Computer Science & Computer Engineering Department (CS/CPE/CT)

College of Science and Health
Department Chair: Kenny Hunt
221 Wing Technology Center; 608.785.6805
Email: khunt@uwlax.edu
Department Email: compsci@uwlax.edu

www.cs.uwlax.edu (http://www.cs.uwlax.edu/)

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: cell phones, video games, medical imaging systems, and engines. The way we interact with information through software leaves very little that computer science does not touch. In computer science, students will learn the foundational material for software development, computer systems, and theory of computation so they, too, can harness the power of computers to solve the world's problems. This creative discipline is for anyone who wishes to work hard and use logic, software, and hardware to solve problems.

We think there is no better place to pursue your degree than UWL. 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 is the second oldest in Wisconsin, only behind Madison, and has a fine reputation within the computing industry for over 50 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.

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).

2023-24 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 July.

Professor

Kenny Hunt

Mao Zheng

Associate Professor

Samantha Foley

Allison Sauppe

Assistant Professor

Elliott Forbes

Dipankar Mitra

W. Michael Petullo

Jason Sauppe

Lei Wang

Administrative Support

Tonya Peterson

Majors

- Computer engineering major - BS (http://catalog.uwlax.edu/undergraduate/computerscience/computerengineeringbs/)
- 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 major: cybersecurity emphasis - BS (http://catalog.uwlax.edu/undergraduate/computerscience/computersciencemajorcybersecurityemphasis/)

Dual degree programs

- Dual degree program in computer science and master of software engineering (http://catalog.uwlax.edu/undergraduate/computerscience/dualdegreeprogramincomputerscienceandmasterofsoftwareengineering/)
- Dual degree program in computer science: cybersecurity emphasis and master of software engineering (http://catalog.uwlax.edu/undergraduate/computerscience/dualdegreecscybersecurity-bs-mse/)
- Dual degree program in computer science and engineering (http://catalog.uwlax.edu/undergraduate/computerscience/dualdegreeprogramincomputerscienceandengineering/)

Minors

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

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.

Computer Engineering Courses

CPE 105 Cr.1
Introduction to the Computing Environment
This course introduces students to the use of the UNIX/Linux environment for file and process management, programming workflow, and the automation of computing tasks. The ethical responsibilities of shared computing resources are emphasized. Offered Fall.

CPE 212 Cr.3
Digital Logic
This course is an introduction to the fundamentals of digital logic circuit analysis and design. Basic Boolean logic primitives are introduced and described in truth tables, schematics and Boolean expressions. Combinational logic circuits are minimized with DeMorgan's Law and Karnaugh Maps. Level-sensitive and edge-triggered sequential logic elements are used as building blocks for finite state machines. Circuits are simulated using a structural hardware description language. Prerequisite: CS 120; concurrent enrollment in CS 270. Offered Spring.

CPE 301 Cr.3
Introduction to Transient Analysis
This course introduces students to fundamentals of electrical circuit response over a period of time defined by the user and various analysis techniques for the response of first and second order electric circuits. The topics include time domain analysis of dynamic linear (first and second order) circuits, introduction to Laplace transformations, the use of Laplace transform in circuit analysis, and introduction to Fourier series analysis and its applications in circuits. Prerequisite: MTH 308; PHY 334. Offered Fall.

CPE 302 Cr.3
Introduction to Control Systems
This course introduces students to control systems, the study of the analysis and regulation of the output behavior of dynamic systems subject to input signals, including the modeling, analysis, and design of linear feedback control systems. The emphasis of this course is on the basic theories and feedback controller design methods of linear time-invariant systems. Prerequisite: MTH 308; PHY 334. Offered Occasionally.

CPE 309 Cr.3
Systems Development
This course is an introduction to systems programming and the UNIX/Linux user-space interface to the operating system. Low-level C programming constructs are discussed and used to write efficient and robust systems code. The various tools used in file inspection, systems development and maintaining a portable build environment are also examined. Prerequisite: CPE 105; CS 270. Offered Fall.

CPE 321 Cr.3
Introduction to Digital Signal Processing
This course is an introduction to the fundamentals of digital signals and systems. Various representations for discrete time signals will be introduced. Students will learn sampling theory, aliasing and reconstruction. The Discrete Fourier Transform will be used to determine the frequency content of signals. Linear time-invariance will be introduced and used as a basis for developing complex systems. Prerequisite: MTH 308; PHY 334. Offered Spring.

CPE 395 Cr.1-3
Independent Study
Individualized study of topics in Computer Engineering not covered by courses regularly taught in the department. Repeatable for credit - maximum six. Prerequisite: junior standing. Offered Fall, Spring, Summer.

CPE 406 Cr.3
Architecture of Parallel Systems
This course covers the fundamental architectural requirements of parallel computer systems. Students will study the programming models used in parallel code and the assumptions made by parallel programmers for various programming models. The course then explores the underlying architectural decisions that can be made and how they either support or violate the assumptions of these parallel programming models. Topics include cache coherence, cache consistency, and processor interconnect. Prerequisite: CS 441. Offered Occasionally.

CPE 419 Cr.1-3
Topics in Computer Engineering
A special topics course in computer engineering that will function as a forum for new ideas and testing ground for new courses. Repeatable for credit - maximum six. Prerequisite: junior standing. Offered Occasionally.

CPE 420 Cr.3
Digital Design
This course covers the design and implementation of large, complex digital systems. Students will describe designs with fully-synthesizable, behavioral Verilog. The efficiency of designs will be analyzed for bottlenecks that can be solved with architectural and/or implementation optimizations. Emphasis will be placed on the test of systems. Functional testing will include the use of simulation test benches with calls to high-level languages. Prerequisite: CPE 212; CS 370. Offered Occasionally.

CPE 446 Cr.3
ASIC Design
This is a course in digital ASIC design, including a deeper insight into the design of high-performance and power-efficient CMOS circuits. The parasitics inherent to MOSFETs are used as a basis for transistor sizing. Techniques for estimating wire loads are discussed. CMOS circuits will implement cells of arbitrary digital logic functions for both static and dynamic logic. Prerequisite: CPE 212; CS 370; PHY 335. Offered Occasionally.

CPE 463 Cr.3
Advanced Computer Architecture
In this course in modern computer architecture, students will study advanced techniques for extracting instruction-level parallelism from single-threaded programs. The in-order superscalar execution model will be introduced, and then extended to include the out-of-order execution model in two forms - Tomasulo's algorithm and the MIPS R10k architecture. Students will be introduced to branch prediction and prefetching as mechanisms for alleviating bottlenecks in all processor models. Prerequisite: CS 370. Offered Occasionally.

CPE 466 Cr.3
Code Generation and Optimization
This course studies the algorithms used by a modern optimizing compiler for generating efficient, high-performance program executables that still maintain correct program semantics. The course uses the compiler intermediate representation as a starting point for a variety of code transformations necessary for local and global optimizations, profile-guided optimizations, constructing large optimization regions, register allocation, and instruction scheduling. Prerequisite: CS 370. Offered Occasionally.

CPE 478 Cr.3
Virtual Machines
This course explores the design of virtual machines and their related systems. Students will study efficient emulation of user-level programs, both within the same instruction set as the host machine, as well as across instruction sets. System-level considerations will be introduced to expand the reach of possible virtualization strategies. Both hardware and software techniques for efficient virtualization will be employed. Prerequisite: CPE 309; CS 441. Offered Spring.

CPE 481 Cr.1
Professionalism in Engineering
This course surveys issues related to the professional responsibility in engineering careers. Guidance in ethical decision-making and where to find assistance in navigating ethical dilemma are discussed. The importance of professional societies and their codes of conduct are emphasized. Students will recognize the importance of, and strategies to engage in, life-long learning within their field. Prerequisite: junior standing. Offered Fall.

CPE 483 Cr.1
Engineering Project Management
This course is an introduction to the issues relevant to managing engineering projects. Topics include effective group organization, decision making, time and cost estimation, progress tracking, defect tracking, conflict resolution and leadership. Additional topics may include case studies in management strategies, tools for managing products and projects, and process improvement techniques. Prerequisite: STAT 245; junior standing. Offered Spring.

CPE 498 Cr.2
Senior Capstone
This course is a major computer engineering group project that requires a detailed analysis of the problem domain, organization into groups, effective management, detailed design, implementation and demonstration. The project will be guided by a department faculty member with interests in computer engineering. Submission of a written project report is required, followed by an oral examination by the Project Evaluation Committee in the department. Repeatable for credit - maximum four. Prerequisite: senior standing. Pass/Fail grading. Offered Fall, Spring.

CPE 499 Cr.1-3
Research in Computer Engineering
This course is an opportunity to become acquainted with literature in the computer engineering field and to work on a professional level research project within an area of interest of the computer engineering faculty. A seminar reviewing the results of the study will be a requirement for completion of the course. Repeatable for credit - maximum six. Prerequisite: junior standing. Consent of instructor. Offered Fall, Spring, Summer.

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 115 Cr.3
Introduction to Python Programming
This course is an introduction to the fundamentals of software development using Python for students with no prior programming knowledge. Topics include variables, data structures, arithmetic and logical operators, selection control, repetition control, methods, parameter passing, basic data analysis, and rudimentary software engineering techniques. Topics on basic data analysis include relevant tools and Application Programming Interfaces (APIs) for working with and visualizing small to medium-sized data sets from files and online sources. Students complete numerous programming projects using the Python language. Prerequisite: not open to students with credit in CS 120. Offered Spring.

+CS 120 Cr.4
Software Design I
This course is 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 160 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 Fall.

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, generics, 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
This course is 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: grade of "C" or better in 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, CS 364. 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
This course explores how complex systems can be modeled and simulated using computer programs with an emphasis on models that are discrete, dynamic, and stochastic. Students will formulate models using both event-oriented and process-oriented modeling paradigms, implement the models in code, run computational experiments to gather output, and analyze and interpret the results. Topics include Monte Carlo simulation, discrete-event simulation, probability and random variables, pseudorandom number generation, and output analysis. 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 356 Cr.3
Software Exploitation
This course examines techniques for exploiting vulnerable software. Topics include binary reverse engineering, source code analysis, intrusion, and exploitation. The course will also discuss matters of reconnaissance, privilege escalation, lateral movement, obfuscation, and exfiltration. Students are expected to write low-level exploits using modern tools and deploy them against vulnerable services in a laboratory environment. Prerequisite: CS 270, CS 340. Offered Annually.

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
Hardware/Software Integration
This is a course in hardware and software interfacing for microcontrollers. This course includes inter-component communication using I2C, SPI and UART. Practical issues are discussed including signal integrity, debounce, level-shifting, and clock and power distribution. Lect. 2, Lab 2. Prerequisite: CS 272; or CPE 212, PHY 334, and PHY 335. 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. Prerequisite: CS 202; CS 340; junior standing. Offered Spring.

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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. 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). This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/ expectations. 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. Prerequisite: CS 270; CS 340; junior standing. Offered Fall, 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. Prerequisite: CS 441; junior standing. Offered Occasionally.

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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/ expectations. 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. Prerequisite: CS 340; junior standing. Offered Fall - Odd Numbered Years.

CS 452/552 Cr.3
Artificial Intelligence
This course is an introduction to the fundamental principles of artificial intelligence. Topics include search strategies, adversarial search, constraint satisfaction, planning and scheduling, logic and inference, accounting for uncertainty, and probabilistic reasoning. Projects include writing a substantial artificial intelligence application program. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. 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. Topics include encryption, authentication, access control, types of attacks and mitigations, software security, network security protocols, and the concepts of trust, privacy and ethics. Students are expected to compare security policies and techniques, apply concepts using modern tools and techniques, and explore recent security events. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. Prerequisite: CS 270, CS 340; junior standing. Offered Spring.

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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. Prerequisite: CS 356; junior standing. Offered Occasionally.

CS 457/557 Cr.3
Machine Learning
This course is an investigation of programs that can dynamically adapt their behavior. The course focuses on two main ideas: data classification and deciding about actions. In both cases, a learning algorithm is one that improves performance, either by generating a more accurate classifier, or by finding a choice of action that leads to better outcomes. Students will learn various computational and mathematical models and techniques that can be applied to such problems. Topics include regression algorithms, decision trees, Markov processes, neural networks, reinforcement learning algorithms, and deep learning techniques. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. Prerequisite: CS 340; junior standing. Offered Fall - Even Numbered Years.

CS 461/561 Cr.3
Introduction to Data Science
This course examines key components of the data science lifecycle, including data collection and cleaning, exploratory data analysis and visualization, and extracting insight via statistical models and machine learning algorithms for regression, classification, and clustering. Algorithmic efficiency and scalability will be emphasized, and techniques for working with big data will be introduced. Students will use a modern programming language (e.g., R, Python) with appropriate packages suitable for data analysis. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. Prerequisite: CS 340; STAT 245 or CS 351; junior standing. Offered Alternate 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, system software and storage structures for databases, concurrency control, crash recovery, database administration, parallel and distributed databases, object-oriented databases. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. Prerequisite: CS 370; junior standing. Offered Occasionally.

CS 471/571 Cr.3
Computer Networks
This course is 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. Prerequisite: CS 340, CS 372; junior standing. 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. Prerequisite: CS 340; MTH 207; junior standing. 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. This course is taught largely at an undergraduate level. Graduate students will have additional course requirements/expectations. Prerequisite: CS 340; MTH 207; junior standing. 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.