Computer science is one of the youngest and most exciting scientific disciplines. Computer scientists study computational processes (known as algorithms)—their formal properties, their hardware and software implementations, their languages, and their applications. Computer science is a scientific discipline, using rigorous techniques to derive principles and experimentation to test hypotheses. It is also a mathematical discipline, since it borrows many of its notations and methodologies from mathematics.

The Department of Computer Science at Bowdoin offers major and minor programs, as well as an interdisciplinary major with mathematics. These programs support the fundamental liberal arts philosophy that emphasizes breadth and depth of study, critical analysis of ideas, exposure to different modes of inquiry, a mature style of writing and other forms of discourse, and multicultural awareness through off-campus study. At the same time, they provide a solid foundation for either postgraduate study or a career in industry.

In the first two years, computer science majors develop foundational knowledge of the principles of the field. Later, they focus on advanced topics and applications in more depth. Majors have the opportunity, through electives and independent study projects, to customize their studies to emphasize particular areas of computer science that interest them. Computer science faculty members at Bowdoin actively pursue research in a variety of subjects, including the analysis of algorithms, artificial intelligence (especially planning and robotics), cognitive science, cryptography, and programming languages. Students are encouraged to take independent study courses that will engage them in research projects with faculty mentors and give them a wide-ranging view of the world of computer science.

The department also enjoys a strong sense of community among students and faculty, fostered by small class sizes and frequent co-curricular activities (lunches, informal talks and discussions, outside speakers, and other social events) throughout the year. During the summer, some of our majors remain at Bowdoin to work on research projects (supported by various Bowdoin fellowships or outside grants) or in a technical capacity in the Computing Center.

In this liberal arts setting, students have a unique opportunity to investigate the many strong connections between computer science and other related fields, such as mathematics, philosophy, biochemistry, psychology, economics, art, and languages. For example, the adaptation of computers to model human behavior (one of the goals of artificial intelligence) involves not only a mastery of computer science but also an understanding of psychology.

### Facilities

To support computer science coursework and research, three laboratories are available. One lab contains a network of Macs, which are used primarily by the introductory and data structures courses. A second lab contains a network of Macintosh workstations and is managed by student lab assistants. The third is a robotics lab, and is used for student projects and ongoing faculty research.

The first two laboratories provide access to a wide range of programming languages and other software, including C, C++, Java, Lisp, Scheme, Haskell, Prolog, MPI, LogicWorks, LaTeX, and a variety of mathematical, typesetting, and database tools. They provide computer science students with the modern technologies and software environments that they will encounter in graduate school or the computing profession.

The robotics lab contains a variety of state-of-the-art robots, including Pioneer AT-DXs and Sony Aibos. Students have the opportunity to work on independent projects using the robots and to participate in the ongoing robot-related research of the faculty. One such project is called RoboCup, a competition involving robotic soccer. Bowdoin’s student-led team finished in the top ten in the recent world championships. The department makes extensive use of robots throughout the curriculum.

### The Curriculum

The computer science curriculum at Bowdoin serves a wide range of student interests. Computing: Tools and Issues is an introductory course for students who are not likely to major or minor in computer science but would like a solid foundation to augment a major in another field. Students who may pursue a major or minor in computer science take Introduction to Computer Science, which provides a basic intro-
dution to problem solving and computers. Both courses are open to all students at the College, and each has a slightly different focus.

The intermediate and advanced courses in computer science provide depth and breadth of study across the discipline. At the intermediate level, core courses offered are Data Structures, Algorithms, Programming Languages, and Theory of Computation. At the advanced level, courses offered include Cryptography and Network Security, GIS Algorithms and Data Structures, Cognitive Architecture, Artificial Intelligence, Optimization and Uncertainty, and a variety of independent study, honors, and advanced projects on topics such as robotics, parallel computing, natural language processing, software design, compilers, and others, as determined by faculty and student interests.

The Major and Minor Programs

Any student may elect a minor in computer science, a major in computer science, or an interdisciplinary major in computer science and mathematics. For majors, Introduction to Computer Science and Data Structures are required and should be taken first. Students who have taken an Advanced Placement course in secondary school may be able to obtain Bowdoin course credit and advanced placement with a score of 4 or 5 on the AP exam.

The computer science major consists of nine computer science courses (Introduction to Computer Science, Foundations of Computing or Introduction to Mathematical Reasoning, the four intermediate "core" courses, and any three elective courses numbered 260 and above) and two courses outside the department (from a range of courses providing mathematical foundations). Depending on student and faculty interests, one of the computer science electives may be an independent study, honors, or advanced projects course.

The computer science minor consists of the introductory course followed by four additional courses: Data Structures, and any three other intermediate-level computer science courses.

Completion of the interdisciplinary major in computer science and mathematics also requires eleven courses, six from computer science and five from mathematics. For more details about this major, consult the College Catalogue.

Independent Study, Honors, and Student/Faculty Research

Advanced students are encouraged to pursue an independent study project, which is often done in conjunction with an ongoing faculty research project. Opportunities for independent projects in fields with strong interdisciplinary ties to computer science are also encouraged.

Independent study provides a special opportunity to work closely with a faculty member on a topic of mutual interest. Participation in faculty research provides an excellent introduction to the kind of independent thinking that is required in both graduate study and in the rapidly advancing computing profession itself. Honors projects can lead to co-authorship of a paper and the opportunity to attend an international conference.

Graduation with honors in computer science requires a one-year project that includes a thesis.

Faculty

Eric L. Chown, associate professor of computer science, B.A., M.S. (Northwestern), Ph.D. (Michigan), specializes in artificial intelligence and cognitive science. Besides teaching computer science courses in artificial intelligence, cognitive architecture, and computer programming, he also leads Bowdoin's RoboCup team and does research on human learning. He is chair of the department.

Stephen M. Majercik, assistant professor of computer science, A.B. (Harvard), M.F.A., M.B.A. (Yale), M.S. (Southern Maine), Ph.D. (Duke), specializes in artificial intelligence and stochastic planning. His research interests include planning and reasoning under uncertainty, and human-computer collaboration.

Adriana Palacio, assistant professor of computer science, B.S. (Universidad de los Andes–Bogotá), M.S., Ph.D. (California–San Diego), specializes in cryptography and information security.

Laura I. Toma, assistant professor of computer science, B.S., M.S. (Universitatea Politehnica Bucuresti), M.S., Ph.D. (Duke), specializes in large-scale data analysis and algorithms, with emphasis on Geographic Information Systems.

Allen B. Tucker, Jr., Anne T. and Robert M. Bass Professor of Natural Sciences Emeritus, A.B. (Wesleyan), M.S., Ph.D. (Northwestern), specializes in programming languages and natural language processing. He is the author or co-author of books and publications in the areas of programming languages, natural language processing, and computer science education.

After Bowdoin

Bowdoin's curricular offerings in computer science prepare graduates well for further study or professional careers in teaching, industry, and financial services. Recent computer science majors have entered graduate programs at Brown, Carnegie Mellon, Dartmouth, Indiana, Michigan, Northwestern, Penn, Stanford, UCLA, Utah, and Washington.

Many computer science graduates have taken positions at firms such as Amazon.com, American Management Systems, Chase Manhattan, Goldman Sachs, Meditech, Coopers and Lybrand, Liberty Mutual, Sun Microsystems, and a variety of others inside and outside the technology industry.

Others have followed different short- and long-term career paths, such as the Peace Corps, medicine, law, neuroscience, physics, and music composition.

Honors Projects

Recent honors theses completed by majors in the department include:

- Developing an Autonomous Robot Soccer Team at an Undergraduate Institution
- 1/O-Efficient Shortest Paths on Grid-Based Terrains
- Chunking: A Modified Dynamic Programming Approach to Solving Stochastic Satisfiability Problems
- 1/O-Efficient Refinement of Triangulated Terrains
- Robot Localization and Abstract Mapping in Dynamic Environments
- Recognizing Useful Advice and Using it Efficiently in a Reinforcement Learning Framework