CS - Software Engineering Graduate Program

Master of Software Engineering (MSE) Program
Director: Mao Zheng
217 Wing Technology Center; 608.785.6808
Email: mzheng@uwlax.edu
Department: 221 Wing Technology Center, 608-785-6805
Dept email: compsci@uwlax.edu

www.uwlax.edu/academics/department/computer-science-and-computer-engineering/ (https://www.uwlax.edu/academics/department/computer-science-and-computer-engineering/)

The focus of the Master of Software Engineering (MSE) Program is to teach the advanced state-of-the-art technologies in software development with hands-on experience and to apply the knowledge to some challenging real-world problems. The program will guide the students to acquire both technical skills and software project management skills that are required to lead and to carry out software development projects.

Program length

The Master of Software Engineering (MSE) Program is typically a two-year program. The program length is based on how long the required UWL coursework would take to complete for a full-time student who does not need to complete any prerequisite coursework. Program length may be extended if students attend part-time (if approved by program) or due to the requirements of an individual student's plan of coursework, research or capstone project.

2024-25 Faculty/Staff

The following is the graduate 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 Sauppé

Assistant Professor

J. Elliott Forbes

Dipankar Mitra

W. Michael Petullo

Jason Sauppé

Lei Wang

Administrative Support

Sallie Coron

Graduate degree

- Software engineering - MSE (http://catalog.uwlax.edu/graduate/programrequirements/softwareengineering/mse/)

Courses

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. Prerequisites: CS 340. Offered Spring.

CS 503 Cr.1-3
Special Topics in Computer Science for Teachers
A special topics course used to introduce K-12 teachers to computer science content and to curricula and pedagogy designed for K-12 students. Not applicable to the Computer Science Program or Master of Software Engineering degree. Prerequisite: current K-12 teacher certification (any discipline). Consent of instructor. Offered Occasionally.

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. 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. 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. Consent of instructor. Offered Fall, Spring, Summer.

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. 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. 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. 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. 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. 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 741. 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. 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. 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. 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. 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 340. 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 have additional course requirements/expectations. Prerequisite: CS 356. Offered Spring.

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. 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. 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. This course cannot be taken for credit both at the undergraduate level and at the graduate level. 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. 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. 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. 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. 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 575. Offered Spring - Even Numbered Years.

CS 741 Cr.3
Software Engineering Principles
This course introduces fundamentals of software engineering and various life cycle models for software development. It focuses on software processes addressing various life cycle activities such as requirements engineering, design, implementation, testing, and maintenance. Object-oriented design using the Unified Modeling Language (UML) will be introduced. Application of software engineering methods to different application domains will be briefly discussed. Prerequisite: CS 340. Offered Fall.

CS 743 Cr.3
Software Verification and Validation
This course explains the need for verification and validation, discusses the methods (formal, informal and diagrammatic) and techniques (prototyping and theoretical proof techniques) that implement verification and validation, and provides hands-on experience to apply these methods and techniques to some simple case studies. Automation of verification and validation methods will also be briefly discussed. Prerequisite: CS 741 or concurrent enrollment. Offered Fall.

CS 744 Cr.3
Software Project Management
This course addresses principles, standards, guidelines and techniques for software project management. Emphasis will be given to modern software development approaches. Topics covered in this courses include people management, work allocation, schedule, project planning, cost estimation, risk management, project deployment, licenses, and ethical and legal issues. Prerequisite: CS 741. Offered Spring.

CS 746 Cr.3
Software Modeling and Analysis
This course introduces various software models, and techniques to analyze software designs using these models. Both diagrammatic and mathematical models will be included. Informal, rigorous, and formal analysis will be covered. Prerequisite: CS 225, CS 340. Offered Spring.

CS 750 Cr.1-3
Topics in Software Engineering
This is a topics course in Software Engineering. New topics will be introduced based on the evolution of Software Engineering research. Some such topics are real-time systems, embedded systems, software for safety-critical applications, software architectures, component-oriented programming, CORBA, COM/DCOM, and CASE (Computer-Aided Software Engineering). Topics may vary each semester. Repeatable for credit - maximum six. Prerequisite: CS 741. Consent of instructor. Offered Occasionally.

CS 751 Cr.1-3
Seminar in Software Engineering
This course is meant for those who want to specialize in one or more areas in Software Engineering such as software reuse, software architectures, software testing, software verification, etc. The workload for the course will include a number of presentations in the class and one or more written reports. Topics of specialization may vary for each semester. Repeatable for credit - maximum six. Prerequisite: CS 741. Consent of instructor. Offered Occasionally.

CS 752 Cr.1-3
Independent Study
This course is meant for those who want to acquire an in-depth knowledge on any Software Engineering topic. Typically, the student may be required to focus on one particular topic and conduct some research on this topic, or to do some software development activities such as analysis, design, implementation or testing. If registered for more than once, a different topic must be chosen each time. Each student is required to submit a report at the end of the term. Repeatable for credit - maximum six. Prerequisite: CS 741. Consent of instructor. Offered Fall, Spring, Summer.

CS 795 Cr.1
Software Development Internship
An academically relevant field experience in government, industry, business, or community agencies. Students must have their internships approved and be advised by the computer science department. Determination of relevancy shall be made by the Career Services Office with the advice and consent of the computer science department. The experience will be supervised closely by the intern's on-site supervisor, by the Career Services staff, and by the student's faculty internship adviser. Students should contact the Career Services Office. Internship does not count for credit towards the MSE program. Repeatable for credit - maximum two. Prerequisite: Master of Software Engineering graduate student status; nine MSE credtis earned; 3.5 or higher GPA. Student must be on their internship work site during the semester for which they are registered for academic credit. Consent of instructor. Pass/Fail grading. Offered Fall, Spring, Summer.

CS 798 Cr.1-6
Software Development Project
A major project that requires a detailed analysis of the problem domain, detailed design, implementation and demonstration. The project will be guided by a graduate CS faculty member. Submission of a written project report is required, followed by an oral examination by the Project Evaluation Committee in the CS department. Repeatable for credit - maximum 12. Maximum of six credits per semester. Prerequisite: project proposal must be approved by the Project Evaluation Committee in the CS department. Pass/Fail grading. Offered Fall, Spring, Summer.