

COMPUTER SCIENCE

Stuart Building, Room 235
10 W. 31st St.
Chicago, IL 60616
312.567.5150
info@cs.iit.edu
science.iit.edu/computer-science

Interim Chair

Shlomo Argamon

Associate Chairs

Cynthia Hood
Bogdan Korel

Faculty with Research Interests

For information regarding faculty visit the Department of Computer Science website.

Computers have changed what we do and how we do it in our homes, in our offices, and throughout our world. The discipline of computer science focuses upon the many challenging problems encountered in the development and use of computing systems and software. Areas of study in computer science range from theoretical analyses into the nature of computing and computing algorithms, through the development of advanced computing devices and networks, to the design and implementation of sophisticated software systems.

The department offers two undergraduate programs in core computer science: a Bachelor of Science (B.S.) in Computer Science and Bachelor of Science (B.S.) in Computer Information Systems, as well as a Bachelor of Science (B.S.) in Artificial Intelligence.

Both core computer science programs provide an excellent background in the field and allow for ample study in other areas. Where these programs differ is in the approach they take to computer science. The B.S. in Computer Science provides an in-depth experience focusing on the theory and practice of computer science while the B.S. in Computer Information Systems provides a more interdisciplinary experience, balancing study in computer science with study in another field. Both programs begin with a set of introductory courses that work together to provide students with a firm foundation in computer science. These introductory courses include regular labs wherein students analyze and solve a variety of problems using accepted software development techniques. Having completed the introductory core, a student is prepared to work independently within a well-structured design and coding paradigm in the classroom or on the job.

The last two years of study build upon this foundation. The B.S. in Computer Science focuses on the concepts and techniques used in the design and development of advanced software systems. Students in this program explore the conceptual underpinnings of computer science—its fundamental algorithms, programming languages, database systems, operating systems, and software engineering techniques. In addition, students choose from a rich set of electives including artificial intelligence, machine learning, biometrics, cloud computing, data mining, natural language processing, computer vision, computer graphics, information retrieval, information security, intelligent text analysis, knowledge management systems, and mobile application development, among others. As with the introductory sequence, these advanced courses stress “hands-on” learning by doing. Students can opt to complete one of the computer science specializations: data science, distributed and cloud computing, information security, or information and knowledge management systems. An allotment of free electives allows students to combine study in computer science with study in another field to complete a minor.

The B.S. in Computer Information Systems program emphasizes the use of computing for complex problem solving. Students in this program pursue an interdisciplinary course of study that combines a solid foundation in computer science with a focus in another discipline. This program is designed for students who seek to blend their computer science abilities with skills specific to another domain to solve problems in that domain. Examples include computing with a business focus (e.g., management information systems) or computing with a natural science focus (e.g., computational physics).

The B.S. in Artificial Intelligence (AI), one of the first degrees of its kind in the country, gives students the broad and deep foundations necessary to build computational systems that can interpret sensory input, learn from experience, understand human language, plan ahead, and support intelligent decision making. The program starts with introductory courses in programming, computer science, mathematics, and statistics that provide students with a firm technical foundation. Students are then prepared to learn core AI concepts and techniques including state-space search, game-playing, machine learning, neural networks, planning, computer vision, and language understanding. For breadth of understanding necessary in this interdisciplinary field, students choose from a diverse set of advanced courses in cognitive science, philosophy of mind, and linguistics, and also must complete a minor in another area. Given the societal impact of AI, emphasis is placed on ethics and social responsibility throughout the coursework. The program also includes a large variety of advanced AI electives to enable students to develop technical mastery in specific sub-fields.

The mission statement for the Department of Computer Science may be found on the Department of Computer Science website.

Degree Programs

- Bachelor of Science in Artificial Intelligence
- Bachelor of Science in Computer Information Systems
- Bachelor of Science in Computer Science

Co-Terminal Options

The Department of Computer Science also offers the following co-terminal degrees, which enables a student to simultaneously complete both an undergraduate and graduate degree in as few as five years:

- Bachelor of Science in Applied Mathematics/Master of Computer Science
- Bachelor of Science in Applied Mathematics/Master of Science in Computer Science
- Bachelor of Science in Biology/Master of Computer Science
- Bachelor of Science in Biology/Master of Science in Computer Science
- Bachelor of Science in Computer Engineering/Master of Computer Science
- Bachelor of Science in Computer Engineering/Master of Science in Computer Science
- Bachelor of Science in Computer Science/Master of Science in Applied Mathematics
- Bachelor of Science in Computer Science/Master of Computer Science
- Bachelor of Science in Computer Science/Master of Science in Computer Science
- Bachelor of Science in Computer Science/Master of Data Science
- Bachelor of Science in Computer Science/Master of Intellectual Property Management and Markets
- Bachelor of Science in Physics/Master of Computer Science
- Bachelor of Science in Physics/Master of Science in Computer Science

These co-terminal degrees allow students to gain greater knowledge in specialized areas while, in most cases, completing a smaller number of credit hours with increased scheduling flexibility. For more information, please visit the Department of Computer Science website (science.iit.edu/computer-science).

Minors

- Minor in Artificial Intelligence
- Minor in Computational Structures
- Minor in Computer Architecture
- Minor in Computer Networking
- Minor in Computer Science
- Minor in Database Management
- Minor in Operating Systems
- Minor in Programming Languages
- Minor in Software Engineering

Course Descriptions

CS 100

Introduction to the Profession

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

Lecture: 1 Lab: 2 Credits: 2

Satisfies: Communications (C)

CS 104

Introduction to Computer Programming for Engineers

Introduces the use of high-level programming language as a problem-solving tool in engineering including basic data structures and algorithms, structured programming techniques, and software documentation. Designed for students who have had little or no prior experience with computer programming. Students should only take one of these courses (CS 104, CS 105, CS 115).

Lecture: 2 Lab: 1 Credits: 2

CS 105

Introduction to Computer Programming

Introduces the use of high-level programming language as a problem-solving tool, including basic data structures and algorithms, structured programming techniques, and software documentation. Designed for students who have had little or no prior experience with computer programming. Students should only take one of these courses (CS 104, CS 105, CS 115).

Lecture: 2 Lab: 1 Credits: 2

CS 110

Computing Principles

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

Lecture: 2 Lab: 1 Credits: 2

CS 115

Object-Oriented Programming I

Introduces the use of a high-level object-oriented programming language as a problem-solving tool, including basic data structures and algorithms, object-oriented programming techniques, and software documentation. Designed for students who have had little or no prior experience with computer programming. For students in CS and CS-related degree programs. Students should only take one of these courses (CS 104, CS 105, CS 115).

Lecture: 2 Lab: 1 Credits: 2

CS 116

Object-Oriented Programming II

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

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

Lecture: 2 Lab: 1 Credits: 2

Satisfies: Communications (C)

CS 201

Accelerated Introduction to Computer Science

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

Prerequisite(s): CS 104 with min. grade of C or CS 105 with min. grade of C or CS 110 with min. grade of C or CS 115 with min. grade of C

Lecture: 3 Lab: 2 Credits: 4

Satisfies: Communications (C)

CS 330

Discrete Structures

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

Prerequisite(s): CS 201 or CS 116

Lecture: 3 Lab: 1 Credits: 3

Satisfies: Communications (C)

CS 331

Data Structures and Algorithms

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

Prerequisite(s): CS 116 or CS 201

Lecture: 3 Lab: 1 Credits: 3

CS 340

Programming Paradigms and Patterns

This class balances the imperative, object-oriented bent of the introductory programming sequence by presenting alternative programming paradigms and asks students to write complex programs from scratch while amassing a repertoire of reusable programming patterns and techniques. Programming assignments, drawn from various domains of computer science, will highlight the importance of selecting appropriate data structures, algorithms, and techniques for the problem at hand.

Prerequisite(s): CS 116 or CS 201

Lecture: 3 Lab: 0 Credits: 3

CS 350**Computer Organization and Assembly Language Programming**

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

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

Lecture: 3 Lab: 1 Credits: 3

CS 351**Systems Programming**

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

Prerequisite(s): (CS 331 and CS 350) or (CS 331 and ECE 242)

Lecture: 3 Lab: 1 Credits: 3

CS 397**Special Projects**

Instructor permission required.

Credit: Variable

CS 401**Introduction to Advanced Studies I**

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

Prerequisite(s): CS 200 or CS 201

Lecture: 2 Lab: 2 Credits: 3

CS 402**Introduction to Advanced Studies II**

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

Prerequisite(s): CS 401

Lecture: 2 Lab: 2 Credits: 3

CS 403**Foundations to Advanced Studies**

This course is a six-credit hour integration of CS 401 and CS 402. The course is an introduction to data structures but is designed to expand programming skills/concepts using software development methodology techniques. The course also provides an introduction to computer architecture and systems programming including assembly language programming, event handling and multi-threading.

Lecture: 4 Lab: 4 Credits: 6

CS 406**Introduction to Discrete Structures and Algorithms**

This course will provide students with an understanding of some basic discrete mathematics and techniques for designing computer algorithms and measuring and analyzing their behavior. We emphasize the necessary mathematical ideas such as how to prove a mathematical statement, how to analyze the worst case and the average case complexity of an algorithm. We also introduce students to some new techniques used in designing algorithms, such as the approximation algorithms and randomized algorithms.

Lecture: 3 Lab: 0 Credits: 3

CS 411**Computer Graphics**

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

Prerequisite(s): CS 331 or CS 401 or CS 403

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 422**Data Mining**

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

Prerequisite(s): CS 331 or CS 401 or CS 403

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 425**Database Organization**

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

Prerequisite(s): CS 331 or CS 401 or CS 403

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 429**Information Retrieval**

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

Prerequisite(s): CS 331 or CS 401

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 430**Introduction to Algorithms**

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

Prerequisite(s): (CS 331 and CS 330) or (CS 331 and MATH 230) or CS 401 or CS 403

Lecture: 3 Lab: 1 Credits: 3

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

CS 440**Programming Languages and Translators**

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

Prerequisite(s): (CS 330 and CS 331) or (MATH 230 and CS 331) or CS 401 or CS 403

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 442**Mobile Applications Development**

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

Prerequisite(s): (CS 331 or CS 401) and (CS 351* or CS 402*), An asterisk (*) designates a course which may be taken concurrently.

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 443**Compiler Construction**

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

Prerequisite(s): CS 440

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 445**Object Oriented Design and Programming**

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

Prerequisite(s): CS 331 or CS 401 or CS 403

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 447**Distributed Objects**

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

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 450**Operating Systems**

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

Prerequisite(s): CS 351 or (CS 401 and CS 402) or CS 403

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

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

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

Prerequisite(s): CS 351 or CS 450

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 455**Data Communications**

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

Prerequisite(s): CS 450

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

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

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

Prerequisite(s): CS 350 or ECE 242 or (CS 401 and CS 402) or CS 403

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 458**Introduction to Information Security**

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

Prerequisite(s): CS 425 or CS 450 or CS 455

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 470**Computer Architecture**

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

Prerequisite(s): CS 350 or ECE 242

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 480**Introduction to Artificial Intelligence**

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

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

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 481**Artificial Intelligence Language Understanding**

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

Prerequisite(s): (CS 331 or CS 401 or CS 403) and MATH 474

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 482**Information and Knowledge Management Systems**

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

Prerequisite(s): (CS 425 and CS 422 and CS 429) or (CS 425 and CS 422 and CS 481) or (CS 425 and CS 429 and CS 481)

Lecture: 3 Lab: 0 Credits: 3

Satisfies: CS Technical Elective (T)

CS 485**Computers and Society**

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

Prerequisite(s): COM 421 or COM 424 or COM 425 or COM 428 or COM 435

Lecture: 3 Lab: 0 Credits: 3

Satisfies: Communications (C)

CS 487**Software Engineering I**

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

Prerequisite(s): (CS 331 or CS 401 or CS 403) and CS 425

Lecture: 3 Lab: 0 Credits: 3

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

CS 491**Undergraduate Research**

Instructor permission required.

Credit: Variable

CS 492

Introduction to Computer Science Research

Prepares undergraduate computer science majors for conducting research.

Lecture: 1 Lab: 0 Credits: 1

CS 495

Topics in Computer Science

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

Credit: Variable

CS 497

Special Projects

Special projects.

Credit: Variable