

COMPUTER SCIENCE (CS)

CS 100

Introduction to the Profession

An introduction to science and engineering as a profession. Examines the problem-solving process used in engineering and science. Emphasizes the interdisciplinary and international nature of problem-solving and the need to evaluate solutions in terms of a variety of constraints: computational, financial, and social.

Lecture: 1 **Lab:** 2 **Credits:** 2

Satisfies: Communications (C)

CS 104

Introduction to Computer Programming for Engineers

Introduces the use of high-level programming language as a problem-solving tool in engineering including basic data structures and algorithms, structured programming techniques, and software documentation. Designed for students who have had little or no prior experience with computer programming.

Lecture: 2 **Lab:** 1 **Credits:** 2

CS 105

Introduction to Computer Programming

Introduces the use of high-level programming language as a problem-solving tool, including basic data structures and algorithms, structured programming techniques, and software documentation. Designed for students who have had little or no prior experience with computer programming.

Lecture: 2 **Lab:** 1 **Credits:** 2

CS 110

Computing Principles

An introduction to the following "big ideas" of computer science: (1) computing is a creative activity; (2) abstraction reduces information and detail to facilitate focus on relevant concepts; (3) data and information facilitate the creation of knowledge; (4) algorithms are used to develop and express solutions to computational problems; (5) programming enables problem solving, human expression, and creation of knowledge; (6) the internet pervades modern computing; and (7) computing has global impacts.

Lecture: 2 **Lab:** 1 **Credits:** 2

CS 115

Object-Oriented Programming I

Introduces the use of a high-level object-oriented programming language as a problem-solving tool, including basic data structures and algorithms, object-oriented programming techniques, and software documentation. Designed for students who have had little or no prior experience with computer programming. For students in CS and CS-related degree programs.

Lecture: 2 **Lab:** 1 **Credits:** 2

CS 116

Object-Oriented Programming II

Introduces more advanced elements of object-oriented programming, including dynamic data structures, recursion, searching and sorting, and advanced object-oriented programming techniques. For students in CS and CS-related degree programs.

Prerequisite(s): [(CS 115 with min. grade of C)]

Lecture: 2 **Lab:** 1 **Credits:** 2

CS 201

Accelerated Introduction to Computer Science

Problem-solving and design using an object-oriented programming language. Introduces a variety of problem-solving techniques, algorithms, and data structures in object-oriented programming.

Prerequisite(s): [(CS 104 with min. grade of C) OR (CS 105 with min. grade of C) OR (CS 115 with min. grade of C)]

Lecture: 3 **Lab:** 2 **Credits:** 4

CS 330

Discrete Structures

Introduction to the use of formal mathematical structures to represent problems and computational processes. Topics covered include Boolean algebra, first-order logic, recursive structures, graphs, and abstract language models. Credit will not be granted for both CS 330 and MATH 230.

Prerequisite(s): [(CS 116) OR (CS 201)]

Lecture: 3 **Lab:** 1 **Credits:** 3

CS 331

Data Structures and Algorithms

Implementation and application of the essential data structures used in computer science. Analysis of basic sorting and searching algorithms and their relationship to these data structures. Particular emphasis is given to the use of object-oriented design and data abstraction in the creation and application of data structures.

Prerequisite(s): [(CS 116) OR (CS 201)]

Lecture: 3 **Lab:** 1 **Credits:** 3

CS 350

Computer Organization and Assembly Language Programming

Introduction to the internal architecture of computer systems, including micro-, mini-, and mainframe computer architectures. Focuses on the relationship among a computer's hardware, its native instruction set, and the implementation of high-level languages on that machine. Uses a set of assembly language programming exercises to explore and analyze a microcomputer architecture. Credit will not be granted for both CS 350 and ECE 242.

Prerequisite(s): [(CS 116*) OR (CS 201*)]An asterisk (*) designates a course which may be taken concurrently.

Lecture: 3 **Lab:** 1 **Credits:** 3

Satisfies: Communications (C)

CS 351

Systems Programming

Examines the components of sophisticated multilayer software systems, including device drivers, systems software, applications interfaces, and user interfaces. Explores the design and development of interrupt-driven and event-driven software.

Prerequisite(s): [(CS 331 and CS 350) OR (CS 331 and ECE 242)]

Lecture: 3 **Lab:** 1 **Credits:** 3

CS 397

Special Projects

Instructor permission required.

Credit: Variable

CS 401**Introduction to Advanced Studies I**

First course in a two-course sequence that is designed to prepare students for graduate study in computer science. Explores the implementation and application of fundamental data structures and algorithms with an emphasis on object-oriented programming in Java. Examines the relationship between these elements and the mathematical structures that form the foundation of computer science. This course does not apply toward M. S./Ph. D. credit in Computer Science.

Prerequisite(s): [(CS 200) OR (CS 201)]

Lecture: 2 Lab: 2 Credits: 3

CS 402**Introduction to Advanced Studies II**

Second course in a two-course sequence that is designed to prepare students for graduate study in computer science. Explores the development of the multiple layers of software that form a sophisticated software system, from device drivers to application interfaces to user interfaces. Examines how computer architecture influences software development. Emphasizes the design and implementation of interrupt-driven/event-driven software.

Prerequisite(s): [(CS 401)]

Lecture: 2 Lab: 2 Credits: 3

CS 411**Computer Graphics**

Overview of display devices and applications. Vector graphics in two and three dimensions. Image generation, representation, and manipulation. Homogeneous coordinates. Modeling and hidden line elimination. Introduction to raster graphics. Perspective and parallel projections.

Prerequisite(s): [(CS 331) OR (CS 401) OR (CS 403)]

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 422**Data Mining**

This course will provide an introductory look at concepts and techniques in the field of data mining. After covering the introduction and terminologies to Data Mining, the techniques used to explore the large quantities of data for the discovery of meaningful rules and knowledge such as market basket analysis, nearest neighbor, decision trees, and clustering are covered. The students learn the material by implementing different techniques throughout the semester.

Prerequisite(s): [(CS 331) OR (CS 401) OR (CS 403)]

Lecture: 3 Lab: 0 Credits: 3

Satisfies: Communications (C), CS Technical Elective (T)

CS 425**Database Organization**

Overview of database architectures, including the Relational, Hierarchical, Network, and Object Models. Database interfaces, including the SQL query language. Database design using the Entity-Relationship Model. Issues such as security, integrity, and query optimization.

Prerequisite(s): [(CS 331) OR (CS 401) OR (CS 403)]

Lecture: 3 Lab: 0 Credits: 3

Satisfies: Communications (C), CS Technical Elective (T)

CS 429**Information Retrieval**

Overview of fundamental issues of information retrieval with theoretical foundations. The information-retrieval techniques and theory, covering both effectiveness and run-time performance of information-retrieval systems are covered. The focus is on algorithms and heuristics used to find documents relevant to the user request and to find them fast. The course covers the architecture and components of the search engine such as parser, stemmer, index builder, and query processor. The students learn the material by building a prototype of such a search engine. Requires strong programming knowledge.

Prerequisite(s): [(CS 331) OR (CS 401)]

Lecture: 3 Lab: 0 Credits: 3

Satisfies: Communications (C), CS Technical Elective (T)

CS 430**Introduction to Algorithms**

Introduction to the design, behavior, and analysis of computer algorithms. Searching, sorting, and combinatorial algorithms are emphasized. Worst case, amortized, and expected bounds on time and space usage.

Prerequisite(s): [(CS 330 and CS 331) OR (CS 331 and MATH 230) OR (CS 401) OR (CS 403)]

Lecture: 3 Lab: 1 Credits: 3

Satisfies: Communications (C), CS Technical Elective (T)

CS 440**Programming Languages and Translators**

Study of commonly used computer programming languages with an emphasis on precision of definition and facility in use. Scanning, parsing, and introduction to compiler design. Use of compiler generating tools.

Prerequisite(s): [(CS 330 and CS 331) OR (CS 331 and MATH 230) OR (CS 401) OR (CS 403)]

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 442**Mobile Applications Development**

Students will learn a variety of software engineering techniques and design patterns to assist in the rapid development and prototyping of applications, leveraging frameworks and APIs provided by current mobile development platforms (such as Android and iOS). Application lifecycles, data management and persistence mechanisms, and user interface design, among other topics, will be covered. Industry speakers will be invited to speak about best practices. Students (individually or in teams) will take ideas from concept to final implementation and will present their work at the end of the semester. When appropriate, students may take the additional step of deploying their work on the appropriate application marketplace(s).

Prerequisite(s): [(CS 331) OR (CS 401)]AND[(CS 351*) OR (CS 402*)]An asterisk (*) designates a course which may be taken concurrently.

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 443**Compiler Construction**

This course covers the design and implementation of a compiler for modern languages by implementing the following: abstract syntax trees; intermediate representations; static analysis; fix-point operations; symbol tables and type checking; and first-order and high-order function implementation. Students will incrementally create a series of compilers.

Prerequisite(s): [(CS 440)]

Lecture: 3 Lab: 0 Credits: 3

CS 445**Object Oriented Design and Programming**

Introduction to methodologies for object-oriented design and programming. Examines the object model and how it is realized in various object-oriented languages. Focuses on methods for developing and implementing object-oriented systems.

Prerequisite(s): [(CS 331) OR (CS 401) OR (CS 403)]

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 447**Distributed Objects**

This course provides an introduction to architecture, analysis, design, and implementation of distributed, multi-tier applications using distributed object technology. The course focuses on the services and facilities provided by an Object Request Broker (ORB). Students will use a commercially available ORB and Database Management System to develop distributed object applications.

Prerequisite(s): [(CS 445)]

Lecture: 3 Lab: 0 Credits: 3

Satisfies: Communications (C), CS Technical Elective (T)

CS 450**Operating Systems**

Introduction to operating system concepts-including system organization for uniprocessors and multiprocessors, scheduling algorithms, process management, deadlocks, paging and segmentation, files and protection, and process coordination and communication.

Prerequisite(s): [(CS 351) OR (CS 401 and CS 402) OR (CS 403)]

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 451**Introduction to Parallel and Distributed Computing**

This course covers general introductory concepts in the design and implementation of parallel and distributed systems covering all the major branches such as cloud computing, grid computing, cluster computing, supercomputing, and many-core computing.

Prerequisite(s): [(CS 450)]

Lecture: 3 Lab: 0 Credits: 3

CS 455**Data Communications**

Introduction to data communication concepts and facilities with an emphasis on protocols and interface specifications. Focuses on the lower four layers of the ISO-OSI reference model.

Prerequisite(s): [(CS 450)]

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 456**Introduction to Wireless Networks and Performance**

This class provides an opportunity for students to obtain a fundamental understanding of the nature and operation of the full range of wireless networks (personal, local area, wide area, and satellite) and their performance characteristics, future potential, and challenges through class lectures, assigned readings, homework, projects, and various hands-on experiences.

Prerequisite(s): [(CS 350) OR (CS 401 and CS 402) OR (CS 403) OR (ECE 242)]

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 458**Introduction to Information Security**

An introduction to the fundamentals of computer and information security. This course focuses on algorithms and techniques used to defend against malicious software. Topics include an introduction to encryption systems, operating system security, database security, network security, system threats, and risk avoidance procedures.

Prerequisite(s): [(CS 425) OR (CS 450) OR (CS 455)]

Lecture: 3 Lab: 0 Credits: 3

Satisfies: Communications (C), CS Technical Elective (T)

CS 470**Computer Architecture**

Introduction to the functional elements and structures of digital computers. Detailed study of specific machines at the register transfer level illustrates arithmetic, memory, I/O and instruction processing.

Prerequisite(s): [(CS 350 and ECE 218) OR (ECE 218 and ECE 242)]

Lecture: 2 Lab: 2 Credits: 3

Satisfies: Communications (C), CS Technical Elective (T)

CS 480**Artificial Intelligence Planning and Control**

Introduction to computational methods for intelligent control of autonomous agents, and the use of programming paradigms that support development of flexible and reactive systems. These include heuristic search, knowledge representation, constraint satisfaction, probabilistic reasoning, decision-theoretic control, and sensor interpretation. Particular focus will be places on real-world application of the material.

Prerequisite(s): [(CS 331 and MATH 474*) OR (CS 401 and CS 402)]An asterisk (*) designates a course which may be taken concurrently.

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 481**Artificial Intelligence Language Understanding**

Theory and programming paradigms that enable systems to understand human language texts and extract useful information and knowledge. For example, extraction of structured event representations from news stories or discovering new research hypotheses by analyzing thousands of medical research articles. The course covers a variety of text analysis and text mining methods, with an emphasis on building working systems. Connections to information retrieval, data mining, and speech recognition will be discussed.

Prerequisite(s): [(CS 331) OR (CS 401) OR (CS 403)]AND[(MATH 474)]

Lecture: 3 **Lab:** 0 **Credits:** 3

Satisfies: CS Technical Elective (T)

CS 482**Information and Knowledge Management Systems**

This capstone course is designed as a project course whose purpose is to enable students to see how the various algorithms and systems they have learned about in their prerequisite courses can be used in context to create useful knowledge management tools. Class periods will be divided among discussion of design of information and knowledge management systems, lectures on effective project management techniques, and hands-on advising of student project group meetings.

Prerequisite(s): [(CS 422, CS 425, and CS 429) OR (CS 422, CS 425, and CS 481) OR (CS 425, CS 429, and CS 481)]

Lecture: 3 **Lab:** 0 **Credits:** 3

Satisfies: Communications (C), CS Technical Elective (T)

CS 485**Computers and Society**

Discussion of the impact of computer technology on present and future society. Historical development of the computer. Social issues raised by cybernetics.

Prerequisite(s): [(COM 421) OR (COM 424) OR (COM 425) OR (COM 428) OR (COM 435)]

Lecture: 3 **Lab:** 0 **Credits:** 3

Satisfies: Communications (C)

CS 487**Software Engineering I**

Study of the principles and practices of software engineering. Topics include software quality concepts, process models, software requirements analysis, design methodologies, software testing and software maintenance. Hands-on experience building a software system using the waterfall life cycle model. Students work in teams to develop all life cycle deliverables: requirements document, specification and design documents, system code, test plan, and user manuals.

Prerequisite(s): [(CS 331) OR (CS 401) OR (CS 403)]AND[(CS 425)]

Lecture: 3 **Lab:** 0 **Credits:** 3

Satisfies: Communications (C), CS Technical Elective (T)

CS 491**Undergraduate Research**

Instructor permission required.

Credit: Variable

CS 495**Topics in Computer Science**

This course will treat a specific topic, varying from semester to semester, in which there is particular student or staff interest.

Credit: Variable

CS 497**Special Projects**

Special projects.

Credit: Variable

CS 511**Topics in Computer Graphics**

Covers advanced topics in computer graphics. The exact course contents may change based on recent advances in the area and the instructor teaching it. Possible topics include: Geometric modeling, Subdivision surfaces, Procedural modeling, Warping and morphing, Model reconstruction, Image based rendering, Lighting and appearance, Texturing, Natural phenomena, Nonphotorealistic rendering Particle systems, Character animation, Physically based modeling and animation.

Prerequisite(s): [(CS 411)]

Lecture: 3 **Lab:** 0 **Credits:** 3

CS 512**Computer Vision**

Introduction to fundamental topics in computer vision and the application of statistical estimation techniques to this area. Intended to give the student a good basis for work in this important field. Topics include: Feature extraction, Probabilistic modeling, Camera calibration, Epipolar geometry, Statistical estimation, Model reconstruction, Statistical filtering, Motion estimation, Recognition, Shape from single image cues.

Prerequisite(s): [(CS 430)]

Lecture: 3 **Lab:** 0 **Credits:** 3

CS 513**Geospatial Vision and Visualization**

Geospatial information has become ubiquitous in everyday life as evidenced by on-line mapping services such as NOKIA Here Map, Microsoft Bing Map, the "place" features on social network websites such as Facebook, and navigation apps on smart phones. Behind the scenes is digital map content engineering that enables all types of location-based services. Course material will be drawn from the instructor's research and development experience at NOKIA Location and Commerce (formerly NAVTEQ), the Chicago-based leading global provider of digital map, traffic, and location data. This course will provide a comprehensive treatment of computer vision, image processing and visualization techniques in the context of digital mapping, global positioning and sensing, next generation map making, and three-dimensional map content creations. Real world problems and data and on-site industry visits will comprise part of the course curriculum.

Lecture: 3 **Lab:** 0 **Credits:** 3

CS 520**Data Integration, Warehousing, and Provenance**

This course introduces the basic concepts of data integration, data warehousing, and provenance. We will learn how to resolve structural heterogeneity through schema matching and mapping. The course introduces techniques for querying several heterogeneous datasources at once (data integration) and translating data between databases with different data representations (data exchange). Furthermore, we will cover the data-warehouse paradigm including the Extract-Transform-Load (ETL) process, the data cube model and its relational representations (such as snowflake and star schema), and efficient processing of analytical queries. This will be contrasted with Big Data analytics approaches that (besides other differences) significantly reduce the upfront cost of analytics. When feeding data through complex processing pipelines such as data exchange transformations or ETL workflows, it is easy to lose track of the origin of data. Therefore, in the last part of the course we cover techniques for representing and keeping track of the origin and creation process of data (its provenance). The course emphasizes practical skills through a series of homework assignments that help students develop a strong background in data integration systems and techniques. At the same time, it also addresses the underlying formalisms. For example, we will discuss the logic based languages used for schema mapping and the dimensional data model as well as their practical application (e.g., developing an ETL workflow with rapid miner and creating a mapping between two example schemata). The literature reviews will familiarize students with data integration and provenance research.

Prerequisite(s): [(CS 425)]

Lecture: 3 Lab: 0 Credits: 3

CS 521**Object-Oriented Analysis and Design**

This course describes a methodology that covers a wide range of software engineering techniques used in system analysis, modeling and design. These techniques integrate well with software process management techniques and provide a framework for software engineers to collaborate in the design and development process. The methodology features the integration of concepts, including software reusability, frame works, design patterns, software architecture, software component design, use-case analysis, event-flow analysis, event-message analysis, behavioral-life cycle analysis, feature, multiple-product, risk and rule analysis, and automatic code generation. (Credit will not be given for CS 521 if CS751 is taken)

Prerequisite(s): [(CS 445) OR (CS 487)]

Lecture: 3 Lab: 0 Credits: 3

CS 522**Advanced Data Mining**

Continued exploration of data mining algorithms. More sophisticated algorithms such as support vector machines will be studied in detail. Students will continuously study new contributions to the field. A large project will be required that encourages students to push the limits of existing data mining techniques.

Prerequisite(s): [(CS 422)]

Lecture: 3 Lab: 0 Credits: 3

CS 525**Advanced Database Organization**

Comprehensive coverage of the problems involved in database system implementation and an in-depth examination of contemporary structures and techniques used in modern database management systems. Teaches advanced skills appropriate for DBMS architects and developers, database specialist, and the designers and developers of client/server and distributed systems. Focus is on transaction management, database structures and distributed processing.

Prerequisite(s): [(CS 425)]

Lecture: 3 Lab: 0 Credits: 3

CS 529**Information Retrieval**

The course covers the advanced topics in Information Retrieval. The topics such as Summarization, cross-lingual, Meta-Search, Question Answering, Parallel and distributed IR systems are discussed. The students get involved in research ideas, and get involved in individual and group projects.

Prerequisite(s): [(CS 429)]

Lecture: 3 Lab: 0 Credits: 3

CS 530**Theory of Computation**

Computability topics such as Turing machines, nondeterministic machines, undecidability, and reducibility. Computational complexity topics such as time complexity, NP-completeness and intractability, time and space hierarchy theorems. Introduces the complexity classes P, NP, NL, L, PSPACE, NC, RNC, BPP and their complete problems.

Prerequisite(s): [(CS 430)]

Lecture: 3 Lab: 0 Credits: 3

CS 531**Topics in Automata Theory**

Topics selected from mathematical systems and automata theory, decision problems, realization and minimization, algebraic decomposition theory and machines in a category.

Prerequisite(s): [(CS 430)]

Lecture: 3 Lab: 0 Credits: 3

CS 533**Computational Geometry**

This course covers fundamental algorithms and data structures for convex hulls, Voronoi diagrams, Delauney triangulation, Euclidean spanning trees, point location, and range searching. Also included are lower bounds and discrepancy theory. Optimization in geometry will be covered. This includes fixed dimensional linear programming and shortest paths. Graphic data structures such as BSP trees will be covered.

Prerequisite(s): [(CS 430)]

Lecture: 3 Lab: 0 Credits: 3

CS 535**Design and Analysis of Algorithms**

Design of efficient algorithms for a variety of problems, with mathematical proof of correctness and analysis of time and space requirements. Topics include lower bounds for sorting and medians, amortized analysis of advanced data structures, graph algorithms (strongly connected components, shortest paths, minimum spanning trees, maximum flows and bipartite matching) and NP-Completeness.

Prerequisite(s): [(CS 430)]

Lecture: 3 Lab: 0 Credits: 3

CS 536**Science of Programming**

Formal specification of how programs execute operational semantics, how mathematical functions programs compute denotational semantics, and how to use logic to characterize properties and invariants of the program execution (axiomatic semantics).

Prerequisite(s): [(CS 331) OR (CS 401)]

Lecture: 3 Lab: 0 Credits: 3

CS 537**Software Metrics**

Theoretical foundations for software metrics. Data collection. Experimental design and analysis. Software metric validation. Measuring the software development and maintenance process. Measuring software systems. Support for metrics. Statistical tools. Setting up a measurement program. Application of software measurement.

Prerequisite(s): [(CS 487)]

Lecture: 3 Lab: 0 Credits: 3

CS 538**Combinatorial Optimization**

Linear programs and their properties. Efficient algorithms for linear programming. Network flows, minimum cost flows, maximum matching, weighted matching, matroids. Prerequisite: CS 430 and a linear algebra course.

Prerequisite(s): [(CS 430)]

Lecture: 3 Lab: 0 Credits: 3

CS 539**Game Theory: Algorithms and Applications**

This course focuses on computational issues in the theory of games, economics, and network design. Interest in the algorithmic aspects of games is motivated by the computational issues of fundamental aspects of games and economic theory, e.g. Nash equilibrium and market equilibrium. Computing and approximating Nash equilibrium will be studied. Of considerable interest to the computer science community are problems that arise from the Internet and computer networks and are similar to issues that arise in traditional transport networks, e.g. Wardrop equilibrium.

Prerequisite(s): [(CS 430) OR (CS 530)]

Lecture: 3 Lab: 0 Credits: 3

CS 540**Syntactic Analysis of Programming Languages**

Formal definition of syntax with emphasis on context-free languages. Elementary techniques for scanning and parsing programming languages. Symbol table management. Semantic routines and code generation. The class will write a simple translator.

Prerequisite(s): [(CS 440)]

Lecture: 3 Lab: 0 Credits: 3

CS 541**Topics in Compiler Construction**

Advanced topics in compiler construction, including incremental and interactive compiling, error correction, code optimization, models of code generators, etc. The objective of the course is to provide an in-depth coverage of compiler optimization techniques, including both classical optimization and areas of current interest in compiler research.

Prerequisite(s): [(CS 440)]

Lecture: 3 Lab: 0 Credits: 3

CS 542**Computer Networks I: Fundamentals**

This course focuses on the engineering and analysis of network protocols and architecture in terms of the Internet. Topics include content distribution, peer-to-peer networking, congestion control, unicast and multicast routing, router design, mobility, multimedia networking quality of service, security and policy-based networking.

Prerequisite(s): [(CS 455)]

Lecture: 3 Lab: 0 Credits: 3

CS 544**Computer Networks II: Network Services**

Qualitative and quantitative analysis of networks. A combination of analytical and experimental analysis techniques will be used to study topics such as protocol delay, end-to-end network response time, intranet models, Internet traffic models, web services availability, and network management.

Prerequisite(s): [(CS 542) OR (ECE 545)]

Lecture: 3 Lab: 0 Credits: 3

CS 545**Distributed Computing Landscape**

Introduction to the theory of concurrent programming languages. Topics include formal models of concurrent computation such as process algebras, nets, and actors; high-level concurrent programming languages and their operational semantics; and methods for reasoning about correctness and complexity of concurrent programs.

Prerequisite(s): [(CS 450)]

Lecture: 3 Lab: 0 Credits: 3

CS 546**Parallel and Distributed Processing**

This course covers general issues of parallel and distributed processing from a user's point of view which includes system architectures, programming, performance evaluation, applications, and the influence of communication and parallelism on algorithm design.

Prerequisite(s): [(CS 430 and CS 450)]

Lecture: 3 Lab: 0 Credits: 3

CS 547**Wireless Networking**

This course introduces cellular/PCS systems, short-range mobile wireless systems, fixed wireless systems, satellites, and ad hoc wireless systems. It explains in detail the underlying technology as well as regulations, politics, and business of these wireless communications systems. It looks beyond the hype, examining just what is and is not possible with present-day and future wireless systems. As an advanced graduate course, it will combine extensive reading and in-class discussion of the research literature with in-depth independent research projects of students' own choosing.

Prerequisite(s): [(CS 455)]

Lecture: 3 Lab: 0 Credits: 3

CS 548**Broadband Networks**

The course studies the architectures, interfaces, protocols, technologies, products and services for broadband (high-speed) multimedia networks. The key principles of the protocols and technologies used for representative network elements and types of broadband network are studied. Specifically, cable modems, Digital Subscriber Lines, Power Lines, wireless 802.16 (WiMax), and broadband cellular Internet are covered for broadband access; for broadband Local Area Networks (LANs), Gigabit Ethernet, Virtual LANs and wireless LANs (802.11 WiFi and Bluetooth) are discussed; for broadband Wide Area Networks (WANs) the topics covered include optical networks (SONET/SDH, DWDM, optical network nodes, optical network nodes, optical switching technologies), frame-relay, ATM, wire-speed routers, IP switching, and MPLS. Also, quality of service issues in broadband networks and a view of the convergence of technologies in broadband networks are covered.

Prerequisite(s): [(CS 455)]

Lecture: 3 Lab: 0 Credits: 3

CS 549**Cryptography and Network Security**

This course provides an introduction to the theory and practice of cryptography and network security. The course covers conventional encryption such as classical encryption techniques, modern encryption techniques and encryption algorithms. Students are introduced to the basic number theory, which is used as the foundation for public-key encryption. The public-key cryptography such as encryption methods and digital signatures is covered. Message authentication and hash functions are also discussed. Students will learn techniques of key management, secret sharing and conducting interactive proofs. In addition, the practical network and security protocols are discussed.

Prerequisite(s): [(CS 430)]

Lecture: 3 Lab: 0 Credits: 3

CS 550**Advanced Operating Systems**

Advanced operating system design concepts such as interprocess communication, distributed processing, replication and consistency, fault tolerance, synchronization, file systems. Study of systems highlighting these concepts.

Prerequisite(s): [(CS 450)]

Lecture: 3 Lab: 0 Credits: 3

CS 551**Operating System Design and Implementation**

This course covers in detail the design and implementation of processes, interprocess communication, semaphores, monitors, message passing, scheduling algorithm, input/output, device drivers, memory management, file system design, security and protection mechanisms. The hardware-software interface and the user process-system call-kernel interface are examined in detail. Students modify and extend a multiuser operating system.

Prerequisite(s): [(CS 450)]

Lecture: 3 Lab: 0 Credits: 3

CS 552**Distributed Real-Time Systems**

With the advancement of computer hardware, embedded devices, and network technology, real-time applications have become pervasive, ranging from smart automobiles to automated traffic control. Different from general-purpose applications, correct executions of real-time applications depend on both functional correctness and temporal correctness. This course is to study the fundamentals of distributed real-time computing with the focus on its temporal aspects.

Prerequisite(s): [(CS 450)]

Lecture: 3 Lab: 0 Credits: 3

CS 553**Cloud Computing**

This course is a tour through various topics and technologies related to cloud computing. Students will explore solutions and learn design principles for building large network-based systems to support both compute-intensive and data-intensive applications across geographically distributed infrastructure. Topics include resource management, programming models, application models, system characterizations, and implementations. Discussions will often be grounded in the context of deployed cloud computing systems such as Amazon EC2 and S3, Microsoft Azure, Google AppEngine, Eucalyptus, Nimbus, OpenStack, Google's MapReduce, Yahoo's Hadoop, Microsoft's Dryad, Sphere/Sector, and many other systems. The course involves lectures, outside invited speakers, discussions of research papers, programming assignments, and a major project (including both a written report and an oral presentation).

Prerequisite(s): [(CS 450) OR (CS 455)]

Lecture: 3 Lab: 0 Credits: 3

CS 554**Data-Intensive Computing**

This course is a tour through various research topics in distributed data-intensive computing, covering topics in cluster computing, grid computing, supercomputing, and cloud computing. The course will explore solutions and learn design principles for building large network-based computational systems to support data-intensive computing. This course is geared for junior/senior-level undergraduates and graduate students in computer science.

Prerequisite(s): [(CS 450)]

Lecture: 3 Lab: 0 Credits: 3

CS 555**Analytic Models and Simulation of Computer Systems**

Analytic and simulation techniques for the performance analysis of computer architecture, operating systems and communication networks. Rigorous development of queuing models. Study of simulation languages and models.

Prerequisite(s): [(CS 450)]

Lecture: 3 Lab: 0 Credits: 3

CS 556**Cyber-Physical Systems: Languages and Systems**

Different from general-purpose and traditional computer applications, cyber-physical systems have both continuous and discrete components, hence requiring new methodologies to integrate traditional continuous control theory/systems with traditional discrete software systems. The focus of this course is to discuss and understand the challenges in emerging cyber-physical systems and to explore possible solutions from the perspectives of systems specification, system modeling, programming languages, systems designs, and software engineering. This course will focus on the languages and systems aspects of cyber-physical systems.

Lecture: 3 Lab: 0 Credits: 3

CS 557**Cyber-Physical Systems: Networking and Algorithms**

The goal of the course is to provide students with the necessary foundations to apply wireless sensor networking, scheduling theory, and algorithms in the field of computer science. The focus is to discuss and understand the challenges in emerging cyber-physical systems, open distributed real-time systems, and wireless sensor networks. The course will examine different perspectives of wireless networks such as various MAC protocols, routing protocols, scheduling protocols, localization, clock synchronization, data aggregation and data fusion, compressive and cooperative sensing, security, fault detection and diagnosis, online program, and networked control systems. The course will also examine the interaction of different systems.

Lecture: 3 Lab: 0 Credits: 3

CS 558**Advanced Computer Security**

This course will teach various modern topics in network and computer security. It will provide a thorough grounding in cyber-security for students who are interested in conducting research on security and networking and for students who are more broadly interested in real-world security issues and techniques. Students will undertake a semester-long research project with the goal of technical publications. Lecture topics will include, but not limited to: (1) Unwanted traffic, such as denial of service (DoS), and spam; (2) Malware, such as botnet, worm, and virus; (3) Network configuration and defense, such as firewall, access control, and intrusion detection systems; (4) Cyber physical system security, such as critical infrastructure protection (e.g., smart grid); and (5) Hot topics, such as software-defined networking (SDN), network verification, data center and enterprise network security, web security and more.

Prerequisite(s): [(CS 450) OR (CS 455) OR (CS 458)]

Lecture: 3 Lab: 0 Credits: 3

CS 560**Computer Science in the Classroom**

Emphasis on how to organize a selected computer science course. Discussion of what to teach, the problems typically encountered in teaching, and how to best organize the concepts in a computer science course.

Lecture: 3 Lab: 0 Credits: 3

CS 561**The Computer and Curriculum Content**

Presentation techniques from white board to web-based instructional units using currently available software. Emphasis on incorporating the computer as a teaching tool in the presentation of class material. Single Concept Learning Modules (SCLM) are developed.

Lecture: 3 Lab: 0 Credits: 3

CS 565**Computer Assisted Instruction**

Hardware and software for the effective use of the computer in an educational environment, CAI (Computer-Assisted/Aided Instruction) being one of the major areas of investigation .

Prerequisite(s): [(CS 560) OR (CS 561)]

Lecture: 3 Lab: 0 Credits: 3

CS 566**Practicum in the Application of Computers to Education**

Provides supervised experience in the development of computer-based teaching units. Evaluation of different theoretical and/or technical approaches to the use of computer in the classroom.

Prerequisite(s): [(CS 560 and CS 561)]

Lecture: 1 Lab: 4 Credits: 3

CS 570**Advanced Computer Architecture**

Computer system design and architecture such as pipelining and instruction-level parallelism, memory-hierarchy system, interconnection networks, multicore and multiprocessors, and storage architecture. Selected study on current experimental computer systems.

Prerequisite(s): [(CS 450 and CS 470)]

Lecture: 3 Lab: 0 Credits: 3

CS 572**Advanced Topics in Computer Architecture**

Current problems in computer architecture.

Prerequisite(s): [(CS 570)]

Lecture: 3 Lab: 0 Credits: 3

CS 579**Online Social Network Analysis**

This course will explore the latest algorithms for analyzing online social networks, considering both their structure and content. Fundamentals of social graph theory will be covered including distance, search, influence, community discovery, diffusion, and graph dynamics. Fundamentals of text analysis will also be covered with an emphasis on the type of text used in online social networks and common applications. Topics include sentiment classification, information extraction, clustering, and topic modeling. Emphasis will be placed on the application of this technology to areas such as public health, crisis response, politics, and marketing.

Prerequisite(s): [(CS 430)]

Lecture: 3 Lab: 0 Credits: 3

CS 580**Topics in Machine Learning**

This course covers advanced topics in machine learning. The exact course contents may change based on recent advances in the area and the instructor teaching it. Possible topics include active learning, reinforcement learning, online learning, non-parametric learning, inductive learning, statistical relational learning, dimensionality reduction, ensemble methods, transfer learning, outlier detection, specific application areas of machine learning, and other relevant and/or emerging topics.

Lecture: 3 Lab: 0 Credits: 3

CS 581**Topics in Artificial Intelligence**

Covers various advanced topics in AI, including both theory and practice. Content may vary by instructor. Possible topics include: Planning: STRIPs planning; Partial-order planning; Situation calculus; Theorem proving; GraphPlan/SatPlan; Transformational planning; Simulated annealing; Motion planning; Case-based reasoning; Multi-agent coordination; Negotiation planning; Representation and Reasoning: Logical representation; Frame problem; Probabilistic reasoning; Bayesian networks; Game Playing: Minimax search; Evaluation functions; Learning evaluation functions; Markov Decision Processes; Reinforcement learning for games; Developing AI agents; Multi-agent planning.

Prerequisite(s): [(CS 480)]

Lecture: 3 Lab: 0 Credits: 3

CS 582**Computational Robotics**

Covers basic algorithms and techniques used in Computational Robotics, to give the student a good basis for work in this highly relevant field. Topics include: Locomotion, Non-visual sensors and algorithms, Uncertainty modeling, data fusion, State space models, Kalman filtering, Visual sensor, Sampling theory, Image features, Depth reconstruction, Multiple view geometry, Ego-motion, Active vision, Reasoning, Spatial decomposition, Geometric representations, Topological representations, Path planning, Spatial uncertainty, Active control, Pose maintenance, Dead reckoning, Correlation-based localization, Sensorial maps, Task planning and task interference, Multi-agent coordination.

Prerequisite(s): [(CS 430)]

Lecture: 3 Lab: 0 Credits: 3

CS 583**Probabilistic Graphical Models**

This course will cover probabilistic graphical models – powerful and interpretable models for reasoning under uncertainty. The generic families of models such as directed, undirected, and factor graphs as well as specific representations such as hidden Markov models and conditional random fields will be discussed. The discussions will include both the theoretical aspects of representation, learning, and inference, and their applications in many interesting fields such as computer vision, natural language processing, computational biology, and medical diagnosis.

Lecture: 3 Lab: 0 Credits: 3

CS 584**Machine Learning**

Introduce fundamental problems in machine learning. Provide understanding of techniques, mathematical concepts, and algorithms used in machine learning. Provide understanding of the limitations of various machine learning algorithms and the way to evaluate performance of learning algorithms. Topics include introduction, regression, kernel methods, generative learning, discriminative learning, neural networks, support vector machines, graphical models, unsupervised learning, and dimensionality reduction.

Prerequisite(s): [(CS 430)]

Lecture: 3 Lab: 0 Credits: 3

CS 585**Natural Language Processing**

An introduction to the problems of computing with human languages. Parsing. Semantic representations. Text generation. Lexicography. Discourse. Sublanguage studies. Applications to CAI, database interfaces and information retrieval.

Prerequisite(s): [(CS 430)]

Lecture: 3 Lab: 0 Credits: 3

CS 586**Software Systems Architectures**

This course covers the state-of-the-art in architectural design of complex software systems. The course considers commonly-used software system architectures, techniques for designing and implementing these architectures, models and notations for characterizing and reasoning about architectures, and case studies of actual software system architectures.

Prerequisite(s): [(CS 487)]

Lecture: 3 Lab: 0 Credits: 3

CS 587**Software Project Management**

Concepts of software product and process quality. Role of TQM in software project management. Use of metrics, feasibility studies, cost and effort estimates. Discussion of project planning and scheduling. The project team and leadership issues. The Capability Maturity Model: basic tenets and application of process evaluation.

Prerequisite(s): [(CS 487)]

Lecture: 3 Lab: 0 Credits: 3

CS 588**Advanced Software Engineering Development**

Software development process improvement is a major objective of this course. This is achieved through a series of individual programming and process projects. Students learn to plan their projects, apply measurements, estimate size, schedule tasks, and classify defects in order to improve the quality of both their development process and their software products.

Prerequisite(s): [(CS 487)]

Lecture: 3 Lab: 0 Credits: 3

CS 589**Software Testing and Analysis**

Concepts and techniques for testing and analysis of software. Software testing at the unit, subsystem, and system levels. Specification-based testing. Code-based testing. Model-based testing. Methods for test generation and validation. Static and dynamic analysis. Formal methods and verification. Reliability analysis.

Prerequisite(s): [(CS 487)]

Lecture: 3 Lab: 0 Credits: 3

CS 590**Seminar in Computer Science**

Investigation and discussion by faculty and students concentrated on some topic of current interest. May be taken more than once.

Prerequisite: Instructor permission required.

Lecture: 0 Lab: 0 Credits: 3

CS 591**Research and Thesis of Masters Degree**

Instructor permission required.

Credit: Variable

CS 594**Research Problems**

Instructor permission required.

Credit: Variable

CS 595**Topics in Computer Science**

This course will treat a specific topic, varying from semester to semester, in which there is a particular student or staff interest. May be taken more than once.

Credit: Variable

CS 597**Reading and Special Problems**

May be taken more than once. (Credit: Variable) Instructor permission required.

Credit: Variable

CS 612**Topics in Computer Vision**

Covers advanced topics in computer vision to enhance knowledge of students interested in this highly important area. The topics in this course may change between semesters depending on the instructor teaching the course and the current state of the art in this area. Possible topics include: Image based modeling and rendering, Multiple view geometry, Auto-calibration, Object recognition, Motion analysis, Tracking, Perceptual user interfaces, Face and gesture recognition, Active vision.

Prerequisite(s): [(CS 512)]

Lecture: 0 Lab: 0 Credits: 3

CS 630**Advanced Topics in Algorithms**

Theoretical analysis of various types of algorithms. Topics vary, and may include approximation, quantum, on-line, distributed, randomized, and parallel algorithms. Requires CS 430. Instructor permission required.

Prerequisite(s): [(CS 430)]

Lecture: 3 Lab: 0 Credits: 3

CS 642**Advanced Topics in Networking**

Introduction to advanced networking research. A particular focus area will be considered, keeping current with advances in computer networking. Quantitative methods will be emphasized.

Prerequisite(s): [(CS 542)]

Lecture: 3 Lab: 0 Credits: 3

CS 681**Topics in Computational Linguistics**

CS 585 Covers various topics in linguistics as they may be applied to various computational problems in AI, NLP, or IR. The topics in this course may change between semesters depending on the instructor teaching the course and the current state of the art in this area.

Possible topics include: Systemic Functional Linguistics, Clausal structure, Group structure, Complex structure, Cognitive Linguistics, Process semantics.

Prerequisite(s): [(CS 585)]

Lecture: 3 Lab: 0 Credits: 3

CS 689**Advanced Topics in Software Engineering**

Course content is variable and reflects the current trends in software engineering. Instructor permission required.

Lecture: 3 Lab: 0 Credits: 3

CS 691**Research and Thesis Ph.D.**

Instructor permission required.

Credit: Variable

CS 695**Doctoral Seminar**

Doctoral seminar.

Lecture: 0 Lab: 0 Credits: 1

CS 750**Computer Aided Software Engineering**

This course presents the state-of-the-art of computer-aided software engineering technologies. CASE encompasses a collection of automated tools and methods that provide automated support to the software specification, design, development, testing, maintenance, and management of large and complex software systems. Students will develop working understanding of CASE methodologies and tools.

Prerequisite(s): [(CS 487)]

Lecture: 2 Lab: 0 Credits: 2

CS 763**Automated Software Testing**

This course will examine both the state-of-the-art and the state-of-practice in automated software testing on a system level and an unit level. Relevant issues include theoretical foundations of automated testing, automation tools and techniques, empirical studies and industrial experience. Key topics include, but are not limited to: Fundamentals of automated software testing, automated test design, modeling and generation, automated test execution, automated test management, automated test metrics, automated tools, automated feature and regression testing Environments to support cost-effective automated software testing, discussions on the barriers to industrial use of automated testing.

Prerequisite(s): [(CS 487)]

Lecture: 2 Lab: 0 Credits: 2