. The implications of Church's thesis and unsolvable problems such as the halting problem are discussed. Prerequisite: CS 340; junior standing. Offered Spring - Even Numbered Years.

CS 454/554 Cr.3

Digital Image Processing

This course introduces the fundamentals of digital image processing techniques with an emphasis on the design and implementation of image processing algorithms. Topics include color models, point-processing techniques, convolution, Fourier domain processing, the discrete cosine transform, image compression methodologies, image restoration and enhancement, sampling and image display. Prerequisite: CS 340; junior standing. Offered Fall - Even Numbered Years.

CS 455/555 Cr.3

Fundamentals of Information Security

This course presents the fundamental concepts of information security. Basic policies, techniques and tools for maintaining the security of host computers, information networks and computer software are presented. Elementary cryptography is explored with special attention to applications in data encryption, hashing and digital signatures. Fundamental security management procedures also are introduced, as are the legal and ethical issues associated with computer security. Students will be expected to apply the knowledge gained to construct security policies and practice security in the form of access privileges, firewalls, vulnerability scanners and intrusion detection tools. Prerequisite: CS 220; junior standing. Offered Fall - Odd Numbered Years.

CS 456/556 Cr.3

Secure Software Development

Traditionally, software engineering has viewed flaws as the inconsistency of software behavior with its functional requirements. Software security problems, however, can occur in software that contains no such flaws but is nonetheless susceptible to external attack. This course examines known reasons for software security vulnerabilities with an emphasis on best practices for their detection and mitigation, along with general principles for engineering software in ways that enhance security. Prerequisite: CS 340; junior standing. Offered Spring - Even Numbered Years.

CS 464/564 Cr.3

Advanced Database Management Systems

Advanced topics in database management systems. Topics include the relational data model, relational calculus, embedded SQL programming, database application programming, indexing, systems software and storage structures for databases, concurrency control, crash recovery, database administration, parallel and distributed databases, object oriented databases. Prerequisite: CS 364; junior standing. Offered Spring - Odd Numbered Years.

CS 470/570 Cr.3

Parallel and Distributed Computing

A study of architectures, control software, and applications for parallel and distributed systems. A survey of parallel and distributed architectures including data flow machines, vector processors, shared memory multiprocessors, and message based multiprocessors. Software topics include process communication and synchronization, global state maintenance, negotiation, scheduling, data parallelism, control parallelism, and languages for parallel and distributed computing. Prerequisite: CS 370; junior standing. Offered Occasionally.

CS 471/571 Cr.3

Data Communications

An introduction to data communications, including the electrical properties and software protocols. In addition to presentations of the concepts and techniques used for data communications, several currently used standards and communications networks will be examined. Prerequisite: CS 270; CS 340; junior standing. Offered Spring - Even Numbered Years.

CS 472/572 Cr.3

Internet of Things

This course explores the possibilities which are created when everyday things become connected to the internet and how this can create new ways for humans to interact with computation and for computation to enable human activities. This course involves building small, sensor equipped hardware devices and cloud based software systems using various technologies. Prerequisite: CS 340, CS 372. Offered Annually.

CS 475/575 Cr.3

Computer Graphics and Modeling

An introduction to computer graphics in modern computing environments. Topics include geometric transformations, fundamental drawing algorithms, scalable vector graphics (SVG), OpenGL, WebGL, surface shaders, scene graphics, photorealistic rendering, surface mesh data structures, animation and modeling and GPGPU computing. Prerequisite: CS 340; MTH 207. Offered Fall - Odd Numbered Years.

CS 476/576 Cr.3

Data Visualization

An introduction to visualizing various forms of data (abstract and concrete) using computer graphics. The course will consider both scientific visualization where the data itself determines the spatial representation and information visualization where appropriate spatial representations are imposed on the data. Prerequisite: CS 475. Offered Spring - Even Numbered Years.

CS 499 Cr.1-3

Research in Computer Science

An opportunity to become acquainted with literature in the field and to work on a professional level research project within an area of interest of the computer science faculty. A seminar reviewing the results of the study will be a requirement for completion of the course. Repeatable for credit - maximum six. Consent of instructor. Offered Fall, Spring.

Computational Thinking Courses

+CT 100 Cr.3

Introduction to Computational Thinking

Computational thinking represents a universally applicable collection of concepts and techniques borrowed from computer scientists. This course is designed to teach how to think algorithmically; to examine the ways that the world's information is encoded and how this impacts our lives; to explore the capabilities and limitations of computers from the past, the present and the future; to apply software design diagrammatic techniques to model real-world systems; to learn how the rules of logic apply to computation, reasoning and discourse; to examine how computers both enhance and constrain our lives; to explore many of the problem solving strategies used by software developers and how they are useful to you. Offered Fall, Spring.