COMPUTER SCIENCE

Director of undergraduate studies: Stanley Eisenstat, 208 AKW, 432-1246, stanley.eisenstat@yale.edu

FACULTY OF THE DEPARTMENT OF COMPUTER SCIENCE

Professors Dana Angluin, James Aspnes, Julie Dorsey, Stanley Eisenstat, Joan Feigenbaum, Michael Fischer, David Gelernter, Paul Hudak, Drew McDermott, Vladimir Rokhlin, Holly Rushmeier (Chair), Brian Scassellati, Martin Schultz (Emeritus), Zhong Shao, Avi Silberschatz, Daniel Spielman, Y. Richard Yang, Steven Zucker

Associate Professor Daniel Abadi

Assistant Professor Bryan Ford

Lecturer Brad Rosen

The Department of Computer Science offers both B.S. and B.A. degree programs, as well as combined majors with the Departments of Electrical Engineering (see Electrical Engineering and Computer Science), Mathematics (see Computer Science and Mathematics), and Psychology (see Computer Science and Psychology). Each major program not only provides a solid technical education but also allows students either to take a broad range of courses in other disciplines or to complete the requirements of a second major.

The Computer Science and combined major programs share a common core of five computer science courses. The first is CPSC 201, Introduction to Computer Science, a survey that demonstrates the breadth and depth of the field to students who have taken the equivalent of an introductory programming course. The remaining core courses cover discrete mathematics; data structures; systems programming and computer architecture; and algorithm analysis and design. Together these courses include the material that every major should know.

The core courses are supplemented by electives (and, for the joint majors, core courses in the other discipline) that offer great flexibility in tailoring a program to each student’s interests. The capstone is the senior project, through which students experience the challenges and rewards of original research under the guidance of a faculty mentor.

Prospective majors are encouraged to discuss their programs with the director of undergraduate studies as early as possible.

Introductory courses The department offers a broad range of introductory courses to meet the needs of students with varying backgrounds and interests. With the exception of CPSC 201, none assumes previous knowledge of computers.

1. CPSC 079b examines the methods used to define shapes, materials, and lighting in computer-generated images. Students use a modeling/rendering system to create an animated video with rich visual effects. Proficiency in high school–level mathematics is assumed.

2. CPSC 101b introduces nonmajors to some of the central ideas in computer science, including algorithms, elementary programming, hardware, complexity, and representation of information.

3. CPSC 112a or b teaches students majoring in any subject area how to program a computer and solve problems using the language Java. Students with previous programming experience should consider taking CPSC 201 instead.

4. CPSC 150a explores how some of the key ideas in computer science have affected philosophy of mind, cognitivism, connectionism, and related areas. This humanities-style course has significant readings and a paper, and satisfies the writing or the humanities and arts distributional requirement.

5. CPSC 151b studies the history of the graphical user interface in an attempt to guess its future. This course satisfies the writing distributional requirement.

6. CPSC 183a explores the myriad ways that law and technology intersect, with a special focus on the role of cyberspace. This course satisfies the social science distributional requirement.

7. CPSC 201a or b surveys the field of computer science, including systems (computers and their languages) and theory (algorithms, complexity, and computability). Students with sufficient programming experience may elect CPSC 201 without taking CPSC 112. (These courses meet at the same time so that students are easily able to change levels if necessary.)

8. CPSC 202a presents the formal methods of reasoning and the concepts of discrete mathematics and linear algebra used in computer science and related disciplines.

Requirements of the major The B.S. and the B.A. degree programs have the same required core courses: CPSC 201; CPSC 202a or MATH 244a; CPSC 223b, 323a, 365b, and 490. The B.S. degree program requires six additional intermediate or advanced courses in Computer Science, for a total of twelve; the B.A. degree program, four, for a total of ten. CPSC 480 and 490 may not be counted toward these electives. All courses in the major must be taken for a letter grade.

Students majoring in Computer Science are advised to complete CPSC 201 and 223 by the end of the sophomore year.
For students who already know how to program, typical B.S. programs starting in the freshman and sophomore years are:

**Freshman**

<table>
<thead>
<tr>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPSC 201a</td>
</tr>
<tr>
<td>CPSC 202a</td>
</tr>
<tr>
<td>CPSC 223b</td>
</tr>
</tbody>
</table>

**Sophomore**

<table>
<thead>
<tr>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPSC 323a</td>
</tr>
<tr>
<td>CPSC 365b</td>
</tr>
</tbody>
</table>

**Junior**

<table>
<thead>
<tr>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two electives</td>
</tr>
</tbody>
</table>

**Senior**

<table>
<thead>
<tr>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPSC 490a</td>
</tr>
</tbody>
</table>

Students considering graduate study in computer science are advised to take CPSC 421 and 422, as well as courses covering the breadth of computer science, including programming languages and systems, artificial intelligence, scientific computing, and theoretical computer science.

For typical B.A. programs, two of the electives would be omitted.

**Electives** The Computer Science department encourages interdisciplinary study in which computer science plays a major role. Advanced courses in other departments that involve concepts from computer science and are relevant to an individual program may, with permission of the director of undergraduate studies, be counted toward the requirements.

Students interested in using computers to solve scientific and engineering problems are advised to take CPSC 440 as well as computational courses offered in Applied Mathematics and in Engineering and Applied Science.

The core mathematical background necessary to complete the Computer Science major is provided in CPSC 202. However, many advanced courses in graphics, computer vision, neural networks, and numerical analysis assume additional knowledge of linear algebra and calculus. Students who plan to take such courses as electives and who are unsure whether they have the appropriate mathematical background are encouraged to take MATH 222 or 225 and MATH 120.

**Senior requirement** In the senior year students must take CPSC 490, an independent project course in which students select an adviser to guide them in research in a subfield of computer science. With permission of the director of undergraduate studies, students may enroll in 490 more than once or before their senior year.

**Schedule approval** All Computer Science majors in the sophomore, junior, and senior years should have their programs approved by the director of undergraduate studies.

**Combined B.S./M.S. degree program in Computer Science** Exceptionally able and well-prepared students may complete a course of study leading to the simultaneous award of the B.S. and M.S. degrees after eight terms of enrollment. Eligibility requirements are described under "Simultaneous Award of the Bachelor’s and Master’s Degrees" under "Special Arrangements" in the Academic Regulations. Specific requirements for the combined degree in Computer Science are as follows:

1. Candidates must satisfy the Yale College requirements for the B.S. degree in Computer Science.
2. In fulfilling these requirements, students must complete eight graduate courses from the approved list, up to two of which may, with the permission of the director of undergraduate studies and the director of graduate studies, also be applied toward completion of the B.S. degree. At most one of these eight courses may be CPSC 690, 691, or 692.
3. At the end of their fifth term of enrollment students must have achieved at least three-fourths A or A– grades in all of their course credits directly relating to the major.

**REQUIREMENTS OF THE MAJOR**

**Prerequisites** None

**Number of courses** B.S. – 12 term courses taken for letter grades (incl senior project); B.A. – 10 term courses taken for letter grades (incl senior project)

**Specific courses required** B.S. and B.A. – CPSC 201; CPSC 202 or MATH 244; CPSC 223, 323, 365

**Distribution of courses** B.S.– 6 addtl intermediate or advanced Comp Sci courses; B.A. – 4 addtl intermediate or advanced Comp Sci courses

**Substitution permitted** Advanced courses in other depts, with DUS permission

**Senior requirement** Senior project (CPSC 490)
Introductory Courses

[ CPSC 079, Digital Photorealism ]

**CPSC 112a or b, Introduction to Programming**  Daniel Abadi and staff
Development on the computer of programming skills, problem-solving methods, and selected applications. No previous experience with computers necessary.  QR

*CPSC 150a / HUMS 407a, Computer Science and the Modern Intellectual Agenda*  David Gelernter
Introduction to the basic ideas of computer science (computability, algorithm, virtual machine, symbol processing system), and of several ongoing relationships between computer science and other fields, particularly philosophy of mind. No previous experience with computers necessary. Enrollment limited to 25.  WR, HU

*CPSC 151b / HUMS 408b, The Graphical User Interface*  David Gelernter
The role of graphical user interfaces (GUIs) on standard platforms such as desktop PCs, laptops, and small-screen devices. Discussion of how and why GUIs developed as they did, why they have evolved so little since the desktop computers of the 1970s, and how changing hardware and user requirements might reshape them in the future. Enrollment limited to 25.  WR

**CPSC 183a, Law, Technology, and Culture**  Brad Rosen
An exploration of the myriad ways in which law and technology intersect, with a special focus on the role of cyberspace. Topics include digital copyright, free speech, privacy and anonymity, information security, innovation, online communities, the impact of technology on society, and emerging trends. No previous experience with computers or law necessary.  SO

*CPSC 185b, Control, Privacy, and Technology*  Brad Rosen
The evolution of various legal doctrines with and around technological development. Topics include criminal law, privacy, search and seizure, digital rights, and the implications of technologically permitted methods of control on the law. Special attention to case law and policy. After CPSC 183.  WR, SO

**CPSC 201a or b, Introduction to Computer Science**  Dana Angluin
Introduction to the concepts, techniques, and applications of computer science. Topics include computer systems (the design of computers and their languages); theoretical foundations of computing (computability, complexity, algorithm design); and artificial intelligence (the organization of knowledge and its representation for efficient search). Examples stress the importance of different problem-solving methods. After CPSC 112 or equivalent.  QR
Math: Stat/Applied Math

**CPSC 202a, Mathematical Tools for Computer Science**  James Aspnes
Introduction to formal methods for reasoning and to mathematical techniques basic to computer science. Topics include propositional logic, discrete mathematics, and linear algebra. Emphasis on applications to computer science: recurrences, sorting, graph traversal, Gaussian elimination.  QR

**CPSC 223b, Data Structures and Programming Techniques**  Stanley Eisenstat
Topics include programming in C; data structures (arrays, stacks, queues, lists, trees, heaps, graphs); sorting and searching; storage allocation and management; data abstraction; programming style; testing and debugging; writing efficient programs. After CPSC 201 or equivalent.  QR  RP

*CPSC 290a or b, Directed Research*  Stanley Eisenstat
Individual research. Requires a faculty supervisor and the permission of the director of undergraduate studies. May be taken more than once for credit.

**MATH 244a / AMTH 244a, Discrete Mathematics**  Staff
Basic concepts and results in discrete mathematics: graphs, trees, connectivity, Ramsey theorem, enumeration, binomial coefficients, Stirling numbers. Properties of finite set systems. Recommended preparation: MATH 115 or equivalent.  QR
Math: Stat/Applied Math

Intermediate Courses

**CPSC 323a, Introduction to Systems Programming and Computer Organization**  Stanley Eisenstat
Machine architecture and computer organization, systems programming in a high-level language, issues in operating systems, software engineering, prototyping in scripting languages. After CPSC 223.  QR  RP

**CPSC 365b, Design and Analysis of Algorithms**  Daniel Spielman
Paradigms for problem solving: divide and conquer, recursion, greedy algorithms, dynamic programming, randomized and probabilistic algorithms. Techniques for analyzing the efficiency of algorithms and designing efficient algorithms and data structures. Algorithms for graph theoretic problems, network flows, and numerical linear algebra. Provides algorithmic background essential to further study of computer science. After CPSC 202 and 223.  QR
Math: Stat/Applied Math
Advanced Courses

*CPSC 421bG, Compilers and Interpreters  Zhong Shao
Compiler organization and implementation: lexical analysis, formal syntax specification, parsing techniques, execution environment, storage management, code generation and optimization, procedure linkage and address binding. The effect of language-design decisions on compiler construction. After CPSC 323. QR

CPSC 422bG, Operating Systems  Bryan Ford
The design and implementation of operating systems. Topics include synchronization, deadlock, process management, storage management, file systems, security, protection, and networking. After CPSC 323. QR

*CPSC 423aG, Principles of Operating Systems  Avi Silberschatz
A survey of the underlying principles of modern operating systems. Topics include process management, memory management, storage management, protection and security, distributed systems, and virtual machines. Emphasis on fundamental concepts rather than implementation. After CPSC 323.

[ CPSC 424, Parallel Programming Techniques ]

CPSC 426aG, Building Decentralized Systems  Bryan Ford
Challenges and techniques for building decentralized computing systems, in which many networked computers need to cooperate reliably despite failures and without assuming centralized management. Topics include decentralized storage systems, mobile and remote execution, hosting untrusted code, fault tolerance, naming, capabilities, information flow control, distributed shared memory, distributed hash tables, content distribution, and practical uses of cryptography. After CPSC 323. QR

[ CPSC 427, Object-Oriented Programming ]

CPSC 430aG, Formal Semantics  Zhong Shao
Introduction to formal approaches to programming language design and implementation. Topics include lambda calculus, type theory, denotational semantics, type-directed compilation, higher-order modules, and application of formal methods to systems software and Internet programming. After CPSC 202 and 323. QR

[ CPSC 431, Computer Music: Algorithmic and Heuristic Composition ]

CPSC 432bG, Computer Music: Sound Representation and Synthesis  Paul Hudak
Study of the theoretical and practical fundamentals of computer-generated music, with a focus on low-level sound representation, acoustics and sound synthesis, scales and tuning systems, and programming languages for computer music generation. Theoretical concepts are supplemented with pragmatic issues expressed in a high-level programming language. Ability to read music is assumed. After CPSC 202 and 223.

CPSC 433aG, Computer Networks  Y. Richard Yang
An introduction to the design, implementation, analysis, and evaluation of computer networks and their protocols. Topics include layered network architectures, applications, transport, congestion, routing, data link protocols, local area networks, performance analysis, multimedia networking, network security, and network management. Emphasis on protocols used in the Internet. After CPSC 323. QR

[ CPSC 434, Mobile Computing and Wireless Networking ]

CPSC 437bG, Introduction to Databases  Avi Silberschatz

[ CPSC 438, Database System Implementation and Architectures ]

CPSC 439bG, Software Engineering  Ruzica Piskac
Introduction to fundamental concepts in software engineering and to the development and maintenance of large, robust software systems. The process of collecting requirements and writing specifications; project planning and system design; methods for increasing software reliability, including delta debugging and automatic test-case generation; type systems, static analysis, and model checking. Students build software in teams. After CPSC 323. QR RP

CPSC 440bG, Numerical Computation  Vladimir Rokhlin
Algorithms for numerical problems in the physical, biological, and social sciences: solution of linear and nonlinear systems of equations, interpolation and approximation of functions, numerical differentiation and integration, optimization. After CPSC 112 or an equivalent introductory programming course; MATH 120; and MATH 222 or 225 or CPSC 202. QR
Math: Stat/Applied Math
CPSC 445a, Introduction to Data Mining  Vladimir Rokhlin
A study of algorithms and systems that allow computers to find patterns and regularities in databases, to perform prediction and forecasting, and to improve their performance generally through interaction with data. After CPSC 202, 223, and MATH 222, or equivalents.  QR

*CPSC 457a, Sensitive Information in a Wired World  Joan Feigenbaum
Issues of ownership, control, privacy, and accuracy of the huge amount of sensitive information about people and organizations that is collected, stored, and used by today's ubiquitous information systems. Readings consist of research papers that explore both the power and the limitations of existing privacy-enhancing technologies such as encryption and "trusted platforms." After or concurrently with CPSC 365 and 467.  QR

*CPSC 462a / AMTH 462a, Graphs and Networks  Daniel Spielman
A mathematical examination of graphs and their applications in the sciences. Families of graphs include social networks, small-world graphs, Internet graphs, planar graphs, well-shaped meshes, power-law graphs, and classic random graphs. Phenomena include connectivity, clustering, communication, ranking, and iterative processes. Prerequisites: linear algebra and discrete mathematics; a course in probability is recommended.  QR

CPSC 465b, Theory of Distributed Systems  James Aspnes
Models of asynchronous distributed computing systems. Fundamental concepts of concurrency and synchronization, communication, reliability, topological and geometric constraints, time and space complexity, and distributed algorithms. After CPSC 323 and 365.  QR

CPSC 467a, Cryptography and Computer Security  Michael Fischer
A survey of such private and public key cryptographic techniques as DES, RSA, and zero-knowledge proofs, and their application to problems of maintaining privacy and security in computer networks. Focus on technology, with consideration of such societal issues as balancing individual privacy concerns against the needs of law enforcement, vulnerability of societal institutions to electronic attack, export regulations and international competitiveness, and development of secure information systems. Some programming may be required. After CPSC 202 and 223.  QR

[ CPSC 468, Computational Complexity ]
[ CPSC 469, Randomized Algorithms ]

*CPSC 471b, Advanced Topics in Artificial Intelligence  Drew McDermott
An in-depth study of one area of artificial intelligence. Topics vary from year to year. The topic for 2013–2014 is artificial intelligence and philosophy of mind. After CPSC 470 or with permission of instructor.  WR

CPSC 472a, Intelligent Robotics  Brian Scassellati
Introduction to the construction of intelligent, autonomous systems. Sensory-motor coordination and task-based perception. Implementation techniques for behavior selection and arbitration, including behavior-based design, evolution-based design, dynamical systems, and hybrid deliberative-reactive systems. Situated learning and adaptive behavior. After CPSC 201 and 202 or equivalents. May not be taken after CPSC 473.

*CPSC 473b, Intelligent Robotics Laboratory  Brian Scassellati
Students work in small teams to construct novel research projects using one of a variety of robot architectures. Project topics may include human-robot interaction, adaptive intelligent behavior, active perception, humanoid robotics, and socially assistive robotics. Enrollment limited to 20. After CPSC 472.

CPSC 475a / BENG 475a, Computational Vision and Biological Perception  Steven Zucker
An overview of computational vision with a biological emphasis. Suitable as an introduction to biological perception for computer science and engineering students, as well as an introduction to computational vision for mathematics, psychology, and physiology students. After CPSC 112 and MATH 120, or with permission of instructor.  QR, SC, RP

CPSC 476b / BENG 476b, Advanced Computational Vision  Steven Zucker
Advanced view of vision from a mathematical, computational, and neurophysiological perspective. Emphasis on differential geometry, machine learning, visual psychophysics, and advanced neurophysiology. Topics include perceptual organization, shading, color and texture analysis, and shape description and representation. After CPSC 475.  QR, SC

[ CPSC 478, Computer Graphics ]

CPSC 479b, Advanced Topics in Computer Graphics  Holly Rushmeier
An in-depth study of advanced algorithms and systems for rendering, modeling, and animation in computer graphics. Topics vary and may include reflectance modeling, global illumination, subdivision surfaces, NURBS, physically-based fluids systems, and character animation. After CPSC 202 and 223.  QR
*CPSC 480a or b, Directed Reading  Stanley Eisenstat
Individual study for qualified students who wish to investigate an area of computer science not covered in regular courses. A student must be sponsored by a faculty member who sets the requirements and meets regularly with the student. Requires a written plan of study approved by the faculty adviser and the director of undergraduate studies. May be taken more than once for credit.

*CPSC 490a or b, Special Projects  Stanley Eisenstat
Individual research. Requires a faculty supervisor and the permission of the director of undergraduate studies. The student must submit a written report about the results of the project. May be taken more than once for credit.