College of Computer Science and Mathematics

DEPARTMENT OF COMPUTER SCIENCE

FACULTY

- Greg Guthrie, Ph.D., Dean of the College of Computer Science and Mathematics, Professor of Computer Science
- Keith Levi, Ph.D., Chair, Associate Professor of Computer Science
- Mark Rainbow, M.S.E.E., Associate Chair, Assistant Professor of Computer Science
- Bruce Lester, Ph.D., Professor of Computer Science
- Paul Corazza, Ph.D., Associate Professor of Computer Science
- Ralph Bunker, Ph.D., Assistant Professor of Computer Science
- Clyde Ruby, Ph.D., Assistant Professor of Computer Science
- Sunil Sharma, Ph.D., Assistant Professor of Computer Science
- Peter Just, M.S., Assistant Professor of Computer Science
- Joe Lerman, M.S., Instructor of Computer Science
- Payman Salek, M.S., Instructor of Computer Science
- Solomon Seifu, M.S., Instructor of Computer Science
- Ziv Soferman, Ph.D., Adjunct Associate Professor of Computer Science
- Ali Arsanjani, Ph.D., Adjunct Assistant Professor of Computer Science and Management
- Helmuth Trefftz, Ph.D., Visiting Associate Professor of Computer Science

INTRODUCTION

With the rapid advances in science and technology during the last few decades, computing systems have risen to become the key technology that supports and expands almost every area of life, from education and research to commerce and entertainment. With the recent growth of networking systems and the global Internet system connecting millions of people and almost every educational, research, and business institution in the world, computing has become the most powerful and pervasive aspect of modern technology and a vital element of success in almost every area of life.

Today we live in an information-based society. Fundamental knowledge of how computers and computing systems work is a vital part of modern life. The universal role

of computing and the great power that it brings to all areas of life is based on the ability of computing systems to represent and reason about the knowledge which is at the basis of any area of application.

Computer science is the study of these structures and dynamics of information, and their expression into progress and machines. It creates a new and exciting area that merges aspects of mathematics and electronics to form a new discipline of software and computing systems. This allows one to describe abstract concepts or knowledge from any area of interest, and then create powerful systems that produce concrete results — the flight of a satellite, a computer graphics system for movies, scientific computation, management information systems, or desktop word processing.

With such broad areas of application, a computer scientist must have a strong background in both the foundations of knowledge on which these systems are organized, and the principles which are used to create and apply computing to all of these diverse areas of life. Clearly, a computing professional enjoys the ability to work in one of the most exciting and leading areas of technology today and one of the most important areas for the future.

Our computer science programs prepare graduates for success in this field by providing comprehensive knowledge of the discipline and the ability to think clearly and precisely.

Programs Offered

- B.S. in Computer Science
- Minor in Computer Science
- M.S. in Computer Science offered in three formats:

1) a one-year program full time on campus for students with a bachelor's degree in computer science.
2) a three-year on-campus internship program for students with a bachelor's degree in computer science. Students in this program enroll in practicum and directed study courses for two years and are placed in curricular practical training work assignments at the University. The third year is full-time course work.
3) a two-year cooperative program for students with a bachelor's degree in computer science and at least two years of relevant work experience. Students in this program take one year of full-time course work at the University (or through Distance Education) and one year of directed study through a cooperative job placement. (Note: Most costs for this program are covered through internships in American information technology companies.)

- Post-Graduate Certificate in Computer Science
- Specialization in M.S. in Computer Science

SPECIAL FEATURES

- Our programs develop outstanding computer professionals. Graduates are well prepared for careers in business, government, education, or research. Students become thoroughly grounded in programming languages, computer architecture, computer systems, and theory of computation. In addition, they gain experience in applied computer science areas such as computer graphics, compilers, databases, and networking.
- Our students are enjoying notable professional success in industry and education, including Microsoft, IBM, AT&T Bell Labs, Cisco Systems, First Data Corp., Caterpillar, SITA, Bluestem Systems, Google, Commerce Clearing House, Amazon, Marathon Photo, LHS Communications Systems, Software Artisans, and various universities.
- Students develop the essentials for success in the computer science profession, and all areas of life — problem-solving ability, logical thinking, creativity, broad comprehension, and fine focus of attention.
- Students gain experience with the most advanced operating systems and computer environments including Microsoft Windows and Linux.
- Students study the unifying theory of programming languages and explore a variety of modern languages and approaches to programming in various classes, for example, Java and C# (for enterprise and large-scale systems), Python and Ruby (for Web development) and "ML" (for research in the functional approach to programming). Other specialized languages are taught as needed.
- Our faculty use an effective teaching approach that creates a learning environment of ease and enjoyment without the stress and strain that commonly accompany a rigorous discipline.
- Students study the basic principles underlying all computer hardware, and examine principles that have given rise to the most recent advances in high-performance and super computing systems, including *networked*, *parallel*, *distributed*, and *highly concurrent* approaches. Each of these systems uses many computers in combination to solve a large computational task, but they differ in their scope and approach.
- The Department of Computer Science has several very well equipped computing laboratories, which provide Internet access, as well as the departmental network, and campus network. A variety of servers provide support for classes, development, and research activities. Students can also access a wide variety of resources, including scanners, printers, and other campus services including the library online catalogue and materials.

- High-speed campus and Internet access is provided to student housing, all student labs, and several other access places around campus.
- Occasional field trips and guest lectures by successful computer professionals are offered to provide students with the latest developments in computer science and their practical applications in science and industry.
- The electronic computer is amazingly powerful, and yet is limited compared to the computing ability of the 100-billion neuron parallel processing capability of the human brain. This vast capability of the brain physiology is directly cultured through the University's curriculum, so that graduates not only master computer science, but also grow in the ability to spontaneously operate from the total potential of their own brain physiology and make right decisions without mistakes.

DEPARTMENTAL REQUIREMENTS

Entrance Requirements for the Computer Science Major or Minor
Before entering the computer science major or minor, students must successfully complete the course Intermediate Algebra (MATH 153) or its equivalent.

Graduation Requirements for the Bachelor of Science Degree in Computer Science

To graduate with a B.S. in Computer Science, students must successfully complete all general requirements for the bachelor's degree. (Please refer to "Degree Requirements" in "Academic Policies.") As part of these requirements, students must complete 84 credits of course work as listed below. In addition, students must have a minimum 2.5 cumulative grade point average in all computer science courses.

The following required courses:
• CS 201 Computer Programming 1
• CS 203 Computer Programming 2
• CS 220 Data Structures
• CS 222 Data and File Structures
• CS 262 Digital Logic and Computer Organization
• CS 362 Computer Architecture
• MATH 272 Discrete Mathematics
• MATH 281 Calculus 1
• MATH 282 Calculus 2
• MATH 283 Calculus 3
• MATH 286 Linear Algebra 1
• MATH 351 Probability
plus additional credits of computer science courses 300 or above
plus 8 credits of course work in management to equal 84 credits

Requirements for the Minor in Computer Science

To graduate with a minor in computer science, students must complete the following required courses:

- CS 201 Computer Programming 1
- CS 203 Computer Programming 2
- CS 220 Data Structures
- CS 222 Data and File Structures

plus additional credits of computer science courses to equal 28 credits

Entrance Requirements for the Master of Science Degree in Computer Science

To be admitted to the M.S. in Computer Science program, students must hold a bachelor's degree with an undergraduate grade point average of at least 3.0 ("B") and submit scores from the Graduate Record Examination (GRE). In addition, students must have a background in computer science corresponding to the following courses:

- CS 201 Computer Programming 1
- CS 203 Computer Programming 2
- CS 220 Data Structures
- CS 222 Data and File Structures
- CS 310 Systems Programming
- CS 350 Programming Languages
- CS 262 Digital Logic and Computer Organization
- CS 362 Computer Architecture
- MATH 272 Discrete Mathematics

Students without this background can take the needed course work at the beginning of the program, thus increasing the length of the program up to one year. In this case, the grade for the undergraduate prerequisite course work will not be included in the GPA for the Master of Science program.

Four additional mathematics courses are also required for admission:

- Calculus 1 (MATH 281)
- Calculus 2 (MATH 282)
- Linear Algebra I (MATH 286)
- Probability (MATH 351)

Students lacking one of these mathematics courses may be accepted with the understanding that this deficiency will be made up in addition to their regular program of study.

This required background in mathematics and computer science could be acquired through course work at the University or elsewhere, or through equivalent professional work experience.

Transfer credit for graduate courses taken at other qualified universities are limited to a maximum of two courses (8 credits). Additional graduate study can be applied to waive specific course requirements, but not to reduce the number of credits required to graduate.

Graduation Requirements for the Master of Science Degree in Computer Science

To graduate with an M.S. in Computer Science, students must successfully complete all requirements for the master's degree. (Please refer to "Degree Requirements" in "Academic Policies.") Program requirements are:

1) 40 credits of computer science courses at the 400 level or above.
2) At least one of the following must be completed with a grade of "B" or better:
   • CS 435 Algorithms
   • CS 505 Advanced Programming Languages.
3) Two courses (8 credits) must be computer science courses at the 500 level that have been completed with a grade of "B" or higher.
4) The cumulative grade point average for Computer Science courses at the 400 level and above must be at least "B" (GPA of 3.0) or higher. In addition, grades lower than a B are assigned low-grade points " (i.e., B- is 1, C+ is 2, etc). No more than a total of 4 such low-grade points will be allowed in the 40 credits of computer science course work required for graduation.
5) If the master's thesis option is selected by the student and approved by the faculty, then Master's Thesis Research (CS 588) with an oral defense may be used to satisfy up to 8 credits.
6) If, upon admission to the program, the student lacks one of the required mathematics courses, it can be taken to satisfy 4 of the 12 credits of additional computer science course work, if approved by the department.

Entrance Requirements for the Master of Science Degree in Computer Science, Internship Program

Entrance requirements for this program are the same as for the M.S. program listed above. Students who have some deficiencies in these entrance requirements may be provisionally admitted to the program and allowed to make up these deficiencies as part

of the directed study program during the first year of the program or in an additional year of full-time study.

Graduation Requirements for the Master of Science Degree in Computer Science, Internship Program

To graduate with an M.S. in Computer Science, Internship Program, students must successfully complete all general requirements for the master's degree. (Please refer to "Degree Requirements" in "Academic Policies.") In addition, the following 66 credits of course work are required:

- 36 credits of course work corresponding to the standard M.S. program listed above
- 10 credits of practicum course work:
    - CS 576 Practicum in Software Development II
    - CS 591–596
- 20 credits of directed study course work

NOTE: The Forest Academy requirement for this program is either FOR 500 or FOR 501.

Entrance Requirements for the Master of Science Degree in Computer Science, Cooperative Program

Entrance requirements for this program are the same as for the standard M.S. program listed above.

Graduation Requirements for the Master of Science Degree in Computer Science, Cooperative Program

To graduate with an M.S. in Computer Science — Track III, Cooperative Program, students must successfully complete all requirements for the master's degree. (Please refer to "Degree Requirements" in "Academic Policies.") Program requirements are the same as for the M.S. in Computer Science standard program with the following modifications: 44 credits of instruction are required, including,

- 36 credits of course work corresponding to the standard M.S. program listed above
- 8 credits of Practicum (CS 575–CS 579).

NOTE: The Forest Academy requirement for this program is either FOR 500 or FOR 501 in the first semester plus one two-week Forest Academy course (FOR 411–499) for each semester enrolled on the standard schedule.

Entrance Requirements for Post-Graduate Certificate in Computer Science

The Post-Graduate Certificate is offered to students who have graduated from the M.S. in Computer Science program (or have completed all of the graduation requirements). This includes any student who has completed all academic requirements for the M.S. in Computer Science degree (3.0 CS GPA or above) and is within 6 payments of completing their loan (for MUM students only).

Graduation Requirements for Post-Graduate Certificate in Computer Science

There is a 4-week on-campus requirement to this program, followed by two distance education courses. The program is one year in length and Curricular Practical Training (CPT) will be authorized for that length of time.

Students must complete three courses in a specific area of Computer Science. The purpose of a specialization is to prepare a student to be a leader in some specific area of the computing field. Specialization tracks include both foundational and application areas.

The program will begin very shortly after graduation requirements are completed. Four weeks of full-time on-campus studies must be completed in the first 3 months of the starting date of the program. Two distance-education courses must be taken immediately after the on-campus portion of the program.

8 credits of CPT required

A new I-20 will be issued with new program details and will be good for 12 months from starting date.

Entrance Requirements for Specialization in M.S. in Computer Science

Students can choose to extend their M.S. in Computer Science degree by taking the Specialization option. The purpose of a specialization is to prepare a student to be a leader in some specific area of the computing field. Specialization tracks include both foundational and application areas. Students will graduate with an M.S. in Computer Science with a Specialization in a particular area. The graduation will be one year later than the normal M.S. in Computer Science students.

Students must have completed all academic requirements for the Maharishi University of Management M.S. in Computer Science degree (3.0 CS GPA). Students must not have had any extensions to their I-20 program end date.

Graduation Requirements for Specialization in M.S. in Computer Science

This program includes three additional courses taken through distance education, while students continue with the Curricular Practical Training part of their program. The program will begin once all academic requirements for the M.S. in C.S. are completed. Three distance-education courses must be taken consecutively.

8 credits of CPT required

I-20 will be extended for 12 months from starting date.

Specialization or Certificate Areas of Instruction Include:

- Computer Systems (Operating Systems, Parallel Programming, Computer Security)
- Programming Languages (Compilers, Advanced Programming Languages, Advanced Software Development, Parallel Programming)
- Network and Web Computing (Distributed Computing, Distribute Architecture, Parallel Programming, Networks)

COURSES

Undergraduate Courses

CS 101 Nature's Cosmic Computing: Harnessing the Organizing Power of Knowledge
This course investigates the most fundamental knowledge at the basis of all computing and modern computer technology, and how it is connected to principles of Maharishi Vedic Science. We will look at the structure of computing itself, of computer science, and of the wide range of computing applications that are primary to all areas of professions and life today. *Prerequisite:* STC 108/109 (4 credits)

CS 200 Introduction to Computer Science: Creating Games and Animated Stories
This course uses 3-D computer animation technology to introduce computer-programming concepts in a lively and creative setting. Students explore creative story telling and animated games while learning object-oriented programming techniques. Topics include the principles of programming and game design techniques. *Prerequisite:* STC 108/109 (4 credits)

CS 201 Computer Programming 1: The Language of Computing — Expressing the Intelligence that Guides Computation
This first upper-division course in computer science presents the basic principles of computer programming, with emphasis on developing practical programming skills

through laboratory assignments. Topics include formulation of algorithms, top-down design, basic control structures, data types, functions, and subroutines. (4 credits) *Prerequisite:* MATH 153

CS 203 Computer Programming 2: Greater Knowledge and Expression in Programming Languages
Students work in teams on a programming project to practice their knowledge of programming and developing good programming practices. Topics include structured data types, recursion, pointers, and issues of program design, structure, and correctness. (4 credits) *Prerequisite:* CS 201

CS 220 Data Structures: Fundamental Structures of Information at the Basis of All Computation
Students use computer programming laboratory problems to apply the principles of data structure organization in a practical environment and develop advanced programming skills. The organizing power of knowledge is found to be the source of order in computer data structures. Topics include abstract data types, internal representation of data, stacks, queues, linked lists, sparse arrays, hash coding, searching and sorting algorithms, dynamic storage allocation, and computing time of programs. (4 credits) *Prerequisite:* MATH 162, CS 203

CS 222 Data and File Structures: Information Structures to Represent Larger Systems
Students continue the study of high-level data organization techniques. Topics include representations and algorithms for trees and graphs; file organization techniques; sequential, direct and indexed files; B-trees; and inverted and multi-list files. (4 credits) *Prerequisite:* CS 220

CS 262 Computer Organization and Digital Logic: The Physiology at the Basis of All Computers — The Logical and Physical Structures of Digital Computation
This course presents the internal structure of a computer, an introduction to assembly language, and the design of digital logic circuits and their use in structuring the various functional components of a computer, such as the memory and central processing unit. Topics include machine organization, machine language, assembly language, logic gates, flip-flops, decoders, multiplexers, registers, combinatorial logic, and sequential circuits. (4 credits) *Prerequisite:* MATH 153

CS 272 Discrete Structures: Models and Mathematics of the Structures of Natural Law at the Basis of Computation
Discrete mathematics is becoming increasingly important because of its wide applicability in computer science, as well as in management and the other sciences. Two

key processes in discrete mathematics studied in this course are algorithmic problem solving and recursion. *Topics include* —logic and sets, graph theory, and difference equations. (Same as MATH 272) (4 credits) *Prerequisite:* MATH 162

CS 299 Teaching Practicum in Computer Science
In this course, students gain practical experience in the teaching methods of computer science by serving as full-time teaching assistants in a basic undergraduate course. Assistants conduct laboratory sessions with small groups of students, grade laboratory exams and exercises, and assist students individually. (2 credits) *Prerequisite:* consent of the department faculty

CS 310 Systems Programming: Connecting Hardware and Software — The Most Fundamental Level of Software in the Operating System
Students learn the systems programs that link the outer activity of high-level programming languages with the internal activity of the computer hardware. Knowledge of this deeper level of systems programs gives a greater range of possibilities to the programmer. Students learn system software such as compilers, linkers, loaders, and debuggers, and the structure and functions of an operating system including device management, process management, system calls, and memory management. (4 credits) *Prerequisite:* CS 222 and CS 272

CS 335 Software Development: Applying Knowledge of Software Systems for Greater Skill in Action
CS 336 Software Development Laboratory: Practical Experience in Applying the Knowledge of Computer Science to Create Software Systems
In these courses, students participate in a comprehensive system development project to apply and integrate the concepts of software design and implementation. Topics include methods and tools for large system development including analysis, design, testing, and documentation. Students work in teams to develop a substantial programming project. (4 credits each) *Prerequisite:* CS 222

CS 350 Programming Languages: The Abstractions at the Basis of Programming Languages — Gaining Mastery Over All Programming Languages
This course involves substantial programming exercises that give students practical experience with several different programming language paradigms. Topics include syntax and semantics of programming languages; data types and structures; control flow including blocks, subroutines, and recursion; implementation methods for semantic features; and comparison of several programming languages. (4 credits) *Prerequisite:* CS 222

CS 362 Computer Architecture: The Physiology of Computing Systems — The Physical Structures Reflecting the Underlying Computational Processes
This course investigates the levels and components of computer hardware as they contribute to the functioning of the computer. Topics include RTL systems and notations, bus structures, arithmetic logic units, execution and control design, micro-program control, input-output interface, hardware-software interactions, and microprocessors. Students study the integration of these system components in a sample uni-processor system and through case studies of actual machines. (4 credits) *Prerequisite:* CS 262

CS 390 Foundations of Modern Programming: Modern Programming Methods and Systems — Capture the Fundamental Principles of Knowledge for Greater Success in All Areas
This course presents the fundamental principles of object-oriented programming. Students will learn how to write reusable and better-maintained software, and integrate this knowledge with laboratory assignments and projects. *Topics include* —fundamental principles and models of object-oriented programming, UML class diagrams and design principles that promote reusability and maintainability of software. Also studied are stacks, queues, linked lists, and trees, using the Java programming language. *Prerequisite:* CS 220

CS 398 Computer Programming Internship: Knowledge and Experience for Maximum Growth
This course offers practical, professional experience in computer programming. Students apply classroom knowledge to an industrial or University project. During the internship, students submit detailed reports of their computer programming activities. (2 credits) *Prerequisite:* consent of the department faculty and the Academic Standards Committee

Dual Graduate/Undergraduate Courses

CS 401 Modern Programming Practices: Current Concepts and Best Practices in Software Development — Knowledge Is the Basis of Action
This course presents the fundamental principles of object-oriented programming. Students will learn how to write reusable and better-maintained software, and integrate this knowledge with laboratory assignments and projects. *Topics include* —fundamental principles and models of object-oriented programming, UML class diagrams and design principles that promote reusability and maintainability of software. (2 credits) *Prerequisite:* CS 220 or equivalent

CS 410 Modeling and Simulation: Software Models and Methods to Represent the Dynamics of Natural Law
This course studies models to enable a computer to simulate and predict the behavior of systems. Topics include discrete and continuous models, queuing models, process and event simulations, computer system models, and simulation languages and systems. (4 credits) *Prerequisites:* CS 401 or consent of the department faculty.

CS 420 Numerical Analysis: Methods to Map Nature's Infinite Precision into Finite Computing Systems
Scientific and engineering computer application requires advanced numerical techniques of manipulating and solving complex systems of equations with great efficiency and minimum error. Topics include numerical solution of linear equations, curve fitting, interpolation and polynomial equations, numerical integration and differentiation, solution of nonlinear equations, and error analysis. (4 credits) *Prerequisites:* CS 401 or consent of the department faculty.

CS 422 Database Management Systems: Capturing the Organizing Power of Information in Structured Models, Representations, and Query Languages
Database management systems organize and retrieve information, allowing the user to access the desired information easily and efficiently. Topics in this course include relational, hierarchical, and network data models; query languages; relational calculus, data normalization, and schemas; file organization techniques; data security and integrity; and study of a specific commercial database management system. (4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 425 Software Engineering: Knowledge Is the Basis of Action — Principles and Processes for Developing Large-Scale Software Systems
This course introduces the major principles used in the development of software. General principles and methods are identified, and their application is located in various phases and models of software engineering. The focus is on understanding the organizing power inherent in the underlying concepts, principles, and processes, rather than any particular developmental method or model. *Topics include* —the nature and qualities of software; types and qualities of specifications; objectives of design; verification approaches; production process models; and classification of supporting tools and environments. (4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 435 Algorithms: The Dynamics of Intelligence — The Relationship of Structure and Dynamics as the Basis for Efficient and Practical Software Development
This course presents methods for analyzing the efficiency of algorithms as well as a variety of known efficient algorithms. Topics include graph algorithms, combinatorial algorithms, searching and sorting, numerical and arithmetic algorithms, recurrence

relations, computing time and space complexity of algorithms, and NP-complete problems. (4 credits) *Prerequisites:* CS 401 or consent of the department faculty.

CS 440 Compiler Construction: Connecting Name and Form — The Source of All Programming Languages in Grammar and Semantics
Students learn the successive stages and detailed mechanics by which high-level programming languages are translated into machine language by a compiler. Topics include language and grammar specification, compiler structure, compiler generation tools, lexical analysis, parsing, syntax analysis, semantic analysis, intermediate language, code generation and optimization, storage management and linkages, user interface, and a large programming project implementing part of a compiler. (4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 450 Computer Communication Networks: Connecting the Parts and Whole — Frictionless Flow of Information
Computers are connected with high-speed communication lines in local area or wide area networks, for the purpose of sharing databases and distributing workloads to increase efficiency and improve service. Topics include sampling and information theory, error detecting and correction codes, network architecture, communication protocols and models, protocol analysis, hardware components, logical and physical topology, message routing and switching, flow control, local area networks, and data security. (4 credits) *Prerequisites:* CS 401 or consent of the department faculty.

CS 455 Software Technologies: Advanced Principles of Natural Law in Software Systems
This course will cover the most current emerging methods, principles, and practices in software technologies and systems. The topics will vary, based on current technologies and instructor choices. (2 or 4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 460 Scientific Computing: Software Models and Methods to Represent the Mathematical Precision of Natural Law
This course presents methods and principles for the application of computing systems to scientific and engineering problems. Areas studied in this course are numerical methods, scientific computation, and applications. Specific topics covered are computational efficiency, accuracy and precision, root finding, Taylor series and function evaluation, interpolation and approximation, finite difference calculus, curve fitting, and numerical integration. (4 credits) *Prerequisites:* CS 401 or consent of the department faculty.

CS 465 Operating Systems: The Most Fundamental Level of Software — Organizing Hardware Resources into Coherent Virtual Systems
An operating system controls the central resources of the computer system and allocates them to individual users. Course topics include sequential and concurrent processes, mutual exclusion, resource sharing, process cooperation, deadlock, resource allocation, processor scheduling, memory management, segmentation and paging algorithms, timesharing systems, scheduling algorithms, and resource protection. (4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 466 Computer Security
This course goes deeply into the three aspects of computer security: confidentiality, integrity, and availability. Several models for confidential and integrity security policies are studied. The role of cryptography in assuring confidentiality and integrity is examined. Other topics include authentication, auditing, penetration testing, common vulnerabilities and intrusion detection. The course concludes with the case study of a realistic secure system. Students will be asked to read papers from the security literature and apply them to material given in the lectures. (4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 467 Secure Coding Practices
The course examines the 19 issues that account for 95% of the security vulnerabilities that occur in the field. The issues are: buffer overflows, format string problems, integer range errors, SQL injection, command injection, failure to handle errors, cross-site scripting, failing to protect network traffic, use of "magic" URLs and hidden fields, improper use of SSL, use of weak password-based systems, failing to store and protect data securely, information leakage, improper file access, trusting network address information, race conditions (improper thread programming), unauthenticated key exchange, failing to use cryptographically strong random numbers, and poor usability. The final project of the course will analyze and remove vulnerabilities from a Web application. The course will emphasize that a computer programmer needs both broad comprehension and the ability to focus to produce secure software. (2 or 4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 470 Knowledge-Based Systems: Knowledge is the Basis of Thinking, Action, and Achievement — Creating Intelligent Software Systems
The field of artificial intelligence attempts to create computer programs that reflect the values of human intelligence. Course topics include state-space representations, tree and graph searches, predicate calculus and deduction, heuristics, learning and problem solving, natural language processing, expert systems, and programming languages for

artificial intelligence. (4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 471 Parallel Programming
The standard processor for all new computers is now a *multi-core* processor, which has the potential to execute programs much more quickly. However, to utilize this potential, a programmer must have some knowledge of *parallel programming* techniques. During this course, students will spend most of their time writing and debugging parallel programs. The expected outcome will be to develop a new level of practical programming skill. This skill will not only be useful for programming of multi-core processors, but also operating systems programming and distributed database programming. The software tools used during this course include Microsoft Visual C/C++, the OpenMP threading standard, and the Message-Passing Interface (MPI) standard. In addition to multi-core processors, this course also covers techniques for programming a computer *cluster* (many individual workstations networked together and working collectively on a single computation) (4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 472 Web Programming
Learn to develop Web 2.0 applications using many newer technologies such as XHTML, CSS, JavaScript, PHP, MySQL, and Ajax. The course features a project-based approach to learning with hands-on exercises requiring programming skills. Students apply design strategies to make scalable websites and access data from other websites and servers. Emphasis will be on programming and solving design issues. (4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 475 Computer Graphics: How to Represent and Graphically Express the Dynamic Intelligence Captured in Software Systems
One of the fastest growing areas of computer technology, computer graphics is used extensively to present the vast amount of information resulting from a computing process. This course studies data representation, display devices and graphics hardware, display lists, device independence, two-dimensional and three-dimensional graphics, display of curves and surfaces, hidden line and hidden surface removal, shading and rotation techniques, graphics languages, and introduction to image processing. (2–4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 476 Image Processing: Visual Expression of Total Knowledge
The course presents the concepts and operations of digital image processing, which treats all images as a collection of binary pixels. The course studies how these billions of parts are treated as a single integrated image, and the mathematical and algorithmic aspects of and tools for processing these images. Topics include image representation and

transformations, filtering, and Fourier domain filtering and transformations, edge detection, segmentation, and other processing operations. The course includes a substantial lab component. (2-4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 485 Theory of Computation: The Abstract Basis of All Possibilities in Computation

Formal abstract models of computation study the fundamental limitations and capabilities of computers. This course presents a hierarchy of increasingly sophisticated abstract machines in relation to their increasing ability to recognize more general classes of formal languages. Topics include formal grammar, finite-state machines, equivalence of finite-state machines, right-linear and left-linear grammar, context-free languages, Turing machines, unsolvable problems, and recursive functions. (4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 487 Distributed Computing and XML

This course will investigate the uses of XML in distributed computation. First an understanding of the W3C specifications for XML, XML Schema, XPath, XML namespaces, XSLT and XQuery will be acquired. Then three important applications of XML in distributed computing will be investigated: syndication, Web services and Ajax. Finally, advanced issues such as encrypted XML and binary XML will be considered. There will be daily labs using .NET 2.0. (4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 490 Topics in Computing

This course surveys and studies current technologies and application areas in computing. Typically it will include a substantial research and laboratory component to gain experience with advanced areas of computing and computer science. (2–4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 499 Directed Study: Faculty Directed Study of Specialized Topics

(variable credits) *Prerequisite:* consent of the department faculty

Graduate Only Courses

CS 501 Advanced Computer Architecture: Structured Intelligence — Computational Structures That Reflect the Dynamics of Computation

This course presents the methods, principles, and metrics of computer systems architecture. The interactions of hardware components, system architecture, and software algorithms are the basis for evaluating the performance and characteristics of a range of advanced computing systems. Topics include pipelined and multiprocessing architecture,

parallel processing, distributed processing, case studies, and comparisons of existing systems. (4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 505 Advanced Programming Languages: The Integrated Source of All Programming Languages as a Basis for Understanding and Applying Principles of Programming

This course considers advanced topics in programming language design with emphasis on formal methods and abstraction mechanisms. Topics include data and control abstraction, formal specification of syntax and semantics, proofs of program correctness, non-deterministic programming, advanced control structures, and study of specific languages. (4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 510 Advanced Operating Systems: Extending the Qualities of Integration, Unity, and Efficiency to Both Local and Distributed Operating Systems

The course covers advanced topics in operating systems including analytical models and theory. Topics are selected from the following: models for parallel computation, Petrinets, dataflow diagrams, distributed operating systems, queuing theory, system simulation, performance evaluation, dynamic protection concepts and mechanisms, and fault tolerant systems. (4 credits) *Prerequisites:* CS 401 or consent of the department faculty.

CS 515 Advanced Theory of Computation: Locating the Basis of All Computation in the Abstract Field of Pure Intelligence

Formal models for computation and computability are surveyed, including an introduction to complexity theory. Topics include partial recursive, recursive, and primitive recursive functions; recursive and recursively enumerable sets; Gödel numberings; degrees of unsolvability; the recursion theorem; program schemes; and elementary complexity theory. (4 credits) *Prerequisite:* CS 485

CS 525 Advanced Software Development: The Structures and Patterns of Natural Law in Software That Embody Knowledge of Good Design

This course considers the current methods and practices for good design of software systems. *Topics include* —software design patterns, frameworks, architectures, and designing systems to apply these multi-level abstractions. (2–4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 526 Software Architecture: The Unifying Principles in Large Software Systems

This course studies the overall structure, relationships, and dynamics of the software components that comprise various levels of a system, so that they form an integrated result that meets the design objectives. Topics include software components, component models, system specifications and modeling, and architectural patterns. Students will

learn the principles, methods, and examples of good software architectures, and apply them in a project or presentation. (4 credits) *Prerequisite:* CS 401 or consent of the department faculty.

CS 530 Topics in Database Systems: Higher-Level Structures in Information Reflecting Greater Knowledge of Natural Law
This course considers advanced issues in database management systems design and implementation. Topics include database transactions, constraint checking, security, integrity, recovery techniques, schemas and views of data, semantic data models, entity-relationship models, extended relational models, distributed databases, and database machines. (4 credits) *Prerequisite:* CS 422

CS 535 Advanced Software Engineering: Advanced Principles of Coherency and Integration in Software Development Processes
This course considers advanced issues in software engineering. Course topics vary but are selected from areas that represent advanced practices in modern industry, e.g., software testing, object-oriented methodologies, and software requirements. (2–4 credits) *Prerequisite:* CS 425

CS 545 Distributed Computing: Integration of Parts and Wholeness in Large-Scale Distributed Software Systems
This course presents the issues, methods, and techniques for creating multi-computing distributed systems across networked or more tightly coupled interconnect systems. Topics include communication, protocol, and synchronization; performance; and the architecture of server, client/server, multi-tier, and mobile agent distributed object systems. Software issues of portability, extendibility, and interoperability are also studied. (4 credits) *Prerequisite:* CS 401

CS 547 Distributed Computing Architecture: Integrating Parts and Wholeness in Large-Scale Distributed Software Systems
This course discusses advanced issues and principles pertinent to modern enterprise systems, such as object-oriented middleware technologies, Message-Oriented-Middleware (MOM), distributed architecture, design patterns, and frameworks. (4 credits) *Prerequisite:* CS 545

CS 550 Topics in Design and Analysis of Algorithms: Advanced Study of the Relationship of Form & Function in Software — Capturing Nature's Perfect Efficiency
This course includes a survey of efficient algorithms in various areas, including analysis techniques and theoretical issues. Topics vary and are selected from the following: arithmetic and combinatorial algorithms, searching and sorting, numerical algorithms,

probabilistic and parallel algorithms, proofs of correctness and efficiency, lower bounds, and average-case behavior. (4 credits) *Prerequisite:* CS 435

CS 560 Topics in Numerical Methods: Methods to Represent Nature's Infinite Precision in Finite Computing Systems
Specialized computational techniques for solving practical numerical problems in various areas of science and engineering are considered. Topics vary, including areas such as linear programming, optimization techniques, time series analysis, forecasting, Fourier transforms, finite element methods, solution of differential equations, and simulation. (4 credits) *Prerequisites:* CS 420, MATH 306, and MATH 308

CS 570 Teaching of Computer Science
Students gain practical experience teaching computer science by serving as full-time teaching assistants in one of the basic undergraduate courses. Assistants conduct laboratory sessions with small groups of students, grade laboratory programs and exercises, and assist students individually. (2 credits — may be repeated) *Prerequisite:* consent of department faculty

CS 575 Practicum in Software Development (away from Fairfield)
CS 576 Practicum in Computer Operations (in Fairfield)
In this practicum course, students perform computer-related tasks in a technical professional position. The tasks performed may be in the design and development of new systems or the application of existing systems for specific purposes. The job activities must relate to coursework studied during the Master's degree. Practicum job descriptions are formulated by the employer and the student, and require approval in advance by one of the graduate faculty of the department, in consultation with the practicum supervisor where the student is placed. (These courses are primarily for students in the internship or cooperative programs.) (0.5–1 credit per block — may be repeated) Students need written authorization to take these courses.

CS 577 Practicum in Administrative Applications of Computers (for on-campus interns)
CS 578 Practicum in Scientific Applications of Computers
In these practicum courses, students perform computer-related tasks in one of the administrative or academic departments of the University. The tasks performed may be in the design and development of new systems or the application of existing systems for specific purposes, and require approval in advance by one of the graduate faculty of the department, in consultation with the practicum supervisor in the department where the student is placed. (These courses are primarily for students in the internship or cooperative programs.) (0.5 credits each per block — may be repeated) Students need written authorization to take these courses.

CS 579 Practicum in Teaching of Computer Science
Students gain practical experience teaching computer science by serving as full-time teaching assistants in one of the basic undergraduate courses. Assistants conduct laboratory sessions with small groups of students, grade laboratory programs and exercises, and assist students individually. (This course is primarily for students in the internship or cooperative programs.) (0.5 credits per block — may be repeated) Students need written authorization to take this course.

CS 580 Seminar in Current Research Topics
Advanced knowledge and current research issues are presented in a specialized area of computer science. The course includes readings of current journal articles in the field and a substantial independent project by students. (4 credits — may be repeated) *Prerequisite:* consent of instructor

CS 581 Seminar in Professional Computing: Advanced Topics in Nature's Computing Processes
This course provides topical knowledge relevant to professional applications of computing. Topics will vary each time it is offered. Topics may include: object-oriented programming, object-oriented analysis and design, client/server models and distributed systems, real-time programming, real-time systems, software quality assurance and measurement, applied AI and expert systems, and database management tools. (1 credit — may be repeated) *Prerequisite:* consent of department faculty

CS 585 Integration Project and Comprehensive Examination
This course reviews and integrates knowledge presented in the four graduate core courses: CS 465, CS 485, CS 501, and CS 505. Students write a substantial paper using the dynamics of the Unified Field of Natural Law as an intellectual framework to integrate the concepts presented in the four core courses. The course ends with a comprehensive examination covering the core courses. (4 credits) *Prerequisites:* CS 465, CS 485, CS 501, and CS 505

CS 586 Cooperative Research Project
Students conduct an extended project related to their cooperative practicum project. Students work with their supervisor and the faculty to add a research component to a main technical aspect of their work, and will present a final written report and oral presentation. (4 credits) *Prerequisite:* consent of department faculty

CS 588 Directed Research
Students conduct an original research project with the guidance of the computer science faculty. (variable credits) *Prerequisite:* consent of the department and the Academic Standards Committee

CS 591 Directed Study in Computer Science
CS 592 Directed Study in Computer Applications
CS 593 Directed Study in Mathematics
CS 595 Directed Study in Scientific Applications of Computers
CS 596 Directed Study in the Science of Creative Intelligence
In these courses the student spends six hours per week in the evenings covering material from one of the regular courses, or special material selected by the faculty according to the needs and program of study of the student. In some cases, a faculty member outside the Department of Computer Science supervises the directed study. However, the selection of material to be covered and the final evaluation is subject to the approval of the graduate faculty. (These courses are for students in the internship program only.) (1–2 credits each — may be repeated) *Prerequisite:* consent of department faculty

CS 598 Computer Science Internship
This course offers practical, professional experience in computer programming. Students apply classroom knowledge to an industrial or University project. During the internship, students submit detailed reports on their computer programming activities. (2 credits) *Prerequisites:* consent of the department and the Academic Standards Committee and written authorization

CS 599 Directed Study
(4 credits) *Prerequisite:* consent of the department faculty