

Computer Science & Software Engineering (CSE)

CSE 102. Introduction to Computer Science and Software Engineering. (3)

This course introduces students to the computer science and software engineering disciplines. The course focuses on various computing and software design principles and tools used in the profession. Students will be able to model, implement, and test these principles via projects required throughout the course. This course is open to all majors. Credit will be given for only one of CPB 102, CSE 102, ECE 102, MME 102, CEC 102.

CSE 148. Business Computing. (3)

An introduction to business-oriented computer skills. Extensive hands-on use of electronic spreadsheets and database software. Examples and exercises will stress problem-solving in a business context. Credit not awarded for both CSE 148 and CSE 243.

CSE 151. Computers, Computer Science, and Society. (3)

Perspective on the potential and limitations of computing technology. Topics include problem-solving in computing, computers as thinking machines, and the impact of computing on societies. Exposes students to programming languages and various computer tools. Not open to CSE and ISA majors. V. PA-1A.

CSE 153. Introduction to C/C++ Programming. (3)

Introduction to use of C/C++ programming language as an aid to solving mathematical and scientific problems. Students design, write, and implement programs.

CSE 163. Introduction to Computer Concepts and Programming. (3)

Introduction to computers in data processing, survey of various hardware and software concepts, and analysis and solution of problems by computer programming. Lecture/laboratory, project-oriented course to provide numerous opportunities to analyze problems, formulate alternative solutions, implement solutions, and assess their effectiveness. No prior knowledge of computer concepts or programming assumed. V. PA-1A.

Prerequisite: ACT Math score of 19 or higher, OR SAT Math Score of 510 or higher, or permission of instructor.

CSE 174. Fundamentals of Problem Solving and Programming. (3)

Algorithm development and refinement in problem solving. Modular programming using sequence, selection, and repetition control structures. Built-in and programmer-defined functions. Problem solving with collections. Program debugging and testing. Formatted input and output. Data file input and output. Fundamental data types. Character data and string processing. Introduction to recursion. Algorithm efficiency considerations.

2 Lec. 1 Lab.

Prerequisite: ACT Math Score of 20 or higher, or SAT Math Score of 520 or higher, or Miami Math Placement Test score of 7 or higher, or successful completion of MTH 025.

Prerequisite or Co-requisite: MTH 122 or 124 or 125 or 141 or 151.

CSE 177. Independent Studies. (0-6; maximum 10)

CSE 201. Introduction to Software Engineering. (3)

Principles of software engineering: Introduction to all phases of the software development life cycle and associated tools and engineering methods including the unified modeling language (UML).

Prerequisite or Co-requisite: CSE 274.

CSE 202. Software Requirements. (3)

Domain engineering. Techniques for discovering and eliciting requirements. Languages and models for representing requirements. Analysis and validation techniques, including need, goal, and use case analysis. Specifying and measuring external qualities. Traceability. Agile approaches.

Prerequisite or Co-requisite: CSE 201.

CSE 212. Software Engineering for User Interface and User Experience Design. (3)

Principles of user interfaces (UI) and user experiences (UX) for software engineering. Psychological principles; Design methods such as task analysis and user-centered design. Projects demonstrating window, menu, and command design; voice and natural language I/O; response time and feedback; color, icons, sound.

Prerequisite: CSE 201.

CSE 220. Professional Practice. (0)

Students participating in computer technology associate's degree co-op program register for this course during semesters when they are on work assignment. This enables students to maintain continuing student status with the university.

CSE 235. Computer Network Design and Administration. (3)

This course introduces students to the fundamentals of computer networking, the OSI stack, and the practical and conceptual skills needed to build simple local area networks, perform basic device configurations, and implement subnet schemes and their IP addressing. Students will also configure and troubleshoot simulated networks consisting of end devices, switches and routers.

Prerequisite: CYB 234 or CSE 278.

Cross-listed with CYB 235.

CSE 243. Problem Analysis Using Computer Tools. (3)

Students will learn to use personal computer productivity tools to analyze data, work with others in conducting analyses, develop conclusions and effectively communicate results. Students will utilize spreadsheet tools to analyze data and will be challenged to evaluate data from multiple perspectives in order to develop conclusions supported by their analysis. Students will use word processing tools to integrate text and graphical information that clearly and concisely communicates their conclusions. While an important part of the course is learning to use the software tools, the emphasis of the course is learning to use these tools to solve problems and communicate results. Credit not awarded for both CSE 243 and CSE 148. V. PA-1A.

CSE 251. Introduction to Game Programming. (3)

Introduction to computer programming techniques used in games and visual simulations. Simple data and control structures, mathematical foundations, transformations, rendering algorithms, and interfaces. This course is not open to CSE students without permission of instructor.

Prerequisite: high school algebra and trigonometry.

CSE 252. Web Application Programming. (3)

An introduction to programming concepts and practices for creating applications which use the web as the delivery platform. Students will learn technologies including HTML, Javascript, AJAX, client side programming and server side scripting to create interactive web applications. Not an elective for computer science and systems analysis majors.

Prerequisite: CSE 153 or CSE 163 or CSE 174.

CSE 253. Programming Languages. (1-2)

Presents syntax and semantics of a particular programming language currently popular in industrial or academic settings. Addresses construction of programs in the language. Applications of the language presented. Coverage of good programming style and software engineering concepts addressed in context of the language. Not applicable to CSE electives requirement for a CSE major.

CSE 256. Introduction to Programming for the Life Sciences. (3)

Introduction to programming for majors in the life sciences. The ability to write programs to perform tasks related to the organization and analysis of biological data has become a highly-valued skill for researchers in the life sciences, allowing wet-lab researchers to quickly process and sort through large amounts of data to find information relative to their own work. This course serves as an introduction to programming designed specifically for life science majors, targeting the specific skills and techniques commonly needed and explaining the fundamental methods of working with biological data while centering programming assignments around topics of interest to those studying the life sciences. Topics covered include basic programming techniques, representation and manipulation of genomic and protein sequence data, and the automated interface with BLAST and the NCBI GenBank database.

Cross-listed with BIO/MBI.

CSE 262. Technology, Ethics, and Global Society. (3)

Inquiry into a wide range of information technology issues, from moral responsibilities affecting professionals to wider ethical concerns associated with information technology in day-to-day living. Topics include general aspects of ethics; common ethical theories; professional codes of ethics in IT; privacy, security and reliability in using computer systems and the internet; issues and responsibilities in internet usage; legal issues in IT; global perspectives of computing issues; and general problems related to ethical and responsible computing. Not open to first-year students. IIB, IIC.

Prerequisites: ENG 109 or ENG 111.

Cross-listed with CIT 262.

CSE 268. Introduction to Knowledge Representation. (3)

Introduces fundamental concepts in the field of knowledge representation and reasoning. Students learn how to use state-of-the-art logic-based languages, techniques, and tools to develop models that capture information about the world to be used for solving complex problems. They develop small intelligent systems capable of performing reasoning tasks such as planning, question answering, or problem solving. Students also benefit from interaction with experts in the field of knowledge representation, who will be invited to present lectures on their research.

Prerequisite: CSE 271.

CSE 270. Special Topics. (3; maximum 6)

Special topics in computer science, computer information systems, or operations research.

Prerequisite: permission of instructor.

CSE 271. Object-Oriented Programming. (3)

Review of sequence, selection, and iteration. The design and implementation of software using object-oriented programming techniques including inheritance, polymorphism, object persistence, and overloading. Students will analyze program specifications and identify appropriate data types, objects, and classes. Additional programming topics include dynamic memory, recursion, searching and sorting, using existing object libraries, and binary and text file processing.

Prerequisite: CSE 174 with a grade of C- or better.

CSE 273. Optimization Modeling. (3)

Use of deterministic models and computers to study and optimize systems. Includes an introduction to modeling, calculus-based models, financial models, spreadsheet models, and linear-programming models.

Prerequisite: MTH 251 or equivalent.

CSE 274. Data Abstraction and Data Structures. (3)

Abstract data types and their implementation as data structures using object-oriented programming. Use of object-oriented principles in the selection and analysis of various ADT implementations. Sequential and linked storage representations: lists, stacks, queues, and tables. Nonlinear data structures: trees and graphs. Recursion, sorting, searching, and algorithm complexity.

Prerequisites: C- or higher in CSE 271.

CSE 276. Mathematics and Computer Science. (3)

This course examines how mathematics has influenced computer science, and how computer science has influenced mathematics. It covers a range of topics which sit at the intersection of mathematics and computer science, such as encryption, randomness, computational geometry, fractals, prime numbers, numerical methods, image processing, and simulations. Most topics are approached through developing static and animated visualizations of mathematical concepts.

Prerequisite: CSE 271 and (MTH 231 or MTH 331).

CSE 277. Independent Studies. (0-6; maximum 10)**CSE 278. Systems I: Introduction to Systems Programming. (3)**

Principles of Von Neumann computer architecture through the C/C++ programming language. Data representation and computer arithmetic. Memory hierarchy. CPU structure and instruction sets. Network programming including use of sockets. Database programming through SQL.

Prerequisite: CSE 271.

CSE 301. Software Architecture and Design. (3)

An in-depth look at software design. Study of software architecture, design patterns and software product lines. Designing for quality attributes such as performance, safety, security, reusability, reliability, etc. Measuring internal qualities and complexity of software designs. Evolution of designs. Basics of software evolution, reengineering, and reverse engineering. Application of formal methods to specify and evaluate designs.

Prerequisites: CSE 201 and CSE 202.

CSE 302. Software Construction. (3)

General principles and techniques for disciplined detailed software design. Basic theory of grammars and parsing. State-transition and table-based design. Formal specification languages and program derivation. Techniques for handling concurrency and inter-process communication. Tools for model-driven construction.

Prerequisite: CSE 301 and (MTH 231 or MTH 331).

CSE 310. Undergraduate Research Seminar. (1; maximum 3)

Seminar or workshop on topics in computer science, software engineering, or related fields.

CSE 320. Professional Practice. (0)

Students participating in the computer science and systems analysis co-op program register for this course during semesters when they are away from Oxford on work assignment. This enables students to maintain continuing student status with the university.

CSE 340. Internship. (0-20)**CSE 340U. Undergraduate Summer Scholars Program. (1-12)****CSE 372. Stochastic Modeling. (3)**

Survey of methods of stochastic operations research including reliability, Markov processes, queuing theory, and decision theory. Computer used for modeling and solving problems.

Prerequisite: STA 301 or STA 368.

Prerequisite or Co-requisite: STA 401/STA 501.

CSE 374. Algorithms I. (3)

Design, analysis and implementation of algorithms and data structures. Dynamic programming, brute force algorithms, divide and conquer algorithms, greedy algorithms, graph algorithms, and red-black trees. Other topics include: string matching and computational geometry.

Prerequisites: CSE 274 and (MTH 231 or MTH 331).

CSE 377. Independent Studies. (0-6; maximum 10)**CSE 381. Systems 2: OS, Concurrency, Virtualization, and Security. (3)**

Introduction to operating systems concepts. The operating system as a resource manager. The principles for the design and implementation of operating systems. Process scheduling and deadlock prevention. Memory management, virtual memory, paging, and segmentation. Interrupt processing. Device management, I/O systems and I/O processing. Concurrency and multithreading. Virtualization and cloud services. Security and protection.

Prerequisite: CSE 278.

CSE 382. Mobile App Development. (3)

Implementation of cross-platform applications for mobile platforms such as iOS and Android. Programming languages, development environments, debugging, testing, and application design.

Development of applications that: provide an effective graphical interface, access internet resources, permanently store data, access the device's hardware, and display graphical elements.

Prerequisite: CSE 278.

CSE 383. Web Application Programming. (3)

An introduction to the software, concepts and methodologies necessary to design and implement web applications. Students will design and construct web applications utilizing remote servers on multiple platforms. Projects will be used to enable the students to apply the principles and techniques presented in class.

Prerequisite: CSE 278.

CSE 385. Database Systems. (3)

Overview of database management, database system architecture, and database modeling principles. Logical database design. The relational database model, relational integrity constraints, and relational algebra. Relational commercial database management systems and languages. Interactive database processing, view processing, and database application programming. Database integrity. Relational database design by normalization. File structures for database systems.

Prerequisite or Co-requisite: CSE 274.

CSE 386. Foundations of Computer Graphics and Games. (3)

An introduction to techniques used to create images on the computer. The course covers the algorithms and mathematical theory behind three-dimensional image generation with an emphasis on 3D geometry, 3D transformations, and the graphics pipeline. Programming required.

Prerequisites: MTH 151, and (CSE 274 or CSE 278).

CSE 389. Game Design and Implementation. (3)

Study of architectures, algorithms, and software design patterns used in computer games. Students work with a game engine to design and implement several kinds of games. Topics include animation techniques, physics simulation, user controls, graphical methods, and intelligent behaviors.

Prerequisites: CSE 287 or CSE 386.

CSE 401/CSE 501. Software Quality Assurance and Testing. (3)

Quality: how to assure it and verify it, and the need for a culture of quality. Avoidance of errors and other quality problems. Inspections and reviews. Testing, verification, and validation techniques. Product and process assurance. Formal verification. Statistical testing.

Prerequisite: CSE 302.

CSE 411/CSE 511. Introduction to Model-Driven Software Engineering. (3)

An introduction to model-driven software engineering (MDSE) techniques; applying software engineering practices to model-based artifacts; modeling and abstraction; model transformations; model-based testing; tool implementations.

Prerequisite: CSE 201.

CSE 432/CSE 532. Machine Learning. (3)

This course introduces the process, methods, and computing tools fundamental to machine learning. Students will work on large real-world datasets to write code to accomplish tasks such as predicting outcomes, discovering associations, and identifying similar groups. Students will complete a term project showcasing the different steps of the machine learning process, from data cleaning to the extraction of accurate models and the visualization of results.

Prerequisite: CSE 274.

CSE 433/CSE 533. Deep Learning. (3)

This course introduces basic concepts for deep learning and covers the preliminary knowledge of neural networks. Students will learn to implement and train their own neural networks and gain an understanding of research in deep learning. Additionally, students will complete a final project to design and train a customized neural network on selected real-world problems. Through multiple hands-on assignments and the final course project, students will acquire the skills to perform deep learning tasks and the best practices to train and fine-tune deep neural networks.

Prerequisites: CSE 374.

CSE 434/CSE 534. Generative Artificial Intelligence. (3)

This course introduces students to AI tools that allow the creation of new data such as text (Natural Language Generation), images, or videos. Students use Large-Scale Language Models (LLMs) to generate text, evaluate its quality, and integrate the generator in sample applications. Students also create images via deep neural networks, although no prior familiarity with deep learning is required. Students will complete a term project.

Prerequisite: CSE 374.

CSE 443/CSE 543. High Performance Computing & Parallel Programming. (3)

Introduction to practical use of multi-processor workstations and supercomputing clusters. Developing and using parallel programs for solving computationally intensive problems. The course builds on basic concepts of programming and problem solving.

Prerequisite: CSE 381.

CSE 444/CSE 544. Applied Cryptography. (3)

Cryptography is the study of techniques for protecting information and systems, that is, securing them against adversarial attacks. Cryptography appears everywhere. For example, computer passwords, user authentication, credit cards, cryptocurrency, web traffic (HTTPS), disk encryption, etc., all use cryptography. This course presents the techniques and tools used in modern cryptography with a focus on applications. For example, the course discusses many attacks on various real-worlds systems, the mistakes made in building those systems, and how they could be built in a secure way.

Prerequisites: CSE 274 and MTH 231.

CSE 448. Senior Design Project. (2)

Student teams, with varied academic backgrounds, conduct major open-ended research/design projects. Elements of the design process are considered as well as real-world constraints, such as economic and societal factors, marketability, ergonomics, safety, aesthetics, and ethics; feasibility and design studies performed. SC.

Prerequisites: CSE 201 and CSE 274 and senior standing in student's major.

CSE 449. Senior Design Project. (1-2)

Continuation of CSE 448. Student teams, with varied academic backgrounds, conduct major open-ended research/design projects; implementation, testing, and production of design. Nonmajors can register for 1-2 credits. SC.

Prerequisite: CSE 448.

CSE 451/CSE 551. Web Services and Service Oriented Architectures. (3)

Intro to service-oriented architectures; examine purposes and differences between different web service technologies; analyze shortcomings and strengths of integration techniques; development of cross-platform applications using standard interchange languages.

Prerequisites: CSE 383.

CSE 456/CSE 556. Bioinformatic Principles. (3)

Concepts and basic computational techniques for mainstream bioinformatics problems. Emphasis placed on transforming biological problems into computable ones and seeking solutions.

Prerequisites: (BIO/CSE/MBI 256 or CSE 174) and (BIO/MBI 116 or MBI 201 or BIO 342) or permission of instructor.

Cross-listed with BIO 485/BIO 585/585 and MBI 485/MBI 585/585.

CSE 464/CSE 564. Algorithms. (3)

Review of basic data structures and algorithms. Analysis of algorithms. Problem assessment and algorithm design techniques. Algorithm implementation considerations. Concept of NP-completeness. Analysis of algorithms selected from topics relevant to computer science and software engineering (sorting, searching, string processing, graph theory, parallel algorithms, NP-complete problems, etc.)

Prerequisite: MTH 231 or discrete math and CSE 274 or equivalent.

CSE 465/CSE 565. Comparative Programming Languages. (3)

Survey of programming languages and their accompanying paradigms. Basic principles of syntax, semantics, implementation, and pragmatics are addressed. The survey will include representatives from the families of imperative languages, functional languages, logic languages, and hybrid languages. Formal methods of definition and specification are introduced.

Prerequisite: CSE 274 or equivalent.

CSE 466/CSE 566. Bioinformatics Computing Skills. (3)

Study of the core computational and biological concepts in bioinformatics, with programming in Python, MySQL and Ubuntu OS. You will gain hands-on experience in popular bioinformatics applications, including BLAST, sequence alignment, genome browser, and gene annotation, among others.

Prerequisites: BIO 256; or CSE 174; or permission of instructor.

Cross-listed with BIO/CHM/MBI.

CSE 467/CSE 567. Computer and Network Security. (3)

Fundamentals of network, operating system and application security. Students will study and implement a variety of security techniques including defense, response and forensics. Extensive analysis, reading and writing will be integral to this course.

Prerequisite: CSE 383.

CSE 470/CSE 570. Special Topics in CSE. (3; maximum 21)

Advanced special topics in computer science, computer information systems, or operations research.

Prerequisite: permission of instructor.

CSE 473/CSE 573. Automata, Formal Languages, and Computability. (3)

Regular expressions. Closure properties. Sequential machines and finite state transducers. State minimization. Chomsky hierarchy grammars, pushdown acceptors and linear bounded automata. Closure properties of algorithms on grammars. Turing machine as acceptor and transducer. Universal machine. Computable and noncomputable functions. Halting problem.

Prerequisite: CSE 274 or equivalent and (MTH 231 or MTH 331).

CSE 474/CSE 574. Compiler Design. (3)

Examination of the nature of programming languages and programs which implement them. Compiler and interpreter design and implementation techniques. Review of grammars and languages (context free, context sensitive, regular). Design of interactive interfaces. Parsing of context free languages. Lexical analysis. Semantic analysis and code optimization.

Prerequisite: CSE 274 and CSE 278.

CSE 477. Independent Studies. (0-6; maximum 10)**CSE 480/CSE 580. Special Problems. (1-4; maximum 12)**

Special systems problems decided by students in consultation with instructor. For students in departmental or university honors program.

Prerequisite: permission of department chair prior to registration.

CSE 484/CSE 584. Algorithms II. (3)

Study problems in computer science for which we have no known efficient solutions, and the methods used to recognize intractable problems as well as the current approaches taken to cope with those problems. Concept of NP-completeness and poly-time reductions; an introduction to randomized algorithms and the randomized complexity classes PP, RP, and BPP; an introduction to approximation algorithms for solving NP-Hard problems; polynomial-space algorithms and the classes PSPACE and the poly-time hierarchy; Poly-time approximation schemes and approximation algorithms via linear-program rounding.

Prerequisite: CSE 374.

CSE 485/CSE 585. Advanced Database Systems. (3)

Relational algebra. Query processing and optimization. Database security. Data warehouses, column stores, and ETL. NoSQL database systems, including document and graph stores. Emerging database technologies.

Prerequisites: CSE 385 and (MTH 231 or MTH 331).

CSE 486/CSE 586. Introduction to Artificial Intelligence. (3)

Basic concepts of artificial intelligence (AI) including problem solving, search knowledge representation, and rule-based systems covered with symbolic AI language such as PROLOG or LISP. Application areas (natural language understanding, pattern recognition, learning and expert systems) are explored.

Prerequisite: CSE 274 or equivalent and (MTH 231 or MTH 331).

CSE 488/CSE 588. Image Processing & Computer Vision. (3)

An introduction to computer vision including sensors and image formation, camera geometry, signal processing, feature detection, tracking and motion estimation, scene understanding, image classification, segmentation, object detection, and deep learning.

Prerequisites: CSE 274 and (MTH 231 or MTH 331).

CSE 489/CSE 589. Advanced Graphics and Game Engine Design. (3)

The course covers graphics hardware, game engine architectures, and the mathematics and algorithms used to create digital games. Topics will include shader based rendering and programming. Students will implement portions of a game engine which incorporates animation, collision detection, and simulated physics. Programming required.

Prerequisites: CSE 287 or CSE 386.

CSE 491. Undergraduate Research. (1-4; maximum 10)

Research problems in computer science, systems analysis, or operations research, chosen in consultation with a faculty advisor. Requires a public presentation of completed work. For grade only.

Prerequisites: Permission of instructor and approval of department chair.

CSE 587. Game Design and Implementation. (3)

Study of architectures, algorithms, and software design patterns used in computer games. Students work with a game engine to design and implement several kinds of games. Topics include animation techniques, physics simulation, user controls, graphical methods, and intelligent behaviors.

Prerequisite: CSE 386, for 587: graduate standing or permission of instructor.

CSE 601. Computer Science Research Methods. (3)

An introduction to conducting research in computer science and software engineering. Students will develop basic skills required of a graduate student in CSE including writing scientific papers, performing literature reviews, preparing and delivering presentations, research methods, and participating in academic discussions.

CSE 602. Emerging Topics in Computer Science. (1)

Students develop advanced skills necessary for research in computer science and software engineering. They formulate research questions, propose a methodology to conduct independent research, and design experiments to evaluate their planned research. Students defend their thesis proposal, and attend peer thesis proposal presentations to discover emerging topics in computer science.

Prerequisite: CSE 601.

CSE 610. Seminar in Computer Science. (1-3)

Seminar topics in computer science, computer information systems, or operations research. Does not apply toward fulfillment of the requirements of the Master of Systems Analysis. Credit/no-credit only.

Prerequisite: permission of instructor.

CSE 611. Computer Science Seminar Attendance Requirement. (0; maximum 0)

Required seminar attendance for graduate students in all tracks during each fall and spring semester in which they are enrolled as fulltime students. Attendance must be verified at a designated number of approved events each semester. Approved events include proposal and defense presentations, oral exams associated with the Coursework Only track, and presentations by faculty search candidates.

Prerequisite: graduate standing.

CSE 616. Simulation of Physical Systems. (3)

This course is an introduction to the principles and use of simulation, and suitable software tools, to model the behavior of physical systems in the sciences and engineering. Concepts related to discrete event simulation including random number generation, scheduling and processing are addressed. Concepts related to continuous simulation including linear, nonlinear, and dynamic systems are studied. Students will design and implement simulations using suitable modeling and simulation software tools.

CSE 617. Advanced Networks. (3)

Study of advanced networking techniques, client/ server programming, and distributed processing. Critical analysis of these areas develops as students learn the strengths and weaknesses of these technologies through assigned programming projects.

CSE 620. Special Topics in Computer Science Applications. (3; maximum 12)

Special topics in computer science, computer information systems, or operations research.

Prerequisite: permission of instructor.

CSE 621. Foundations of Software Engineering. (3)

Foundational theories for software engineering. Topics include project management, modeling notations, refinements processes, verification and validation, and evolution.

CSE 627. Machine Learning. (3)

Concepts and algorithms of machine learning including version-spaces, decision trees, instance-based learning, networks, evolutionary computation, Bayesian learning and reinforcement learning.

CSE 640. Internship. (0-12; maximum 6)**CSE 648. Professional Computing Experience. (3)**

The Professional Computing Experience course is an immersive, project-based, experiential course intended to draw on skills and knowledge gained in courses throughout the Master in Computer Science program. Students will be immersed in a semester-long open-ended computer science project that requires the selection and application of appropriate skills, tools and methodologies covered in the MCS program to address the problem at hand. In preparation for the project, students will read and analyze "grey literature" (i.e., professional magazines, technical reports, and working papers) on the current and evolving trends in computing fields. Students will appropriately communicate their findings and project results to multiple audiences.

Prerequisite: Completion of the 12 hours of coursework in the Master in Computer Science.

CSE 650. Special Topics in Computer Science Theory. (3; maximum 18)

This course covers special topics in theory within Computer Science. Understanding theory is fundamental in any Computational Science venture as it is the foundation on which all work and applications are built upon. Faculty will be covering a variety of new and emerging theory areas.

CSE 664. Advanced Algorithms. (3)

A review of NP-Completeness and poly-time reductions; an introduction to randomized algorithms and the randomized complexity classes PP, RP, and BPP; an introduction to approximation algorithms for solving NP-Hard problems; polynomial-space algorithms and the classes PSPACE and the poly-time hierarchy; Poly-time approximation schemes and approximation algorithms via linear-program rounding.

CSE 667. Cryptography. (3)

This course presents the techniques and tools used in modern cryptography. The course covers common cryptographic assumptions and tools, including: pseudorandomness, symmetric key cryptography, and asymmetric key cryptography. Recommended co-requisite: CSE 464/CSE 564/564.

Prerequisite: graduate standing or permission of instructor.

CSE 671. Software Quality. (3)

This course emphasizes the importance of quality when developing software systems. Topics include design for reuse, formal specification, applications of software testing, and maintenance of evolving systems.

Prerequisite: none for MSCS students, permission of instructor for others.

CSE 690. Graduate Research. (3)

Research problems in computer science, computer information systems, or operations research, decided upon in consultation with the instructor and student's graduate adviser. Requires a public presentation of completed work. For grade only.

Prerequisite: permission of instructor, student's graduate adviser, and graduate director.

CSE 700. Research for Master's Thesis. (0-9; maximum 9)

Study under graduate faculty supervision of a research problem related to computer science or software engineering. Students should take CSE 601 either concurrently or before taking CSE 700 for the first time. Upon completion of research, the results must be defended before the advisory committee for approval. A minimum of two semesters of research can be counted toward fulfillment of the research requirement.

Prerequisite or Co-requisite: CSE 601.

CSE 704. Non-Thesis Project. (0-12; maximum 12)

This repeatable course is for non-thesis culminating experiences. Permission of the instructor is required.