Department of Computer Science

Stuart Building
10 W. 31st St.
Chicago, IL 60616
312.567.5150
312.567.5067 fax
www.cs.iit.edu
info@cs.iit.edu

Chair:
Edward M. Reingold

The study of computer science is the inquiry into the nature of computation and its use in solving problems in an information-based society. Computer science is an evolving discipline, but it has a well-defined core of knowledge and a set of characteristic methodologies. The methods and skills required of the computer scientist include formalization and abstraction, algorithm design, programming, organization of unstructured knowledge, modeling, language development, and software system architecture and design. The graduate program in computer science at IIT stresses high achievement in both fundamental knowledge and practical problem solving. It offers the student a solid background in the core areas and exposure to cutting-edge computer technologies.

Degrees Offered

Master of Computer Science (MAS)
Master of Computer Science with specialization in
- Computer Networking and Telecommunications
- Information Systems
- Software Engineering

Master of Science in Computer Science
Master of Science for Teachers
Doctor of Philosophy in Computer Science

Joint- and Dual-Degree Programs

With the Department of Electrical and Computer Engineering:
Master of Telecommunications and Software Engineering

With the Department of Chemical and Environmental Engineering:
Master of Science in Computer Science/ Master of Chemical Engineering

Certificate Programs

Computer Networking and Telecommunications
Information Systems
Software Engineering

Research Facilities

The department has research computing facilities that include Sun SPARC and Silicon Graphics UNIX work-stations, and Windows-based PCs. The department also has facilities for research in parallel computing. The equipment includes a large scale "ComputeFarm" consisting of 72 Sun Blade UltraSparc-ii workstations and a 4 CPU Sun Enterprise 450 server connected via high-performance network switches. An advanced technology Access Grid node has been installed by the department, which allows researchers, teachers and students at different locations worldwide to interact via a network multimedia environment.

Research Areas

Algorithms, data structures, artificial intelligence, computer architecture, computer graphics, computer networking and telecommunications, computer vision, database systems, distributed and parallel computing, image processing, information retrieval, natural language processing, and software engineering.

Department of Computer Science

Faculty

Gady Agam, Assistant Professor. B.Sc., M.Sc., Ph.D., Ben-Gurion University. Computer vision, computer graphics, image processing, pattern recognition, image-based rendering, non-linear signal processing, mathematical morphology.

Shlomo Argamon, Associate Professor. B.S. Carnegie-Mellon University, M.Phil, Ph.D. Yale University. Machine Learning, computational linguistics, stylistics, information retrieval.

Charles Bauer, Professor Emeritus. B.S., M.Ed., Loyola University. Computer Education, computers used in education, distance learning, computer-based educational technology, computer learning, teacher training.

Ilene Burnstein, Research Associate Professor. B.S., Brooklyn College; M.S., University of Maryland; Ph.D., Illinois Institute of Technology. Software engineering, knowledge-based testing and debugging tools, test process assessment and improvement models, capability maturity models.

Gruia Calinescu, Assistant Professor. Diploma, University of Bucharest (Romania); Ph.D., Georgia Institute of Technology. Algorithms, approximation algorithms, optical and wireless, ad-hoc networks.

C. Robert Carlson, Professor, Director of Center for Professional Development. B.A., Augustana College; M.S., Ph.D., University of Iowa. Information architecture, object-oriented modeling and design, software maturity models.

Edward Chlebus, Industry Associate Professor., Ph.D., Cracow University, Network modeling, performance evaluation and tele-traffic analysis.

Tzilla Elrad, Research Associate Professor. B.S., Hebrew University (Israel); M.S., Syracuse University; Ph.D., Technion Israel Institute of Technology (Israel). Concurrent programming, formal verification, embedded real-time systems and ADA standards.

Martha Evens, Research Professor. A.B., Bryn Mawr College; A.M., Radcliffe College; Ph.D., Northwestern University. Natural language processing, expert systems and intelligent tutoring/information systems.

Ophir Frieder, IITRI Chair Professor. Ph.D., University of Michigan. Parallel and distributed information retrieval systems, communication systems, high performance database systems, biological and medical data processing architectures.

Nazli Goharian, Clinical Assistant Professor, Ph.D., Florida Tech, Information retrieval, medical informatics and data warehousing.

Peter Greene, Research Professor. A.B., Amherst College; Ph.D., University of Chicago. Neural networks, feeling-based reasoning, artificial intelligence and robotics.

David Grossman, Associate Professor. B.S., Clemson University; M.S., American University; Ph.D., George Mason University. Information retrieval, data mining, integration of structured data and text.

Cynthia Hood, Associate Professor. B.S., Rensselaer Polytechnic Institute; M.E., Stevens Institute of Technology; Ph.D., Rensselaer Polytechnic Institute. Network management, statistical signal processing, learning processing.

Sanjiv Kapoor, Professor. B.Tech., Indian Institute of Technology, Delhi (India); Ph.D., University of Illinois, Urbana-Champaign. Computational geometry, graph algorithms, combinatorial optimization, graphics, data structures.

Bogdan Korel, Associate Professor. M.S., Technical University of Kiev (Ukraine); Ph.D., Oakland University. Software engineering, automated software testing and analysis.

Zhiling Lan, Assistant Professor, B.S. Beijing Normal University, M.S. Chinese Academy of Sciences, Ph.D., Northwestern University. Parallel and distributed computing, performance analysis and modeling.

Xiang-Yang Li, Assistant Professor. B.S., B.M., Tsinghua University, Beijing (China); M.S., Ph.D., University of Illinois, Urbana-Champaign. Computational geometry, mesh generation, wireless ad hoc networks, cryptography, computer graphics.

Edward M. Reingold, Professor. B.S., Illinois Institute of Technology; M.S., Ph.D., Cornell University. Analysis of algorithms, data structures, combinatorial algorithms, mathematics, history, and computer implementation of calendars.

Shangping Ren, Assistant Professor. Ph.D., University of Illinois, Urbana-Champaign. Domain specific computing (including distributed computing, real-time computing, and mobile computing), and domain specific programming abstractions (such as language abstractions for real-time systems, for context-aware adaptive mobile systems, and for coordination among distributed asynchronous entities).

Xian-He Sun, Professor. Ph.D., Michigan State University. Distributed and parallel system, network software environment, scientific computing performance evaluation.

Peng-Jun Wan, Associate Professor. B.S., Tsinghua University (China); M.S., Institute of Applied Mathematics, Chinese Academy of Sciences (China); Ph.D., University of Minnesota. Interconnection design, routing and resource management in optical networks, low-earth orbit satellite networks, wireless local area networks.

Wai Gen Yee, Assistant Professor, B.S. University of Chicago, M.S., Ph.D. Georgia Institute of Technology. Database systems, mobile and distributed computing.

Admission Requirements

Cumulative undergraduate GPA minimum: 3.0/4.0
Cumulative Master of Science GPA minimum (for Ph.D. applicants): 3.5/4.0
GRE minimum combined (quantitative/verbal/analytical) score for tests taken prior to Oct.1, 2002:
M.S.: 1400
MAS : 1200
Ph.D.: 1600, with a minimum in the 70th percentile of the quantitative section
GRE minimum score for tests taken on or after Oct.1, 2002:
M.S.: 1000 (quantitative + verbal) 3.0 analytical writing
MAS: 900 (quantitative + verbal) 2.5 analytical writing
Ph.D.: 1100 (quantitative + verbal) 3.5 analytical writing, with a minimum in the 70th percentile of the quantitative section
TOEFL score (international students from non-English speaking countries): 550/213*

Meeting the minimum GPA and test score requirements does not guarantee admission. Test scores and GPA are just two of several important factors considered. Applicants to master's degree programs in computer science should hold a bachelor's degree in computer science with a minimum overall GPA of 3.0/4.0 or its equivalent. For international students from non-English speaking countries, a minimum TOEFL score of 550/213 is required. All applicants must submit scores from the GRE general test. (The GRE requirement is waived for applicants to the Master of Computer Science program who hold bachelor's degrees from accredited U.S. institutions with a minimum cumulative GPA of 3.0/4.0.) Applicants with bachelor's degrees in other disciplines can be admitted to Master of Science or Master of Computer Science programs. However, students whose training does not include the equivalent of CS 201 (Accelerated Introduction to Computer Science), CS 330 (Discrete Structures), CS 331 (Data Structures and Algorithms), CS 350 (Computer Organization and Assembly Language Programming) and CS 351 (Systems Programming) will be required to complete all of the courses in which a deficiency exists. Some students may be able to complete their deficiencies with the following six-credit hour sequence with a grade of "B" or better:

CS 401 Introduction to Advanced Studies I
CS 402 Introduction to Advanced Studies II

In addition, students who have not had at least one course in calculus will be required to take a calculus course.

Applicants to the Ph.D. program should hold an M.S. degree in computer science with a minimum GPA of 3.0/4.0 for their bachelor's degree and 3.5/4.0 for their M.S. degree, or a minimum GPA of 3.5/4.0 for their bachelor's degree if they apply without a M.S. degree. For non-English speaking applicants without a U.S. degree, a minimum TOEFL score of 550/213 is required.

* Paper-based test score/computer-based test score.

Master of Computer Science

30 credits.

This Professional Master's degree program consists of 30 credit hours of coursework in computer science. This program is designed for those without a prior degree in computer science, or those who are primarily interested in a (non-thesis) program preparing them for careers as working computer science professionals in business and industry. A full-time student enrolled in the program should be able to complete the requirements in 1 to 1.5 years. Specializations in business, software engineering, networking and telecommunications and information systems are available. Admission requirements include:

Four-year bachelor's degree from an accredited university with a minimum cumulative GPA of at least 3.0/4.0. Combined verbal and quantitative GRE examination score of at least 900 and an analytic writing score of at least 2.5, for the post-October 2002 test. The GRE requirement is waived for students with a bachelor's degree from an accredited college or university in the United States with a cumulative GPA of at least 3.0/4.0.

Prerequisites include knowledge of a high level programming language at the level of CS 201 (Java or C/C++ programming is required). Students with insufficient background in computer science will be required to take CS 401 and CS 402 (Introduction to Advanced Studies I and II) and to earn at least a "B" in these courses. These prerequisite courses do not count toward the 30 credit-hour requirement.

A maximum of 12 credit hours of 400-level courses and a maximum of 6 credit hours of accelerated courses are allowed as part of the 30 credit hours requirement. Twenty hours of coursework must be in CS or CSP (CS Professional) courses at the 500 level.

Students are required to take one course in each of the three core areas (Programming, Systems and Theory).

Master of Computer Science continued

Programming core courses

CS 522	Data Mining
CS 525	Advanced Database Organization
CS 529	Information Retrieval
CS 540	Syntactic Analysis of Programming Languages
CS 546	Parallel Processing
CS 551	Operating System Design and Implementation

Systems core courses

CS 542	Computer Networks I: Fundamentals
CS 544	Computer Networks II: Network Services
CS 547	Wireless Networking
CS 550	Advanced Operating Systems
CS 555	Analytic Models and Simulation of Computer Systems
CS 570	Advanced Computer Architecture
CS 586	Software Systems Architecture

Theory core courses

CS 530	Theory of Computation
CS 532	Formal Languages
CS 533	Computational Geometry
CS 535	Design and Analysis of Algorithms
CS 536	Science of Programming
CS 538	Combinatorial Organization

Master of Computer Science with specialization in Business

33.6 credit hours

This program is designed to help computer science professionals extend and deepen their technical and practical knowledge of the field while introducing themselves to core topics in modern business practices. To complete the program students must satisfy Master of Computer Science requirements and also take four specialization courses from the Stuart School of Business:

MBA 510:	Financial and Managerial Accounting
MBA 520:	Organizational Behavior
MBA 530:	Managerial Economics

One other advisor-approved 500-level course offered by the Stuart School of Business for their MBA program (MBA, ACCT, EBUS, ECON, etc).

Notes: Each business course counts as 3.6 quarter credit hours = 2.4 semester credit hours. A student can take no more than two MBA courses per quarter. With advisor approval, students who have already taken MBA 510, 520, or 530 or their equivalents can substitute other courses. Applicants to the program are not required to take the GMAT.

Master of Computer Science with specialization in Computer Networking and telecommunications

30 credit hours

This program is designed to provide an in-depth knowledge of the theories and practices in computer networking and telecommunications. Students must satisfy Master of Computer Science requirements and are also required to select four specialization courses. Note that CS 542, CS 544, and CS 547 also count as Systems core courses.

Specialization courses

CS 542	Computer Networks I: Fundamentals
CS 544	Computer Networks II: Network Services
CS 547	Wireless Networking
CS 548	Broadband Networks
CS 555	Analytic Models and Simulation of Computers Systems
CS 549	Cryptography and Network Security

Master of Computer Science with specialization in Information Systems

30 credit hours

This program is designed to provide in-depth knowledge of the principles of design and development of information systems. Students must satisfy Master of Computer Science requirements and are also required to select four specialization courses. Note that CS 522, CS 525, and CS 529 also count as Programming core courses

Specialization courses

CS 521	Object-Oriented Analysis and Design
CS 522	Data Mining
CS 525	Advanced Database Organization
CS 529	Information Retrieval
CS 545	Distributed Computing Landscape

Master of Computer Science with specialization in Software Engineering

30 credit hours

This program is designed to provide an in-depth knowledge of theory and practices in software engineering, including hands-on experience in software design, development and maintenance. Students must satisfy Master of Computer Science requirements and are also required to select four specialization courses here. Note that CS 536 and CS 586 also count as Theory and Systems core courses respectively.

Core courses

CS 536	Science of Programming
CS 537	Software Metrics
CS 586	Software System Architectures
CS 587	Software Project Management
CS 588	Advanced Software Engineering Development
CS 589	Software Testing and Analysis

Master of Science in Computer Science

32 credit hours

The purpose of this program is to prepare students for the Ph.D. program and/or a research/development career in the industry in the field of computer science. Students have the option to pursue thesis research or project under the guidance of a faculty adviser.

Program Requirements

All programs require a core curriculum of 12 credit hours and 20 credit hours of elective courses, which may include a thesis or project. The student, with a faculty adviser, develops a program of study that specifies the supportive and elective program and describes the thesis or project, if included in the program. The program of study must consist of at least 32 credit hours, at least 20 of which must be 500-level computer science courses. Up to six credits of accelerated courses may be applied to the degree. (Students should see the definition of "accelerated courses" on page 41.)

A student may choose from three options to complete the degree:

Option 1: Master's thesis: Coursework and up to five hours of CS 591 for a total of 32 hours. The result is a master's thesis.

Option 2: Master's project: coursework and up to five hours of CS 597 for a total of 32 hours. The result is a project that results in one of the following:

1. A high-quality paper submitted for publication as an article or as a technical report.

2. A high-quality piece of software. The software should be of distribution quality, but can be proprietary.

Option 3: 32 credit hours of coursework. A student must complete 32 hours of regular coursework including electives and core courses with a GPA of 3.0/4.0 or better.

Students are required to take courses in three core areas: Programming, Systems and Theory. The student is required to take one course in the programming area, one course in the systems area, and two courses in the theory area. The list below contains the core course offerings in the M.S. program:

Programming core courses

CS 522	Data Mining
CS 525	Advanced Database Organization
CS 529	Information Retrieval
CS 540	Syntactic Analysis of Programming Languages
CS 546	Parallel Processing
CS 551	Operating System Design and Implementation

Systems core courses

CS 542	Computer Networks I: Fundamentals
CS 544	Computer Networks II: Network Services
CS 547	Wireless Networks
CS 550	Advanced Operating Systems
CS 555	Analytic Models and Simulation of Computer Systems
CS 570	Advanced Computer Architecture
CS 586	Software Systems Architectures

Theory core courses

CS 530	Theory of Computation
CS 532	Formal Languages
CS 533	Computational Geometry
CS 535	Design and Analysis of Algorithms
CS 536	Science of Programming
CS 538	Combinatorial Optimization

Master of Science for Teachers (M.S.T.)

32 credit hours
Comprehensive exam (project)

The M.S.T. is designed for experienced teachers or training officers to strengthen their academic background in a rapidly changing discipline. The program, though flexible enough to meet a variety of needs, also requires substantive coursework in the core of computer science. The student, with a faculty adviser, develops a program of study, describes the project, and specifies an elective program, which must be approved by the faculty of the department.

The program of study consists of 32 credit hours, at least 20 hours of which must be 500-level courses. It also includes an M.S.T. project that deals with some aspect of computer science or with computer science applied to some other academic discipline. To be awarded the M.S.T. degree, the student must satisfactorily complete the program of study and pass a project defense examination, which consists of an oral defense of the project. Of the 32 credit hours, 12 credit hours must be from the courses listed below:

CS 485	Computers in Society
CS 560	Computer Science in the Classroom
CS 561	The Computer and Curriculum Content
CS 565	Computer-Assisted Instruction
CS 566	Practicum in the Application of Computers to Education

Master of Telecommunications and Software Engineering (M.T.S.E.)

30 credit hours

The Master of Telecommunications and Software Engineering (M.T.S.E.) is a course-only degree program that prepares students for professional practice in telecommunications and information technologies. The program, jointly offered by the electrical and computer engineering (ECE) and computer science (CS) departments, can be completed in one year of full-time study.

Admission Requirements

A person holding a B.S.E.E., B.S.C.P.E. or B.S.C.S. degree has the necessary broad background to undertake the M.T.S.E. program. A student without adequate background in specific areas is required to demonstrate proficiency in prerequisite courses; an abbreviated list is given below.

Specific proficiency courses will be detailed for each student at the time of admission to the M.T.S.E. program. Proficiency in a course may be demonstrated by completing the course with a grade of "A" or "B," or by achieving a grade of "A" or "B" in a proficiency examination administered by the ECE or the CS department. Students should contact the departmental adviser for more details on prerequisites and proficiency requirements.

Students interested in the M.T.S.E. degree whose B.S. degree is not in electrical engineering, computer engineering, or computer science should contact the departmental advisor before applying.

Computer Science Prerequisites

CS 201	Accelerated Introduction to Computer Science
CS 401	Introduction to Advanced Studies

Electrical and Computer Engineering Prerequisites

ECE 211	Circuit Analysis I
ECE 213	Circuit Analysis II
ECE 308	Signals and Systems
MATH 252	Introduction of Differential Equations

Program Requirements

The M.T.S.E. is a professional master's degree requiring a minimum of 30 credit hours of adviser-approved coursework. The M.T.S.E program of studies must include a minimum of 12 credit hours of ECE coursework and a minimum of 12 credit hours of CS coursework. Four required courses and one elective course from each of the three categories given below must appear on the M.T.S.E. program of studies.

Required courses

CS 586	Software Systems Architecture
CS 587	Software Project Management
ECE 513	Communication Engineering Fundamentals
ECE 541	Performance Evaluation of Computers and Communications Networks

Elective categories

I. Software Engineering

CS 521	Object-Oriented Information Systems
CS 537	Software Metrics
CS 589	Software Testing and Analysis

II. Telecommunication Systems

CS 544	Computer Networks II: Network Services
CS 548	Broadband Networks
CS 555	Analytic Models and Simulation of Computer Systems
ECE 545	Computer and Communication Networks

III. Telecommunications

ECE 504	Communications Systems Design
ECE 515	Modern Digital Communications
ECE 519	Coding for Reliable Communications

The remaining nine credits of coursework may be taken from courses listed above, or other courses approved by the M.T.S.E. adviser. Students with no background in communications or software engineering should consider including in their programs of study:

CS 487	Software Engineering
CS 450	Operating Systems
ECE 403	Communication Systems I
ECE 407	Computer Communications Systems

Other courses that students in this program typically choose from include:

ECE 437	Digital Signal Processing I
ECE 511	Analysis of Random Signals
ECE 514	Digital Communication Principles
CS 542	Computer Networks I: Fundamentals
CS 588	Advanced Software Engineering Development

With advisor approval, the M.T.S.E. program can include up to two credit hours of Master's Seminar (ECE 595, ECE 596). It can also include up to four credits of accelerated courses.

Master of Science in Computer Science/Master of Chemical Engineering

44 credit hours

This combined program in computer science and chemical engineering addresses the growing need for process engineers with expertise in computational modeling and simulation of chemical processes. Similarly, the program provides strong engineering background that is required today in many areas of computer science. The program is jointly offered by the Department of Computer Science and the Department of Chemical and Environmental Engineering. Students in this program earn both Master of Science in Computer Science and Master of Chemical Engineering degrees.

Students must fulfill the core course requirements of both departments. Students are required to take 18 credit hours in graduate chemical engineering courses (courses numbered 500 or higher) and 26 credit hours in computer science courses (of which 20 credit hours must be 500-level courses). The 18 credit hours in chemical engineering courses consist of 12 credits in core courses listed in the description of the Master of Chemical Engineering requirements and six credit hours from the following courses:

CHE 507 Computer-Aided Design
CHE 508 Process Design Optimization
CHE 528 Analysis and Simulation of Chemical Processing
CHE 532 Process Modeling
CHE 533 Statistical Analysis of Process Data
CHE 536 Computational Techniques in Engineering
CHE 560 Statistical Quality and Process Control

Doctor of Philosophy

85 credit hours if without MS degree; 54 credit hours if with MS degree.
Qualifying exam
Comprehensive exam
Dissertation and oral defense

The Ph.D. is awarded in recognition of a significant original contribution to one of the fields of computer science and a high level of mastery in several fields of computer science and a significant original contribution to one of those fields. Students work with faculty members to develop programs to match individual interests. The goal is to develop computer scientists who can take complex, undefined problems and restructure and resolve them through imaginative application of their knowledge. Graduates typically go on to teaching and/or research positions in industry and universities. The degree normally requires three to four years beyond the master's degree for full-time students. Part-time students can also enter the program but will need more time to complete the degree. Generally, students can enter the program with either a BS degree or an MS degree in related fields. The requirements of the Ph.D. program are described separately as follows.

Requirements for students entering with B.S. Degree

85 credit hours
Qualifying exam
Comprehensive exam
Dissertation and oral defense

The Ph.D. (post B.S.) program (called the direct Ph.D. program) encourages bright and highly motivated students to participate in research program immediately after the B.S. degree.

Admission Requirements

The applicants should have at least a four-year B.S degree in computer science. Admission to the program is competitive and depends on a student's GRE score and it is expected that applicants will have a high grade-point average. For non-English speaking applicants without a U.S. degree, a minimum TOEFL score of 550 is required. If the TOEFL score is less than 600, the applicant is required to take the English Proficiency Exam administered by the IIT Humanities Department. Applicants must submit three letters of recommendation.

Program Requirements

The program requires students to complete at least 85 and at most 128 advisor-approved semester credit hours of study. This must include

0-12 credits of 400-level courses

36-54 credits of 500- and 600- level courses. Among them, at most 6 credits come from outside the Computer Science Department of IIT. Credits from CS 595 are allowed.

6-12 credits of CS 597 (Reading and Special Problems)

24-48 credits of CS 691 (Research/Thesis for Ph.D. degree). Note CS 691 can only be taken after passing the Ph.D. qualifying exam.

1 credit of Ph.D. seminar

Notice that no credits will be given to accelerated courses (700-level courses). No credits are given to courses in which the student earns a grade of C or below. The student may have to take some other courses as required by the advisor. The credit hours of 500-level coursework must include five core courses with at least one each from Groups I, II, and III described below and at least one course from each of two different groups from among Groups IV, V, and VI. Advanced courses may be substituted after approval of the department. The groups of core courses are:

Group I: Theory of Computation

CS 530	Theory of Computation
CS 533	Computational Geometry
CS 535	Design and Analysis of Algorithms
CS 538	Combinatorial Optimization

Group II: Systems

CS 546	Parallel Processing
CS 550	Advanced Operating Systems
CS 570	Advanced Computer Architecture

Group III: Programming Languages

CS 536	Science of Programming
CS 541	Topics in Compiler Construction
CS 545	Distributed Computing Landscape

Group IV: Networks

CS 542 Computer Networks I: Fundamentals
CS 544 Computer Networks II: Network Services

Group V: Databases

CS 525 Advanced Database Organization

Group VI: Software Engineering

CS 586 Software Systems Architectures

M.S. Exit from Program

Students wishing to leave the direct Ph.D. program with the degree of Master of Science in Computer Science must satisfy all the requirements of the Master's degree and either write an M.S. thesis or pass the Ph.D. qualifying examination.

Ph.D. Qualifying Examination

The Ph.D. Qualifying Exam has two parts: a written examination and an oral examination. The written examination is used to judge a student's breadth of knowledge; the oral examination is used to judge a student's research potential. The first attempt in oral examination and the written examination must be taken no later than a student's 5th semester. The second attempt must be taken no later than a student's 6th semester. These requirements hold for both full-time and part-time students

The written examination is divided into three, independent "area" examinations. To pass the written examination, a student must pass all the area examinations. Extra coursework cannot be used in lieu of passing an area examination. Area examinations can be taken in the same or different semesters. A student who fails an area examination can retake the area examination, but only once. See the computer science webpage for more detail of qualifying examinations.

Comprehensive (Research Proposal) Examination

The purpose of the Comprehensive Examination is to ensure that the candidate has the background to carry out successful research in the chosen area and that the research problem is properly formulated and has sufficient scholarly merit. The student (in concert with the student's research advisor) must develop a written research proposal containing a literature review, a proposed research topic, and a program of research based upon this topic, and then present the proposal orally as well.

The student must request appointment of an examination committee using Form #301. The examination committee may consist of from four to seven members. It must include at least three full-time faculty members from the Computer Science Department and one full-time faculty member from another department in the university. Other committee members from inside or outside the university may be chosen. The student should consult with his/her research advisor concerning the makeup of the committee.

Thesis Defense

Each student must present an oral defense of his/her Ph.D. Thesis. The Thesis Review Committee is appointed in much the same way as the Ph.D. Comprehensive Examination Committee. It will examine the written thesis and examine the student during the oral defense. All Ph.D. Thesis Defenses are open to the public.

Requirements for students entering with an M.S. Degree

54 credit hours
Qualifying exam
Comprehensive exam
Dissertation and oral defense

Admission Requirements

The applicants should have an M.S degree in computer science or related fields. Admission to the program is competitive and depends on a student's GRE score and it is expected that applicants will have a high grade-point average. For non-English speaking applicants without a U.S. degree, a minimum TOEFL score of 550 is required. If the TOEFL score is less than 600, the applicant is required to take the English Proficiency Examination administered by the IIT Humanities Department. Applicants must submit three letters of recommendation.

Program Requirements

If the student has an M.S. degree in computer science, the program requires the student to complete at least 54 adviser-approved semester credit hours of study. This must include

0-12 credits of 400-level courses

18-30 credits of 500- and 600-level courses. Amongst them, at most 6 credits come from outside the computer science department. Credits from CS 595 are allowed.

3-12 credits of CS 597 (Reading and Special Problems)

24-48 credits of CS 691 (Research/Thesis for Ph.D. degree). Note CS 691 can only be taken after passing the Ph.D. qualifying exam.

1 credit of Ph.D. seminar

Notice that no credits will be given to accelerated courses (700-level courses). No credits are given to courses in which the student earns a grade of C or below. The student may have to take some other courses as required by the adviser. The credit hours of 500-level coursework must include three core courses with at least one each from Groups I, II, and III described before. Advanced courses may be substituted after approval of the department.

If the student has an M.S. degree in a field other than computer science, the program requires the student to complete at least 60 advisor-approved semester credit hours of study. This must include

0-12 credits of 400-level courses

24-30 credits of 500- and 600-level courses. Among them, at most 6 credits may come from outside the computer science department. Credits from CS595 are allowed.

3-12 credits of CS 597 (Reading and Special Problems)

24-48 credits of CS 691 (Research/Thesis for the Ph.D. degree). Note CS 691 can only be taken after passing the Ph.D. qualifying exam.

1 credit for CS 695 (the Ph.D. seminar)

Notice that no credits will be given to accelerated courses (700-level courses). No credits are given to courses in which the student earns a grade of C or below. The student may have to take some other courses as required by the adviser. The credit hours of 500-level coursework must include five core courses with at least one each from Groups I, II, and III described before and at least two courses from two different groups from among Groups IV, V, and VI. Advanced courses may be substituted after approval of the department.

Ph.D. Qualifying Examination

The Ph.D. Qualifying Examination has two parts: a written examination and an oral examination. The written exam is used to judge a student's breadth of knowledge; the oral exam is used to judge a student's research potential. The first attempt at the oral examination and the written examination must be made no later than a student's 3rd semester. The second attempt must be made no later than a student's 4th semester. These requirements hold for both full-time and part-time students.

The written examination is divided into three, independent "area" examinations. To pass the written examination, a student must pass all the area examinations. Extra coursework cannot be used in lieu of passing an area examination. Area examinations can be taken in the same or different semesters. A student who fails an area examination can retake the area examination, but only once. See the computer science web page for more details about the qualifying examinations.

Comprehensive (Research Proposal) Examination

The purpose of the Comprehensive Examination is to ensure that the candidate has the background to carry out successful research in the chosen area and that the research problem is properly formulated and has sufficient scholarly merit. The student (in concert with the student's research adviser) must develop a written research proposal containing a literature review, a proposed research topic, and a program of research based upon this topic, and then present it orally as well.

The student must request appointment of an examination committee on Form #301. The examination committee may consist of from four to seven members. It must include at least three full-time faculty members from the Computer Science Department and one full-time faculty member from another department in the university. Other committee members from inside or outside the university may be chosen. The student should consult with his/her research adviser concerning the makeup of the committee

Thesis Defense

Each student must present an oral defense of his/her Ph.D. Thesis. The Thesis Review Committee is appointed in much the same way as the Ph.D. Comprehensive Examination Committee. It will examine the written thesis and examine the student during the oral defense. All Ph.D. Thesis Defenses are open to the public.

Certificate Programs

Certificate programs offer working professionals an opportunity to increase their knowledge and skills in the specific areas of computer science. A certificate representing proven academic performance is presented after the required coursework is completed with a GPA of 3.0/4.0. Courses taken may be later applied toward an M.S. degree program. Applicants should have a bachelor's degree from an accredited college or university. The degree need not be in computer science.

Computer Networking and Telecommunications

Students should select nine credits from the following:

CS 455	Data Communications
CS 542	Computer Networks I: Fundamentals
CS 544	Computer Networks II: Network Services
CS 548	Broadband Networks

Information Systems

The student in this program must complete nine hours of coursework from the following:

CS 425	Database Organization
CS 521	Object-Oriented Analysis and Design
CS 525	Advanced Database Organization

Software Engineering

The student in this program must complete nine hours of coursework from the following list:

CS 445	Object-Oriented Design and Programming
CS 487	Software Engineering
CS 537	Software Metrics
CS 586	Software Systems Architecture
CS 587	Software Project Management
CS 588	Advanced Software Engineering Development
CS 589	Software Testing and Analysis
CS 750	Computer-Aided Software Engineering
CS 763	Automated Software Testing

Accelerated Courses

The department offers accelerated courses for credit in several areas of computer science. These courses go beyond traditional core topics and are designed for working professionals who are interested in keeping abreast of rapidly changing technologies. Accelerated courses provide an opportunity for degree-seeking students at IIT to complete M.S. degree requirements in a shorter time period. If taken by non-degree students, these courses can be applied towards requirements for an M.S. degree at IIT.

Course Descriptions

Numbers in parentheses indicate class, lab and credit hours, respectively.

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, Non-photorealistic rendering Particle systems, Character animation, Physically based modeling and animation. Prerequisite: CS 411. (3-0-3)

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: CS 430. (3-0-3)

CS 520
Database Design and Engineering
Overview of database architectures, including the Relational, Hierarchical, Network and Object Models. Database normalization and design. Implementation issues for database management systems, including the processing and parsing of SQL queries, query optimization, integrity, and concurrency control. Distributed and parallel databases and data warehouse issues are addressed. Solid programming skills are required. Implementation of a database management engine prototype is required. No credit is given for both this course and CS 425. Prerequisite: CS 351 or CS 402 and CS 430. (3-0-3)

CS 521
Object-Oriented Analysis and Design
This course describes a methodology that covers a wide range of 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, frameworks, 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. Prerequisite: CS 425. (3-0-3)

CS 522
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: CS 422. (3-0-3)

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 specialists, and the designers and developers of client/server and distributed systems. Focus is on transaction management, database structures and distributed processing. Prerequisite: CS 425. (3-0-3)

CS 529
Information Retrieval
Continued exploration of information retrieval algorithms. Topics will include: Text classification, meta-search, mediators, semi-structured information retrieval, name search, etc. Students will continuously study new contributions to the field. A large project will be required that encourages students to push the limits of existing information retrieval. Prerequisites: CS 429. (3-0-3)

CS 530
Theory of Computation
Computability topics such as Turing machines, nondeterministic machines, undecidability results, 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. Prerequisites: CS 430. (3-0-3)

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: CS 430. (3-0-3)

CS 532
Formal Languages
This course provides an introduction to the theory of formal languages and machines. Topics to be covered include: strings, alphabets, and languages; grammars, the Chomsky Hierarchy of languages and corresponding machines (regular sets and finite automata, context free languages of various types, Turing machines and recursive functions, undecidable problems), and computational complexity, polynomial-time reductions, NP-completeness. Prerequisite: CS 430. (3-0-3)

CS 533
Computational Geometry
The course covers fundamental algorithms and data structures for Convex Hulls, Voronoi Diagrams, Delauney Triangulation, Euclidean Spanning Trees, Point Location, 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. Prerequisites: CS 430 and a linear algebra course. (3-0-3)

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. Prerequisites: CS 430. (3-0-3)

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 CS 331 or CS 401. (3-0-3)

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: CS 487. (3-0-3)

CS 538
Combinatorial Optimization
Linear programs and their properties. Efficient algorithms for linear programming. Network flows, minimum cost flows, maximum matchings, weighted matchings, and matroids. Prerequisite: CS 430 and a linear algebra course. (3-0-3)

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. Prerequisites: CS 440. (3-0-3)

CS 541
Topics in Compiler Construction
Advanced topics in compiler construction, including incremental and interactive compiling, error correction, code optimization and models of code generation. The objective of the course is to provide an in-depth coverage of compiler optimization techniques, including both classical optimizations and areas of current interest in compiler research. Prerequisite: CS 440. (3-0-3)

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: CS 455. (3-0-3)

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. Prerequisites: CS 542 or ECE 545. (3-0-3)

CS 545
Distributed Computing Landscape
Introduction to the theory of concurrent programming languages. Topics include formal methods of concurrent computation as process algebra, nets and actors, high-level concurrent programming languages and their operational semantics, methods for reasoning about correctness and the complexity of concurrent programs. Prerequisite: CS 450. (3-0-3)

CS 546
Parallel and Distributed Processing
General issues of parallel and distributed processing including systems, programming, performance evaluation and application of parallel and distributed computers. The influence of communication and parallelism on algorithm design. Prerequisites: CS 430, CS 450. (3-0-3)

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 regula-

tions, 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 the students' own choosing. Prerequisite: CS 455. (3-0-3)

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 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: CS 455. (3-0-3)

CS 549
Cryptography and Network Security
This course provides an introduction to the theory and the 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: CS 430. (3-0-3)

CS 550
Advanced Operating Systems
Advanced operating system design concepts, such as multimedia OS, multiprocessor systems, virtual memory management, process migration, process scheduling, synchronization, file systems. Study of systems highlighting these concepts. Prerequisite: CS 450. (3-0-3)

CS 551
Operating System Design and Implementation
This course covers in detail the design and implementation of processes, interprocess communication, semaphores, monitors, message passing, remote procedure calls, scheduling algorithm, input/output, device drivers, memory management, file system design, network file servers, atomic transactions, security and protection mechanisms. The hardware-software interface is examined in detail. Students modify and extend a multi-user operating system. Prerequisites: CS 450. (3-0-3)

CS 552
Distributed Systems
Advanced distributed system design concepts, such as distributed processes and memory management, distributed file systems, consistency and fault tolerance, security and transaction system structures, and distributed programming. Programming in representative distributed environments. Prerequisite: CS 450. (3-0-3)

CS 553
Pervasive Computing
Concepts in computing that create an ubiquitous environment, combining processors and sensors with network technologies (wireless and otherwise) and intelligent software. Issues of middleware and middleware development, including mobility, context awareness, resource discovery, cyber-foraging, agents, QOS, P2P, web services as well as other pervasive and ubiquitous technologies. Prerequisites: CS 450 or CS 455 and 470. (3-0-3)

CS 555
Analytic Models and Simulation of Computer Systems
Analytic and simulation techniques for the performance analysis of computer architecture, operating systems and subsystems. Rigorous development of queuing models. Study of simulation languages and models. Prerequisites: CS 450. (3-0-3)

CS 560
Computer Science in the Classroom
Preparation and formulation of computer science courses. Detailed weekly materials organized and perfected. The goal being to develop Open Course Ware (OCW). (1-4-3)

CS 561
The Computer and Curriculum Content
Emphasis on the presentation concepts. Selecting the best mode of delivery and using the power of the web page to enhance the presentation. (1-4-3)

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: CS 560 or CS 561. (3-0-3)

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 use of computer in the classroom. Prerequisite: CS 560 or CS 561. (1-4-3)

CS 570
Advanced Computer Architecture
Advanced computer system design and architecture, such as pipelining and instruction-level parallelism, memory-hierarchy system, interconnection networks, multiprocessors, and clusters of servers. Selected study on current experimental computer systems. Prerequisite: CS450 and CS470. (3-0-3)

CS 572
Advanced Topics in Computer Architecture
Current problems in computer architecture. Prerequisite: CS 570. (3-0-3)

CS 580
Medical Informatics
This course provides an introduction to computer applications in health care with an emphasis on the contributions of artificial intelligence and database analysis. Topics will include medical expert systems, medical decision analysis, reasoning under uncertainty, medical tutoring systems, medical language processing, medical record systems, hospital and office information systems, laboratory, pharmacy, radiology and bibliographic information retrieval systems. Presentations and papers or projects will be required. Prerequisites: CS 425, CS 480. (3-0-3)

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: CS 480. (3-0-3)

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 sensors, 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: CS 430. (3-0-3)

CS 583
Expert Systems
Medical expert systems and clinical decision analysis. Decision making under uncertainty. Inference mechanisms for expert systems: production rules, pattern recognition, automatic theorem proving, fuzzy logic. Expert system architecture. Knowledge engineering. Information retrieval. Natural language interfaces. Prerequisite: CS 480. (3-0-3)

CS 584
Machine Learning
Covers basic algorithms and techniques used in Machine Learning, to give the student a good basis for work in this highly relevant field. Topics include: Version space learning, Computational learning theory, PAC-learning, VC-dimension, On-line learning, Winnow, Perceptrons, Neural Networks, Backpropagation, Genetic algorithms, Bayesian learning, Experimental design, Decision-tree learning, Covering algorithms for learning rule sets, Minimum description length, Clustering algorithms, Reinforcement learning, Markov decision processes. Prerequisite: CS 480. (3-0-3)

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: CS 445. (3-0-3)

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: CS 487. (3-0-3)

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: CS 487. (3-0-3)

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: CS 487. (3-0-3)

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: CS 487. (3-0-3)

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: Written consent of instructor. (3-0-3)

CS 591
Research and Thesis for M.S. Degree
(Credit: Variable)

CS 595
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. May be taken more than once. Prerequisite: Written consent of instructor. (Credit: Variable)

CS 597
Reading and Special Problems
Prerequisite: Written consent of instructor. May be taken more than once. (Credit: Variable)

CS 612
Topics in Computer Vision
Cover advanced topics in computer vision to enhance the 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: CS 512. (3-0-3)

CS 630
Advanced Topics in Algorithms
Theoretical analysis of various types of algorithms. Topics vary, and may include quantum, approximation, on-line, distributed, randomized, and parallel algorithms. Prerequisite: CS 430 and consent of instructor. (3-0-3)

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: CS 542. (3-0-3)

CS 681
Topics in Computational Linguistics
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: CS 585. (3-0-3)

CS 689
Advanced Topics in Software Engineering
Course content is variable and reflects the current trends in software engineering. Prerequisite: Consent of instructor. (3-0-3)

CS 691
Research and Thesis for Ph.D. Degree
(Credit: Variable)

CS 695
Doctoral Seminar
(1-0-1)

Courses available for the Master of Computer Science Program

CSP 527
Client-Server Applications Development
Through hands-on experience in developing a client-server database project and developing and managing a client-server Internet project, this course teaches advanced skills for effective design and implementation of client-server applications. Students will examine the architectural and functionality decisions, technologies, configurations, languages, and techniques associated with client-server systems. Active/passive client-server technologies, as well as public, enterprise-wide, and inter-enterprise approaches to decision and operation support are discussed and implemented. Prerequisite: CS 425. (3-0-3)

CSP 541
Internet Technologies
This course focuses on the technologies and protocols used by Internet WAN's and LAN's. The fundamental architecture, organization, and routing principles of the Internet are described. Part of the course will focus on emerging Internet technologies. Prerequisite: CS 455. (3-0-3)

CSP 542
Internet Design and Analysis
This course examines the principles of network design. The design process is studied from requirements gathering to deployment. The student will gain experience in estimating application load, network sizing, component choice, and protocol choice. Internetworking between popular components and protocols will be studied. Analytical and simulation techniques are described and used to design several local- and wide-area networks. Prerequisite: CS 455. (3-0-3)

CSP 543
Multimedia Networking
This course covers the architectures, protocols, and design issues for multimedia networks. Topics covered include coding, compression, streaming, synchronization, QoS, and adaptation. Current tools for multimedia networking will be surveyed. Issues with multimedia application development will be explored. Students will design and develop multimedia applications. Prerequisites: CS 455 and experience programming in high-level languages. (3-0-3)

CSP 544
System and Network Security
This course will present an in-depth examination of topics in data and network security, such as: Access control, authentication, security assessment, network and data security tools, and security policies. A significant hands-on component includes network incidents to detect and fix. Prerequisites: CS 430, CS 455. (3-0-3)

CSP 545
Wireless Networking Technologies and Applications
This course will present the foundation of wireless technologies and examine state-of-the-art wireless systems, services, network technologies, and security. Prerequisites: CS 542. (3-0-3)

CSP 550
Internet Programming
This course discusses current fundamental concepts and development techniques for distributed applications. Topics covered include multithreaded programs, sockets, message-passing systems, remote method invocation and procedure calls, peer-to-peer networks, and underlying technologies for internet applications. Prerequisite: CS 455. (3-0-3)

CSP 551
Advanced UNIX Programming
This course provides students a hands-on introduction to UNIX programming topics such as standard application programmer interfaces, concurrent programming, UNIX processes and threads, shell programming, UNIX interprocess communications, client-server designs, and application portability. Prerequisites: C programming, CS 450 or equivalent, and user-level knowledge of UNIX. (3-0-3)

CSP 581
Applied AI Programming
To learn AI programming algorithms and techniques in Common Lisp. Time is split between Common Lisp topics and discussions of implementation strategies for AI algorithms. Prerequisite: CS 440 or equivalent. (3-0-3)

CSP 585
Object-Oriented Design Patterns
This course introduces the principles of design patterns for Object-Oriented software systems. A catalog of design patterns is shown, to illustrate the roles of patterns in designing and contracting complex software systems. The catalog of design patterns also provides a pragmatic reference to a well-engineered set of existing patterns currently in use. Also discussed is the impact of post-object-oriented software development on design patterns. Prerequisite: CS 445. (3-0-3)

CSP 586
Software Modeling and Development with UML
Students will obtain a significant exposure to the UML technology. This will include exposure to modeling, model-driven development, executable models, and round-trip engineering. These technologies will be explained at the application level. Prerequisite: CS 487 or CS 445. (3-0-3)

CSP 587
Software Quality Management
Students will learn methods of software quality management. This will include exposure to software quality assurance, quality measures, and quality control. These quality management methods will be explained at the applications level. Prerequisite: CS 487 or equivalent. (3-0-3)

Fundamental Prerequisite Courses in Computer Science

Students whose background in computer science is deficient are required to take the following courses and earn a grade "B" or better. The credits for these courses may not be used for any degree program in computer science, computer information systems or computer engineering.

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. The course covers advanced programming concepts including pointers, recursion and inheritance, as well as data structures and algorithms including linked lists, stacks, queues, heaps, graphs, and sorting techniques. Prerequisite: CS 201 or equivalent. (3-0-3)

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. The course consists of two parts. The first part introduces the students to computer organization and architecture, numbering systems, logical operations and fundamentals of assembly language programming. The second part of the course is devoted to event driven programming concepts using Java as the programming tool. Multithreading programming concepts are also covered using Java. Prerequisite: CS 401 or consent of instructor. (3-0-3)

Undergraduate Courses Available to Graduate Students in Computer Science

CS 411
Computer Graphics

CS 422
Introduction to Data Mining

CS 425
Database Organization

CS 429
Introduction to Information Retrieval

CS 430
Introduction to Algorithms

CS 440
Programming Languages and Translators

CS 441
Current Topics in Programming Languages

CS 445
Object-Oriented Design and Programming

CS 447
Distributed Objects

CS 450
Introduction to Operating Systems

CS 455
Data Communications

CS 458
Information Security

CS 470
Computer Architecture I

CS 471
Design of Computer Processors

CS 480
Artificial Intelligence

CS 485
Computers and Society

CS 487
Software Engineering