Computer Science

cs.siu.edu
csinfo@cs.siu.edu

COLLEGE OF SCIENCE

Graduate Faculty:

Bosu, Amiangshu, Assistant Professor, Ph.D., University of Alabama, 2015; 2016. Software engineering, empirical software engineering, code review, software security, android security, malware detection, and mining software repositories.

Carver, Norman F., III, Associate Professor, Ph.D., University of Massachusetts, 1990; 1995. Multi-agent systems, sensor interpretation, machine learning.

Che, Dunren, Professor, Ph.D., Beijing University of Aeronautics and Astronautics, Beijing China, 1994; 2001. Database, data mining, cloud computing, big data management and analytics.

Danhof, K. J., Professor, *Emeritus,* Ph.D., Purdue University, 1969; 1969.

Gupta, Bidyut, Professor, Ph.D., University of Calcutta, 1986; 1988. Distributed systems, fault-tolerant computing, mobile communication, routing algorithms, peer-to-peer networks.

Hexmoor, Henry, Associate Professor, Ph.D., University of Buffalo, 1996; 2006. Artificial intelligence, Multi-agent systems, cognitive science, mobile robotics, knowledge representation and reasoning.

Hou, Wen-Chi, Professor, Ph.D., Case Western Reserve University, 1989; 1989. Statistical databases, query optimization, data stream processing, spatial data structures, XML databases, big data.

Houshmand, Shiva, Assistant Professor, Ph.D., Florida State University, 2015; 2015. Authentication, Usable security, Information security, Computer and Network security and Digital forensics.

Hoxha, Bardh, Assistant Professor, Ph.D., Arizona State University, 2017. Formal Methods, Testing and Verification of Cyber-Physical Systems, Motion Planning for Autonomous Vehicles, and Human-Robot Interaction.

Mark, Abraham M., Professor, *Emeritus,* Ph.D., Cornell University, 1947; 1950.

McGlinn, Robert J., Associate Professor, *Emeritus,* Ph.D., Southern Illinois University Carbondale, 1976; 1981.

Mogharreban, Namdar, Associate Professor, *Emeritus,* Ph.D., Southern Illinois University Carbondale, 1989; 1999.

Mousas, Christos, Assistant Professor, Ph.D., University of Sussex, 2014; 2016. Computer Graphics, Computer Animation, Virtual Reality, Game Development.

Phillips, Nicholas C.K., Associate Professor, *Emeritus,* Ph.D., University of Natal, 1967; 1988.

Rahimi, Shahram, Professor and *Chair*, Ph.D., University of Southern Mississippi, 2002; 2002. Computational intelligence, soft computing, multi-agent systems, distributed systems.

Rekabdar, Banafsheh, Assistant Professor, Ph.D., University of Nevada, 2017. Artificial Intelligence, Machine learning, Deep learning, Data mining, big data analytics, Robotics.

Saeedloei, Neda, Assistant Professor, Ph.D., University of Texas, 2011. Formal methods; model-based design, specification and verification of cyber-physical systems, models of computation, logic in computer science.

Sinha, Koushik, Assistant Professor, Ph.D., Jadavpur University, 2007; 2015. Mobile and wireless sensor networks, cloud computing and social computing, resource allocation and task scheduling.

Wainer, Michael S., Associate Professor, *Emeritus,* Ph.D., University of Alabama at Birmingham, 1987; 1988.

Wright, William. E., Professor, *Emeritus,* D.Sc., Washington University, 1972; 1970.

Zargham, Mehdi. R., Professor, *Emeritus,* Ph.D., Michigan State University, 1983; 1983.

The Department of Computer Science offers a graduate program leading to the Master of Science and Doctor of Philosophy degree in computer science. For admission procedures to these degree programs refer to the Graduate School or department website (cs.siu.edu).

A nonrefundable $65 application fee must be submitted with the Graduate School's online application for Admissions to Graduate Study in Computer Science. Applicants must pay this fee by credit card.

Decisions concerning the admission of students to and retention of students in the graduate program will be made by the department faculty subject to the requirements of the Graduate School.

Master of Science Degree in Computer Science

Admission. The evaluation of applicants for admission is based primarily on the student's academic record with particular attention being given to past performance in relevant undergraduate course work. Applicants are expected to have a substantial background in undergraduate computer science courses covering high level and assembly language programming, data structures, computer organization, logic design as well as discrete mathematics, calculus, and linear algebra. The applicant is expected to have completed course work in the above subject areas prior to admission. Normally, a GPA of at least 3.0/4.0 is required by the Department of Computer Science.

Requirements. A student who has been admitted to the graduate program in Computer Science can meet the requirements for the Master of Science degree by completing 30 hours of graduate credit subject to the following constraints:

1. Students must take six hours of Computer Science coursework from the approved courses for each of the following three categories:
   Computer Science Theory
   Software Development/Engineering
   Computing Systems Technologies
   (This requirement accounts for at least 18 hours of the required 30 hours of total graduate credit.)

   The approved courses for each category are:
   Computer Science Theory:
   408, 437, 438, 447, 449, 451, 455, 510, 533, 553, 555, 586
   Software Development/Engineering:
   407, 412, 420, 435, 485, 487, 520, 585
   Computing Systems Technologies:
   401, 406, 410, 416, 430, 436, 440, 514, 530, 534, 540
   Additional courses may be allowed as appropriate, subject to Graduate Program Director approval.

2. The 30 hours of graduate work must include at least four 500-level CS lecture courses.
3. If a student believes they need to take a course from another academic unit at the University in order to gain specific knowledge for their thesis or project work, they must request approval from the Graduate Program Director prior to registering for such a course. The request must include an explanation of why the course is necessary for their program. Approval will be granted only if the justification is deemed adequate. No more than three hours of credit toward the 30 hour requirement will be given, and such courses must be at the 400- or 500-level only.
4. Students are required to choose either a thesis or non-thesis program:

Thesis Option

A student must complete six credit hours of CS 599, Thesis, in 3 credit hour segments taken for two semesters and 24 credit hours of lecture courses. The student is eligible to take the course CS 598 (must be in industry only). This CS 598 course will be considered equivalent to three credit hours of thesis (subject to the approval of the supervising faculty).

Non-Thesis Option

A student must take 27 credit hours of lecture courses. In addition, the student will take CS 598, Graduate Project, under the supervision of a faculty member.

Doctor of Philosophy Degree in Computer Science

Admission. Subject to meeting the admission requirements of the Graduate School, admission requirements for the Ph.D. in computer science consist of:

1. A master's degree in computer science or a related field with a minimum GPA of 3.25/4.0.
2. Graduate Record Examination (GRE) general test scores. It is recommended that results from the GRE subject area in computer science or a related area be included.
3. In exceptional cases, high achieving students with only bachelor degrees will be admitted to the program. Each student, in addition to the Ph.D. program course requirements, must complete at least 15 semester hours of approved computer science courses including CS 401, CS 420, CS 455, and two 500-level lecture courses, with a minimum accumulated GPA of 3.25/4.0 in those courses. If a specific course, or its equivalent, is already part of the student's academic background, an alternate course will be submitted.

Each applicant is reviewed and evaluated on an individual basis. The evaluation of applicants for admission is based primarily on the student's academic record and area of research interest. Application materials should include evidence of scholarly ability and/or achievement (e.g. awards, scholarships, work experience, recommendation letters, and published research papers). Only those who best meet the research goals and objectives of the doctoral program will be selected for admission.

Requirements. The student must fulfill the requirements for the departmental Qualifying Examination within three years of enrollment in the doctoral program. The Qualifying Examination is organized and administered by the student's academic advisor. The faculty prepares a written test based on at least two areas of concentration related to the student's intended dissertation area. Questions will be drawn from regularly scheduled 400- and 500-level graduate courses at SIU. The grade for the exam will be on a Pass or Fail basis for each subject area. If a student fails to pass any subject area of the written examination, a second chance is given for the failed topic test. Students who fail the Qualifying Examination after two attempts will be dismissed from the Ph.D. program.

To fulfill the course requirements of the Ph.D. program, the student must complete at least 24 credit hours of 400/500-level courses and 24 credit hours of CS 600, Dissertation research, all of which are subject to the following constraints:

1. The course work must include two one-credit hour seminar courses, six credit hours from an approved list of computer science 400/500-level courses, and six elective credit hours of CS 500-level courses.
2. The student must file a request with the Department to appoint a dissertation committee to supervise the remaining doctoral work. This committee will consist of five graduate faculty members, one or two of whom will be from a graduate program outside the Department, one preferably from outside this University. The student's dissertation advisor will serve as the chair of this committee.
3. Each student should complete a course of study as determined by the student's dissertation committee.
4. The course of study must include a minimum of six credit hours of 400/500-level courses from academic departments other than computer science. These courses must be selected from a list approved by the Department.
5. Having passed the qualifying exams and after completion of most of the course requirements, a student will begin working on a dissertation proposal. The next step will be a Preliminary Examination consisting of an oral test on the student's proposed research topic. The student will pass the Preliminary Examination only if the members of the committee, with at most one exception, judge the performance of the student's oral examination to be satisfactory. In the event the student's performance is unsatisfactory, the committee will reschedule the exam for a later time. A student who fails the reexamination will be dismissed from the Ph.D. program.
6. A student will be officially admitted to candidacy for the Ph.D. degree after passing the Preliminary Exam and upon completion of all course work. The student must then complete 24 credit hours of dissertation credit, restricted to nine hours per semester. When the research is complete and the dissertation is written, a final oral examination will take place to determine if the research conducted is worthy of the Ph.D. degree. The dissertation must conform to high literary and scholastic standards and comply with all the relevant requirements of the Graduate School. The dissertation must represent original research of good quality. From the dissertation, the candidate should publish (or have accepted for publication) a minimum of two articles in peer-reviewed journals. The candidate must be listed as the primary author of at least one of these journal articles.

7. Each candidate must pass a final oral exam over the candidate's dissertation, conducted by the candidate's dissertation committee. The dissertation will be accepted provided the dissertation advisor and at least three of the other four members of the committee so agree
8. Degree requirements, graduation, and time limits are subject to the general guidelines of the Graduate School.

Courses (CS)

CS 401-3 Computer Architecture. Review of logical circuit design. Hardware description languages. Algorithms for high-speed addition, multiplication and division. Pipelined arithmetic. Implementation and control issues using PLA`s and microprogramming control. Cache and main memory design. Input/Output. Introduction to interconnection networks and multiprocessor organization. Prerequisite: CS 320 with a grade of C or better or graduate standing.

CS 404-3 Autonomous Mobile Robots. This course is a comprehensive introduction to modern robotics with an emphasis on autonomous mobile robotics. Fundamentals of sensors and actuators as well as algorithms for top level control are discussed. Multi-robotics and human-robot interaction issues are explored. A group project is an integral part of this course. Prerequisite: CS 330 with a grade of C or better or graduate standing. CS Fee: $125.

CS 406-3 Basic Linux System Administration. This course will be an introduction to the administration of Linux systems, with emphasis on security for networked systems. Topics to be covered include: installation and configuration of Linux distributions, typical maintenance activities, and security measures for networked systems. Students will have access to lab machines for hands on practice. Prerequisite: CS 306 with a grade of C or better or graduate standing.

CS 407-3 Advanced Linux/UNIX Programming. This course builds on the knowledge gained in CS 306, to prepare students to do advanced development on Linux/UNIX platforms. The topics studied are critical for achieving high performance in large-scale, high-load networked software systems. These topics include development techniques such as profiling, concurrent programming and synchronization, network programming for high-load servers, advanced I/O alternatives, and IPC such as shared memory. The course will involve the study of code from Open Source projects like Apache and Nginx. The focus will be on the C language, but other languages will also be considered. Students must complete a significant network software project. Prerequisites: CS 306 and CS 335, with grades of C or better, or graduate standing with C language and Linux system programming experience.

CS 408-3 Applied Cryptography. This course is a comprehensive introduction to modern cryptography, with an emphasis on the application and implementation of various techniques for achieving message confidentiality, integrity, authentication and non-repudiation. Applications to Internet security and electronic commerce will be discussed. All background mathematics will be covered in the course. Prerequisite: CS 330 with a grade of C or better and MATH 221 or graduate standing.

CS 410-3 Computer Security. A broad overview of the principles, mechanisms, and implementations of computer security. Topics include cryptography, access control, software security and malicious code, trusted systems, network security and electronic commerce, audit and monitoring, risk management and disaster recovery, military security and information warfare, physical security, privacy and copyrights, and legal issues. Prerequisite: CS 306 with a grade of C or better or graduate standing.

CS 412-3 Programming Distributed Applications. This course uses advanced features of the Java programming language to develop networked, distributed, and web-based applications. Topics covered include, but are not limited to, sockets, datagrams, the Java security model, threads, multi-tier architectures, Java RMI, Java database connectivity, and Java-based mobile agents. Prerequisite: CS 306 with a grade of C or better or graduate standing.

CS 416-3 Compiler Construction. Introduction to compiler construction. Design of a simple complete compiler, including lexical analysis, syntactical analysis, type checking, and code generation. Prerequisite: CS 306 and 311 each with a grade of C or better or graduate standing.

CS 420-3 Distributed Systems. A top-down approach addressing the issues to be resolved in the design of distributed systems. Concepts and existing approaches are described using a variety of methods including case studies, abstract models, algorithms and implementation exercises. Prerequisite: CS 335 or graduate standing.

CS 425-3 Principles of Virtualization and Cloud Computing. Cloud Computing (CC) represents a recent major strategic shift in computing and Information Technology. This course explores fundamental principles, foundational technologies, architecture, design, and business values of CC. Understanding will be reinforced through multiple angles including: analysis of real world case studies, hands-on projects and in-depth study of research developments. Prerequisites: CS 330 with a grade of C or better or graduate standing.

CS 430-3 Database Systems. The course concentrates on the relational model, database design, and database programming. Topics include relational model, relational algebra, SQL, constraints and integrity, transaction support, concurrency control, database design, normalization, backup, recovery, and security. A comprehensive product-like project is an integral part of the course. Prerequisite: CS 330 with a grade of C or better or graduate standing.

CS 434-3 Learning From Data. An introduction to classical machine learning theory and practical techniques. Topics to be covered include computational learning theory (VC theory), linear classification and regression models, SVMs and kernel methods, decision trees, the bias-variance tradeoff, overfitting, and regularization. Prerequisites: CS 330 with a grade of C or better or graduate standing.

CS 435-3 Software Engineering. Principles, practices and methodology for development of large software systems. Object-oriented principles, design notations, design patterns and coping with changing requirements in the software process. Experiences with modern development tools and methodologies. A team project is an integral part of this course. Prerequisite: CS 330 with a grade of C or better or graduate standing; CS 306 with a grade of C or better recommended.

CS 436-3 Artificial Intelligence I. Search and heuristics, problem reduction. Predicate calculus, automated theorem

proving. Knowledge representation. Applications of artificial intelligence. Parallel processing in artificial intelligence. Prerequisite: CS 311 and 330 each with a grade of C or better or graduate standing.

CS 437-3 Machine Learning and Soft Computing. An introduction to the field of machine learning and soft computing. It covers rule-based expert systems, fuzzy expert systems, artificial neural networks, evolutionary computation, and hybrid systems. Students will develop rule-based expert systems, design a fuzzy system, explore artificial neural networks, and implement genetic algorithms. Prerequisite: CS 330 with a grade of C or better or graduate standing.

CS 438-3 Bioinformatics Algorithms. This course is an introductory course on bioinformatics algorithms and the computational ideas that have driven them. The course includes discussions of different techniques that can be used to solve a large number of practical problems in biology. Prerequisite: CS 330 with a grade of C or better or graduate standing.

CS 440-3 Computer Networks. Design and analysis of computer communication networks. Topics to be covered include queuing systems, data transmission, data link protocols, topological design, routing, flow control, security and privacy, and network performance evaluation. Prerequisite: CS 330 with a grade of C or better or graduate standing; CS 306 recommended.

CS 441-3 Mobile and Wireless Computing. Concepts of mobile and wireless systems are presented. These concepts include, but are not limited to, Routing and Medium Access for Mobile Ad hoc and Wireless Sensor Networks, Mobile IP, Wireless LAN and IEEE 802.11. Hands-on group lab experience is an integral component in the course. Prerequisite: CS 330 with a grade of C or better, or graduate standing or consent of the instructor.

CS 447-3 Introduction to Graph Theory. (Same as MATH 447) Graph theory is an area of mathematics which is fundamental to future problems such as computer security, parallel processing, the structure of the World Wide Web, traffic flow and scheduling problems. It also plays an increasingly important role within computer science. Topics include: trees, coverings, planarity, colorability, digraphs, depth-first and breadth-first searches. Prerequisite: MATH 349 with C or better.

CS 449-3 Introduction to Combinatorics. (Same as MATH 449) This course will introduce the student to various basic topics in combinatorics that are widely used throughout applicable mathematics. Possible topics include: elementary counting techniques, pigeonhole principle, multinomial principle, inclusion and exclusion, recurrence relations, generating functions, partitions, designs, graphs, finite geometry, codes and cryptography. Prerequisite: MATH 349 with C or better.

CS 451-3 Theory of Computing. The fundamental concepts of the theory of computation including finite state acceptors, formal grammars, Turing machines, and recursive functions. The relationship between grammars and machines with emphasis on regular expressions and context-free languages. Prerequisite: CS 311 and 330 each with a grade of C or better or graduate standing.

CS 455-3 Advanced Algorithm Design and Analysis. An in-depth treatment of the design, analysis and complexity of algorithms with an emphasis on problem analysis and design techniques. Prerequisites: CS 330 with a grade of C or better or graduate standing.

CS 471-3 Optimization Techniques. (Same as MATH 471) Introduction to algorithms for finding extreme values of nonlinear multivariable functions with or without constraints. Topics include: convex sets and functions; the arithmetic-geometric mean inequality; Taylor`s theorem for multivariable functions; positive definite, negative definite, and indefinite matrices; iterative methods for unconstrained optimization. Prerequisite: MATH 221 and MATH 250 with C or better.

CS 472-3 Linear Programming. (Same as MATH 472) Introduction to finding extreme values of linear functionals subject to linear constraints. Topics include: recognition, formulation, and solution of real problems via the simplex algorithm; development of the simplex algorithm; artificial variables; the dual problem and duality theorem; complementary slackness; sensitivity analysis; and selected applications of linear programming. Prerequisite: MATH 221 with C or better.

CS 475-3 Numerical Analysis I. (Same as MATH 475) Introduction to theory & techniques for computation with digital computers. Topics include: solution of nonlinear equations; interpolation & approximation; solution of systems of linear equations; numerical integration. Students will use MATLAB to study the numerical performance of the algorithms introduced in the course. Prerequisites: MATH 221 and MATH 250 with C or better.

CS 476-3 Numerical Analysis II. (Same as MATH 476) Continuation of CS 475. Topics include: solution of ordinary differential equations; computation of eigenvalues and eigenvectors; and solution of partial differential equations. Students will use MATLAB to study the numerical performance of the algorithms introduced in the course. Prerequisites: MATH 305 and MATH 475 with C or better.

CS 480-3 Computational Statistics II. This course utilizes computational and graphical approaches to solve statistical problems. A comprehensive coverage on modern and classical methods of statistical computing will be given. Case studies in various disciplines such as science, engineering and education will be discussed. Various topics such as numerical integration and simulation, optimization and maximum likelihood estimation, density estimation and smoothing as well as re-sampling will be presented. Students will be able to create graphical and numerical display based on their data analysis results using R programming language. Prerequisite: MATH 250 and CS 306 or CS 330 with a grade of C or better or graduate standing.

CS 484-3 User Interface Design and Development. Problems and processes in the design of highly usable systems. Understanding stakeholders, requirements, tasks, prototyping, evaluation, guidelines and design process and heuristics. Interactive software concepts and implementation considerations. A group project is an integral part of this course. Prerequisite: CS 306 with a grade of C or better or graduate standing.

CS 485-3 Computer Graphics. Principles and techniques of computer graphics. Interactive graphics software development using a modern graphics standard. Topics include: primitives, transforms, clipping, modeling, viewing, rendering, texture, animation and ray tracing. A group project is an integral part of this course. Prerequisite: CS 306 with a grade of C or better or graduate standing; MATH 150 and 221 are recommended.

CS 487-3 Software Aspects of Game Development. This course focuses on software implementation and development

aspects of game production including: software process, system architecture, frameworks, entity management and interaction design, game design, production and business issues as well as technical foundations in graphics modeling and rendering, collision detection, physics, artificial intelligence, and multiplayer techniques. Prerequisite: CS 330 with a grade of C or better or graduate standing.

CS 491-1 to 6 (1 to 3 per topic) Special Topics. Selected advanced topics from the various fields of computer science. Special approval needed from the instructor.

CS 492-1 to 6 (1 to 3 per semester) Special Problems. Individual projects involving independent work. Special approval needed from the department.

CS 493-1 to 4 Seminar. Supervised study. Preparation and presentation of reports. Special approval needed from the instructor.

CS 501-3 Advanced Computer Architecture. Hardware and software elements of multiprocessors, multicomputers, pipeline and array machines, data flow architecture and other state-of-the-art architectures. Design principles related to machine structures, interconnection networks, control software and hardware, data storage and access. Prerequisite: CS 401.

CS 503-3 Fault-Tolerant Computing Systems. An introduction to different aspects of fault-tolerance in computing systems. Concurrent checking techniques. Redundancy techniques. Evaluation methods. System-level diagnosis and fault-tolerant VLSI architectures. Prerequisite: CS 401.

CS 510-3 Wireless and Network Security. Advanced security concepts of distributed systems and wireless networks are presented. Topics include IEEE 802.11 security, Wireless Encryption and Authentication, Key Management in Networks, Distributed Denial of Service Attacks, Routing Security, Intrusion Detection and Mobile Code Security. Prerequisite: CS 410 with a grade of C or better or consent of the instructor.

CS 511-3 Formal Specification of Programming Languages. A survey of modeling techniques and Meta languages for the formal specification of the syntax and semantics of high-level programming languages. Prerequisite: CS 311.

CS 512-3 Declarative Programming. An advanced level course on nonprocedural programming with emphasis on logic programming, pure functional programming, and the characteristics of the declarative style common to these two paradigms. Topics include logic programming, functional programming, implementation consideration for each along with current research topics in the areas. Prerequisite: CS 311.

CS 514-3 Advanced Operating Systems. Rigorous treatment of advanced topics in operating systems. Multiprocessors and distributed operating systems. Highly concurrent machines. Performance analysis of memory management and scheduling algorithms. Recovery techniques in distributed computation. Security in operating systems. Prerequisite: CS 335 with a grade of C or better.

CS 516-3 Advanced Compilers. A continuation of 416 including advanced topics in lexical and syntax analysis, error recovery, sematic analysis, code optimization and compiler compilers. Prerequisite: CS 416.

CS 520-3 Advanced Topics in Parallel & Distributed Computing. An advanced treatment of parallel and distributed computing; review of hardware and software considerations for parallel computation; development and analysis of parallel algorithms (with particular attention to the communication and synchronization costs associated with parallel algorithms); effect of granularity on performance; a comparison of the parallel and distributed programming paradigms including a detailed study of the central features of each approach; software systems for distributed computing including exposure to one or more distributed programming environments; the direction of parallel computing as suggested by recent, high level parallel languages; parallelizing serial programs; parallelizing compilers; future directions of parallel and distributed computing systems. The course will include a student project. Prerequisite: CS 420.

CS 530-3 Advanced Database Systems. A detailed treatment of advanced topics in data base systems including, but not limited or restricted to, relational database theory, query optimization, recovery techniques, concurrency control, distributed database systems, security and integrity and database machines. Prerequisite: CS 430.

CS 532-3 to 6 Topics in Information Systems. A detailed study of two or three topics relevant to information systems. Topics may include but are not limited to sorting, searching, information retrieval and automatic text processing, database security and encryption, distributed databases and data communication. Prerequisite: CS 430. Special approval needed from the instructor.

CS 533-3 Data Mining and Big Data Analysis. This course provides a series of comprehensive and in-depth lectures on the core techniques in data mining and knowledge discovery; addresses the unique issues of big data; and discusses potential applications of data mining particularly on big data analysis. Major topics include: data preparation, association mining, classification (and prediction), clustering, characteristics and challenges of big data, and strategies of big data mining and analysis. Prerequisites: CS 330 and CS 430 with grades of C or better or consent of instructor.

CS 534-3 Big Data Management and Analytics. This course provides comprehensive and in-depth discussions of big data management and analytics. Main subjects include computation and programming models, management and analytics algorithms, and platforms/frameworks especially designed for big data. The objective of this course is to equip students with the ability to understand, use, and build big data management and analytics systems or tools. Prerequisites: CS 430 with a grade of C or better or graduate standing.

CS 536-3 Artificial Intelligence II. Theorem proving, the Resolution Principle, strategies, and achievements. Program verification. Natural language processing. Other selected topics. Prerequisite: CS 436.

CS 537-3 Advanced Topics in Expert Systems. This course is designed to provide students with advanced topics in expert systems theory. Topics covered include: knowledge representation, methods of inference, reasoning under uncertainty, and inexact reasoning (fuzzy logic). A practical introduction to expert systems programming serves to reinforce and clarify the theoretical concepts. Prerequisite: CS 330 or consent of instructor.

CS 538-3 Game Theory in Networks. Game theoretic concepts apply whenever actions of several players are interdependent. This course will provide an introduction to classic game theory and strategic thinking including dominance, Nash equilibrium,

and stability. Social choice, social learning, and online mechanism design are then discussed. We will examine how game theoretic concepts can be used in developing reasoning strategies, i.e., algorithms. Application of game theoretic framework to telecommunication and human networks is an integral part of this course. Restricted to graduate standing or consent of instructor.

CS 539-3 Agents and Multiagent Systems. This is an advanced treatment of fundamental concepts in the design of intelligent autonomous agents and agent systems. Classic agent theories, architectures, algorithms, and languages are discussed. An agent-based project is an integral part of this course. Restricted to graduate standing or consent of instructor.

CS 540-3 Advanced Computer Networks. Topics include routing protocols used in internet; data compression techniques; telecommunication systems - its services, architecture and protocols; high speed networks; routing protocols in mobile ad-hoc networks; and a detailed performance analysis of different window flow control and congestion control mechanisms using queuing theory. Prerequisite: CS 440 with a grade of C or better, or consent of the instructor.

CS 553-3 Formal Languages and Automata. The Chomsky hierarchy of formal grammars and the corresponding classes of automata. Turing machines and basic concepts of computability. Recursive and recursively enumerable languages. Closure properties. Undecidable problems about Turing machines and context-free languages. Deterministic context-free languages and the construction of LR parsers. Prerequisite: CS 451.

CS 555-3 Computability and Complexity. Turing machines and other models of computation. Computable functions. Church`s thesis. Solvable and unsolvable problems. Introduction to complexity theory including the classes P and NP. Polynomial time approximation algorithms for NP-complete problems. Prerequisite: CS 451.

CS 572-1 to 12 Advanced Topics in Numerical Analysis. (Same as MATH 572) Selected advanced topics in Numerical Analysis chosen from such areas as: approximation theory; spline theory; special functions; wavelets; numerical solution of initial value problems; numerical solution of boundary value problems; numerical linear algebra; numerical methods of optimization; and functional analytic methods. Special approval needed from the instructor.

CS 585-3 Advanced Topics in Computer Graphics. Study of computer graphics for realistic image synthesis. Object modeling and associated date structures. Advanced rendering techniques such as raytracing and radiosity. Efficiency considerations. Image composition and compression. Current advances and research problems in realistic computer graphics. Prerequisite: CS 485.

CS 586-3 Pattern Recognition. An introduction to the area of pattern recognition and data science. This course will cover basic and advanced theories, algorithms, and practical solutions of statistical pattern recognition. It covers bayesian learning, parametric and non-parametric learning, data clustering, component analysis, boosting techniques, sequential data, reinforcement learning, and deep learning with neural networks.

CS 590-1 to 6 Readings. Supervised readings in selected subjects. Graded S/U only. Special approval needed from the instructor and department.

CS 591-1 to 9 (1 to 3 per topic) Special Topics. Selected advanced topics from the various fields of computer science. Special approval needed from the instructor.

CS 593-1 to 4 Seminar. Preparation and presentation of reports. Graded S/U only. Special approval needed from the instructor.

CS 598-3 to 9 Graduate Project. A practical exercise in the design, implementation, documentation and deployment of a project. A project may be completed through internship, work/study, or a supervised project. For Ph.D. students only, an internship could include face-to-face or online teaching.

CS 599-3 to 9 Thesis. Special approval needed from the instructor and department.

CS 600-1 to 24 (1 to 9 per semester) Doctoral Dissertation. Dissertation research. Hours and credit to be arranged by the student`s academic advisor. Graded S/U only. Restricted to admission to Ph.D. in computer science program.

CS 601-1 per semester Continuing Enrollment. For those graduate students who have not finished their degree programs and who are in the process of working on their dissertation, thesis, or graduate project. The student must have completed a minimum of 24 hours of dissertation research, or the minimum thesis or graduate project hours before being eligible to register for this course. Concurrent enrollment in any other course is not permitted. Graded S/U or DEF only.