COMPUTER SCIENCE

Overview & Learning Goals

Overview

The major in computer science focuses on critical thinking and problem-solving and is designed to blend background in traditional core areas (theory, systems, and artificial intelligence), as well as emerging areas such as human-computer interaction, social and economic networks, nature-inspired computation, distributed systems, and resource-efficient algorithms.

Department/Program Website (https://www.bowdoin.edu/computer-science/)

Learning Goals

A student graduating from Bowdoin with a computer science major will:

1. have critical thinking skills enabling the solution of problems by developing and analyzing algorithms;
2. have a variety of skills enabling the design, implementation, debugging, and testing of complex problems using a programming language;
3. have experience working on a large computer science project;
4. be able to connect the use of computer science to other disciplines and have the experience of working on at least one project that does this directly;
5. have a capstone experience with current research in computer science, including reading the literature, learning advanced material independently, and working on a research project under the supervision of a faculty member;
6. have experience working as part of a team;
7. be able to recognize, identify, and analyze the social and ethical issues that arise from the use of computer science techniques in society; and
8. have experience presenting technical content in both oral and written form.

Faculty

Stephen M. Majercik, *Department Chair*
Suzanne M. Theberge, *Senior Department Coordinator*

Professor: Laura I. Toma
Associate Professor: Stephen M. Majercik
Assistant Professors: Sean K. Barker*, Cibele Freire, Sarah M. Harmon, Mohammad T. Irfan (Digital and Computational Studies)
Visiting Faculty: Allen Harper, Stephen Houser

Faculty/Staff Website (https://www.bowdoin.edu/computer-science/faculty-and-staff/)

Requirements

Computer Science Major

The major in computer science consists of at least ten courses.

Code	Title	Credits
Required Courses		
CSCI 2101	Data Structures	1
CSCI 2200	Algorithms	1
CSCI 2330	Foundations of Computer Systems	1
Select at least one course from each of the following areas, for a total of six computer science electives, at least three of which must be advanced-level courses numbered 3000 or higher.		6
Algorithms/Theory		
CSCI 2210	Theory of Computation	
CSCI 3210	Computational Game Theory	
CSCI 3225	GIS Algorithms and Data Structures	
CSCI 3250	Computational Geometry	
Artificial Intelligence		
CSCI 2400	Artificial Intelligence	
CSCI 3400	Cognitive Architecture	
CSCI 3415	Robotics	
CSCI 3420	Optimization and Uncertainty	
CSCI 3445	Nature-Inspired Computation	
CSCI 3725	Computational Creativity	
CSCI 3455	Machine Learning	
Systems		
CSCI 2320	Principles of Programming Languages	
CSCI 2300	Computer Organization	
CSCI 2505	Mobile Computing	
CSCI 3300	Computer Networks	
CSCI 3310	Operating Systems	
CSCI 3325	Distributed Systems	
Projects Courses		
CSCI 2505	Mobile Computing	
CSCI 3225	GIS Algorithms and Data Structures	
CSCI 3325	Distributed Systems	
CSCI 3415	Robotics	
CSCI 3445	Nature-Inspired Computation	
CSCI 3455	Machine Learning	
CSCI 3715	Human-Computer Interaction	
CSCI 3725	Computational Creativity	
CSCI 3735	Playable Media	
Select one mathematics course numbered 1000 or higher.		1

One independent study at the intermediate or advanced level may be applied toward the required number of computer science courses, but cannot be used to fulfill any other requirements (areas, projects, or 3000 level).

Most students begin the major with CSCI 1101 Introduction to Computer Science, followed by CSCI 2101 Data Structures. CSCI 1101 Introduction to Computer Science has no prerequisites and requires no prior knowledge of computer science. Students who wish to start with a less programming-oriented introduction to the field may choose to take CSCI 1055 The Digital World, followed by CSCI 1103 Programming with Data, followed by CSCI 2101 Data Structures. Students with prior knowledge of computer science can place out of CSCI 1101 Introduction to Computer Science and start with CSCI 1103 Programming with Data or with CSCI 2101 Data Structures.

Students—particularly those who intend to do graduate work in computer science—are encouraged to collaborate with faculty on research projects

through independent studies, honors projects, and fellowship-funded summer research.

Computer Science Minor

The minor in computer science consists of CSCI 2101 Data Structures, plus at least three courses, numbered 2000 or higher. Independent studies in computer science cannot count toward the minor.

Interdisciplinary Major

The department participates in an interdisciplinary major program in computer science and mathematics. See the Interdisciplinary Majors (https://catalogue.bowdoin.edu/departments-programs/interdisciplinary-majors/).

Additional Information

Additional Information and Department Policies

- Each of the courses required for the major or minor must be taken for a regular letter grade (not Credit/D/Fail) with a minimum earned grade of C-.
- The prerequisite for 2000-level courses is a grade of C or better on the final in CSCI 1101 Introduction to Computer Science or CSCI 1103 Programming with Data. For courses at the 2000 level or above, a grade of C- or better must be earned in the course for it to serve as a prerequisite for another computer science course.
- At most two of the nine computer science courses required for the major, or one of the four computer science courses required for the minor, can be transfer credit from other institutions.
- Majors may double-count one course with another department or program. Minors may not double-count any courses with another department or program.
- Advanced Placement and International Baccalaureate scores, in addition to the computer science placement test, are only used for placement.

Courses

CSCI 1055 (a, MCSR) The Digital World
Every Spring. Enrollment limit: 30.

Explores the means and the results of the digital revolution. Questions how information is coded and stored; how it can be safeguarded. Considers how the widespread coding and transmission of data impact issues such as privacy and intellectual property. Examines these topics through the study and use of techniques from computer science, such as programming and cryptography. Closed to students with credit for any course in computer science or digital and computational studies.

Previous terms offered: Spring 2018, Spring 2017.

CSCI 1101 (a, MCSR) Introduction to Computer Science
Sarah Harmon; Cibele Freire.
Every Semester. Fall 2020. Enrollment limit: 30.

What is computer science, what are its applications in other disciplines, and what is its impact in society? A step-by-step introduction to the art of problem solving using the computer and programming. Provides a broad introduction to computer science and programming through real-life applications. Weekly labs provide experiments with the concepts presented in class. Assumes no prior knowledge of computers or programming. Final examination grade must be C or better to serve as a prerequisite for Computer Science 2101.

Previous terms offered: Spring 2020, Fall 2019, Spring 2019, Fall 2018, Spring 2018, Fall 2017, Spring 2017, Fall 2016.

CSCI 1103 (a, MCSR) Programming with Data
Allen Harper.
Every Fall. Fall 2020. Enrollment limit: 30.

Intended for students with some programming experience, but not enough to move directly into Data Structures. An accelerated introduction to the art of problem solving using the computer and the Python programming language. Weekly labs and programming assignments focus on 'big data' and its impact on the world. (Same as: DCS 1300)

Prerequisites: CSCI 1055 or DCS 1100 or DCS 1200 or Placement in above CSCI 1101.

Previous terms offered: Fall 2019, Fall 2018, Fall 2017.

CSCI 2101 (a, MCSR) Data Structures
Allen Harper; Stephen Majercik.
Every Semester. Fall 2020. Enrollment limit: 22.

Solving complex algorithmic problems requires the use of appropriate data structures such as stacks, priority queues, search trees, dictionaries, hash tables, and graphs. It also requires the ability to measure the efficiency of operations such as sorting and searching in order to make effective choices among alternative solutions. Offers a study of data structures, their efficiency, and their use in solving computational problems. Laboratory exercises provide an opportunity to design and implement these structures. Students interested in taking Computer Science 2101 are required to pass the computer science placement examination with a grade of C or better before class starts.

Prerequisites: CSCI 1101 or Placement in above CSCI 1101 or CSCI 1103.

Previous terms offered: Spring 2020, Fall 2019, Spring 2019, Fall 2018, Spring 2018, Fall 2017, Spring 2017, Fall 2016.

CSCI 2200 (a, MCSR) Algorithms
Laura Toma.
Every Semester. Fall 2020. Enrollment limit: 22.

An introductory course on the design and analysis of algorithms. Introduces a number of basic algorithms for a variety of problems such as searching, sorting, selection, and graph problems (e.g., spanning trees and shortest paths). Discusses analysis techniques, such as recurrences and amortization, as well as algorithm design paradigms such as divide-and-conquer, dynamic programming, and greedy algorithms.

Prerequisites: CSCI 2101.

Previous terms offered: Spring 2020, Fall 2019, Spring 2019, Fall 2018, Spring 2018, Fall 2017, Spring 2017, Fall 2016.

CSCI 2210 (a, MCSR) Theory of Computation
Cibele Freire.
Non-Standard Rotation. Fall 2020. Enrollment limit: 30.

Studies the nature of computation and examines the principles that determine what computational capabilities are required to solve particular classes of problems. Topics include an introduction to the connections between language theory and models of computation, and a study of unsolvable problems.

Prerequisites: CSCI 2101.

Previous terms offered: Spring 2019.

CSCI 2320 (a, MCSR) Principles of Programming Languages
Every Year. Enrollment limit: 30.

Studies design principles and paradigms of programming languages. Different paradigms of languages correspond to distinct ways of thinking about problem solving. For example, functional languages (such as Haskell) focus attention on the behavioral aspects of a problem; object-oriented languages (such as Ruby) focus attention on data–how to model and manipulate it. Despite their differences, a common set of principles often guide language design. Covers principles of language design and implementation including syntax, semantics, type systems, control structures, and compilers. Also covers various paradigms of languages including imperative, object-oriented, web, and functional languages.

Prerequisites: CSCI 2101.

Previous terms offered: Spring 2020, Fall 2017.

CSCI 2330 (a, MCSR) Foundations of Computer Systems
Stephen Houser.
Every Semester. Fall 2020. Enrollment limit: 22.

A broad introduction to how modern computer systems execute programs, store information, and communicate. Examines the hardware and software components required to go from a program expressed in a high-level programming language like C to the computer actually running the program. Topics include concepts of program compilation and assembly, machine code, data representation and computer arithmetic, basic microarchitecture, the memory hierarchy, processes, and system-level I/O. Regular, programming-intensive projects provide hands-on experience with the key components of computer systems.

Prerequisites: CSCI 2101.

Previous terms offered: Spring 2020, Fall 2019, Spring 2019, Fall 2018, Spring 2018, Spring 2017.

CSCI 2345 (a, MCSR) In situ D4: Real-world Database Design, Development, and Deployment
Non-Standard Rotation. Enrollment limit: 22.

This project-based course approaches database systems management from the perspective of database designers, developers, data analysts, and diverse sets of users. Topics include relational and non-relational databases (SQL/NoSQL), data modeling, transactions and isolation, and web-based information retrieval applications. Includes both individual programming assignments and a multidimensional, semester-long project culminating in student research and demonstration of a real-world information systems application. In 2019, the research project will focus on designing databases and information retrieval interfaces for the purpose of navigating public spaces and increasing multimodal information access for users with blindness or low vision constraints. The course will also provide opportunities for ongoing student research in the development of accessibility technologies after the completion of the course.

Prerequisites: CSCI 2330.

Previous terms offered: Spring 2020.

CSCI 2350 (a) Social and Economic Networks
Mohammad Irfan.
Non-Standard Rotation. Fall 2020. Enrollment limit: 22.

Examines the social and economic aspects of today's connected world from a multitude of perspectives; namely, network science, computer science, sociology, and economics. The fundamental questions to be addressed are: What are the properties of real-world networks? What are the effects of networks on our behavioral choices like quitting smoking or eating healthy? How do cascades in networks lead to outcomes like videos going viral? How does Google search the Internet and make money doing so? Debates issues around centrality in networks. Uses game theory to study strategic interactions in networks and markets. (Same as: DCS 2350)

Prerequisites: DCS 1100 or CSCI 1101 or CSCI 1103 (same as DCS 1300).

Previous terms offered: Fall 2019, Spring 2018.

CSCI 2400 (a, MCSR) Artificial Intelligence
Stephen Majercik.
Every Year. Fall 2020. Enrollment limit: 35.

Explores the principles and techniques involved in programming computers to do tasks that would require intelligence if people did them. State-space and heuristic search techniques, logic and other knowledge representations, reinforcement learning, neural networks, and other approaches are applied to a variety of problems with an emphasis on agent-based approaches.

Prerequisites: CSCI 2101.

Previous terms offered: Fall 2018, Fall 2017, Fall 2016.

CSCI 2500 (a) Computing, Ethics, and Society
Every Year. Enrollment limit: 16.

Explores ethical and moral case studies associated with advances in computing, artificial intelligence, and emerging technologies. An examination of various codes of ethics for computing professional societies and their limitations in addressing the complexity of evolving technologies is a central focus. Students investigate current issues using an interdisciplinary approach. Course topics include but are not limited to: net neutrality, information privacy and data harvesting, algorithmic bias, autonomous vehicles, intellectual property, networked communications, cybersecurity, government and privacy, workforce disruptions, and professional conduct in a diverse tech workplace. Course materials integrate foundational literature in the field of computer ethics, as well as contemporary sources of public dialogue regarding the ethical conduct of computing and technology development.

Prerequisites: CSCI 2101.

Previous terms offered: Fall 2018.

CSCI 2505 (a) Mobile Computing
Non-Standard Rotation. Enrollment limit: 35.

As computer science enters the post-PC era, basic computing paradigms are shifting to take advantage of mobile platforms such as phones and tablets. Covers all aspects of programming for mobile devices including development environments, programming languages, the use of touch screens for input, and associated sensors such as accelerometers and GPS. Students engage in a series of introductory projects before taking on a large self-designed term project that highlights the differences between mobile applications and more traditional applications designed for the desktop.

Prerequisites: CSCI 2330.

Previous terms offered: Fall 2016.

CSCI 2600 (a, MCSR) Introduction to Data Mining
Non-Standard Rotation. Enrollment limit: 30.

Data mining addresses the issues of designing and utilizing computational techniques for discovering nontrivial patterns in large data sets. Introduces the core goals and methodologies of data mining, as well as some basic approaches for inferring patterns in data. Regular programming assignments provide hands-on experience with concepts

Prerequisites: CSCI 2330.

Previous terms offered: Spring 2017.

CSCI 3210 (a) Computational Game Theory
Mohammad Irfan.
Every Year. Fall 2020. Enrollment limit: 16.

Advanced algorithms course with a focus on game theory. Topics include computational complexity, linear programming, approximation algorithms, and algorithms for solving games. Game theory, also known as the mathematical theory of strategic interactions, rose to prominence due to its applicability to a variety of strategic scenarios ranging from markets and auctions to kidney exchanges to social influence. These scenarios often involve complex interactions in large-scale systems, giving rise to many computational questions, including: how algorithms for certain games are devised; how local interactions lead to global outcomes; how individual choices, such as selfishness, impact outcomes.

Prerequisites: CSCI 2200.

Previous terms offered: Fall 2019, Spring 2018, Spring 2017.

CSCI 3225 (a) GIS Algorithms and Data Structures
Every Year. Enrollment limit: 16.

Geographic information systems (GIS) handle geographical data such as boundaries of countries; course of rivers; height of mountains; and location of cities, roads, railways, and power lines. GIS can help determine the closest public hospital, find areas susceptible to flooding or erosion, track the position of a car on a map, or find the shortest route from one location to another. Because GIS deal with large datasets, making it important to process data efficiently, they provide a rich source of problems in computer science. Topics covered include data representation, triangulation, range searching, point location, map overlay, meshes and quadtrees, terrain simplification, and visualization.

Prerequisites: Two of: CSCI 2330 and CSCI 2200.

Previous terms offered: Fall 2017, Fall 2016.

CSCI 3250 (a) Computational Geometry
Every Year. Enrollment limit: 22.

Computational Geometry refers to the study of geometric problems from a computational point of view, with focus on the design and analysis of algorithms for problems involving collections of points, lines and polygons. Computational Geometry emerged as a field driven by geometric problems in graphics and robotics, and its list of applications has continued to grow to areas such as pattern recognition, graph drawing, surface simplification and meshing, crystallography, molecular simulation, planning and autonomous vehicles. Class covers some of the basic concepts and fundamental geometric problems, such as: convex hulls, art gallery and visibility problems, geometric searching with range trees and kd-trees, intersection problems, proximity problems, point and polygon triangulation, and motion planning. Requirements include regular, programming-intensive projects.

Previous terms offered: Spring 2018, Spring 2017.

CSCI 3300 (a) Computer Networks
Non-Standard Rotation. Enrollment limit: 16.

Computer networks are everywhere: e-mail, the Web, wireless networks, mobile devices, networked sensors, satellite communication, peer-to-peer applications. New applications based on networks appear constantly. Provides an introduction to the exciting field of computer networks by taking a top-down approach. Begins with an overview of computer networks, hardware and software components, the Internet, and the concept of protocols and layered service. Delves into details about the four main layers making up the computer network stack: Application (HTTP, FTP, e-mail, DNS, peer-to-peer applications and socket programming), Transport (TCP, UDP, and congestion control), Network (IP, routers, and routing algorithms) and Link Layer and Local Area Networks (medium access control, switches, and Ethernet). Also covers wireless and mobile networks (CDMA, WiFi, cellular internet access, mobile IP, and managing mobility).

Prerequisites: CSCI 2330.

Previous terms offered: Fall 2019, Fall 2018, Fall 2017, Fall 2016.

CSCI 3310 (a) Operating Systems
Every Other Year. Enrollment limit: 16.

Explores the design and implementation of computer operating systems, which provide a well-known, convenient, and efficient interface between user programs and the underlying computer hardware. The operating system is responsible for sharing resources such as processors, memory, and disks, as well as providing common services needed by many different programs. Topics include process and thread management, synchronization and concurrency, memory management, I/O and file systems, and virtual machines. Intensive programming projects involve implementing key components of operating systems and provide exposure to design principles used in many different types of computer systems.

Prerequisites: CSCI 2330.

Previous terms offered: Spring 2018.

CSCI 3325 (a) Distributed Systems
Every Other Year. Enrollment limit: 16.

Studies the key design principles and implementation challenges of distributed systems, which are collections of independent, networked machines functioning as single systems. Topics include networking and communication protocols, naming, synchronization, consistency and replication, fault tolerance, and security. Students gain exposure to real-world distributed systems through programming-intensive projects, as well as critiques of research papers covering a variety of real-world systems ranging from the Internet to file systems.

Prerequisites: CSCI 2330.

Previous terms offered: Spring 2019.

CSCI 3400 (a) Cognitive Architecture
Non-Standard Rotation. Enrollment limit: 16.

Advances in computer science, psychology, and neuroscience have shown that humans process information in ways that are very different from those used by computers. Explores the architecture and mechanisms that the human brain uses to process information. In many cases, these mechanisms are contrasted with their counterparts in traditional computer design. A central focus is to discern when the human cognitive architecture works well, when it performs poorly, and why. Conceptually oriented, drawing ideas from computer science, psychology, and neuroscience. No programming experience necessary. (Same as: DCS 3400)

Prerequisites: CSCI 2101 or BIOL 2135 or PSYC 2040 or PSYC 2740.

Previous terms offered: Spring 2019, Spring 2018.

CSCI 3415 (a) Robotics
Non-Standard Rotation. Enrollment limit: 16.

Robotics is a challenging discipline that encourages students to apply theoretical ideas from a number of different areas—artificial intelligence, cognitive science, operations research—in pursuit of an exciting, practical application: programming robots to do useful tasks. Two of the biggest challenges are building effective models of the world using inaccurate and limited sensors, and using such models for efficient robotic planning and control. Addresses these problems from both a theoretical perspective (computational complexity and algorithm development) and a practical perspective (systems and human/robot interaction) through multiple programming projects involving simulated and actual robots.

Prerequisites: CSCI 2330.

Previous terms offered: Fall 2016.

CSCI 3420 (a) Optimization and Uncertainty
Every Other Year. Enrollment limit: 16.

Optimization problems and the need to cope with uncertainty arise frequently in the real world. A numeric framework, rather than the symbolic one of traditional artificial intelligence, is useful for expressing such problems. In addition to providing a way of dealing with uncertainty, this approach sometimes permits performance guarantees for algorithms. Topics include constraint satisfaction, systematic and non-systematic search techniques, probabilistic inference and planning, and population-based optimization techniques (e.g., genetic algorithms and ant colony optimization). Formerly Computer Science 3425.

Prerequisites: CSCI 2101.

Previous terms offered: Spring 2020.

CSCI 3445 (a) Nature-Inspired Computation
Every Other Year. Enrollment limit: 16.

The size and complexity of real-world optimization problems can make it difficult to find optimal solutions in an acceptable amount of time. Researchers have turned to nature for inspiration in developing techniques that can find high-quality solutions in a reasonable amount of time; the resulting algorithms have been applied successfully to a wide range of optimization problems. Covers the most widely used algorithms, exploring their natural inspiration, their structure and effectiveness, and applications. Topics drawn from: genetic algorithms, particle swarm optimization, ant colony optimization, honeybee algorithms, immune system algorithms, and bacteria optimization algorithms. Requirements include labs, programming assignments, and a larger final project.

Prerequisites: CSCI 2101.

Previous terms offered: Fall 2019, Fall 2018, Spring 2017.

CSCI 3665 (a) From Data to Visualization: Designing Interactive Approaches to Understanding Information
Non-Standard Rotation. Enrollment limit: 16.

Information visualization is used to reveal patterns and outliers within abstract data, allowing powerful perceptual abilities to support slower and more deliberate cognitive abilities. Interactive visualizations can help investigate data and assist in the formation and exploration of hypotheses. Covers topics such as the transformation of data to visual representations, common approaches to dealing with different types of data, perceptual and cultural issues that influence how visualizations are interpreted, and the development of interactive visualization tools. Culminates in a significant final interactive visualization project.

Prerequisites: Two of: CSCI 2330 and CSCI 2200.

Previous terms offered: Fall 2017, Fall 2016.

CSCI 3715 (a, MCSR) Human-Computer Interaction
Every Year. Enrollment limit: 16.

How can we design technologies that result in positive and valuable (instead of ineffective and frustrating) experiences? Introduces key principles of user interface development by way of theory and hands-on practice. Topics include design principles (as informed by human perception and cognition) and prototyping techniques, as well as how to inspect and measure usability. Culminates in a final project, which is presented at the end of the term.

Prerequisites: CSCI 2330.

Previous terms offered: Spring 2020, Spring 2019, Spring 2018.

CSCI 3725 (a) Computational Creativity
Sarah Harmon.
Every Year. Fall 2020. Enrollment limit: 16.

Introduces theoretical foundations of modeling and evaluating creativity. Students learn techniques to assess creative systems and implement, analyze, and extend algorithms relevant to the latest state of the art. Special topics may include augmented creativity, hybrid systems, narrative intelligence, and algorithmic composition. Culminates in a final report that describes a novel creative technique or framework.

Prerequisites: CSCI 2101.

Previous terms offered: Fall 2019, Fall 2018.