COMPUTER SCIENCE

Overview & Learning Goals

Overview

The major in computer science focuses on critical thinking and problem-solving and is designed to blend traditional core areas (theory, systems, and artificial intelligence) with areas such as human-computer interaction, social and economic networks, machine learning, distributed systems, software engineering, computational geometry, and applied algorithms.

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 the opportunity to engage 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.

Options for Majoring or Minoring in the Department

Students may elect to major in computer science, the computer science and mathematics interdisciplinary major, or to coordinate the computer science major with digital and computational studies, education, or environmental studies. Students pursuing a coordinate or interdisciplinary major may not normally elect a second major. Non-majors may elect to minor in computer science.

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

Faculty

Laura I. Toma, *Department Chair*
Suzanne M. Theberge, *Senior Department Coordinator*

Professor: Laura I. Toma
Associate Professor: Sean K. Barker, Mohammad T. Irfan (Digital and Computational Studies)
Assistant Professors: David Byrd, Jeova Farias, Cibele Freire‡, Sarah M. Harmon
Lecturer: Kai Presler-Marshall
Visiting faculty: Ed Morehouse
Adjunct Lecturer: Stephen Houser (fall semester)

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 nine computer science courses and at least one mathematics course, as follows:

Code	Title	Credits
Required Courses		
Select one mathematics course numbered 1300 or higher.		1
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 two of which must be advanced-level courses numbered 3000 or higher.		6
Algorithms/Theory		
CSCI 2210	Theory of Computation	
CSCI 3220	Logic in Computer Science	
CSCI 3210	Computational Game Theory	
CSCI 3250	Computational Geometry	
CSCI 3225	Algorithms for GIS	
Artificial Intelligence		
CSCI 2400	Artificial Intelligence	
CSCI 3400	Cognitive Architecture	
CSCI 3420	Optimization and Uncertainty	
CSCI 3445	Nature-Inspired Computation	
CSCI 3465	Financial Machine Learning	
CSCI 3725	Computational Creativity	
CSCI 3485	Deep Learning for Computer Vision	
Systems		
CSCI 2320	Principles of Programming Languages	
CSCI 2335	Software Engineering	
CSCI 3310	Operating Systems	
CSCI 3325	Distributed Systems	
Projects Courses		
CSCI 3225	Algorithms for GIS	
CSCI 3325	Distributed Systems	
CSCI 3445	Nature-Inspired Computation	
CSCI 3465	Financial Machine Learning	
CSCI 3485	Deep Learning for Computer Vision	
CSCI 3725	Computational Creativity	

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://bowdoin-public.courseleaf.com/departments-programs/interdisciplinary-majors/).

Additional Information
Additional Information and Department Policies

- Students interested in majoring in computer science must complete CSCI 2101 Data Structures with a letter grade of C- or better by the end of their sophomore year. Prospective majors are also encouraged to satisfy their mathematics requirement as early as possible (ideally in their first two years).
- 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 CSCI 2101 Data Structures is a letter grade of C+ or better in CSCI 1101 Introduction to Computer Science or CSCI 1103 Accelerated Introduction to Computer Science. For all other CSCI 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.
- 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).
- 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.
- 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.

Information for Incoming Students (p. 2)

Students interested in computer science start with one of three courses, based on their computer science and mathematics placement test results: CSCI 1101 Introduction to Computer Science, CSCI 1103 Accelerated Introduction to Computer Science, or CSCI 2101 Data Structures. CSCI 1103 Accelerated Introduction to Computer Science covers the same materials as CSCI 1101 Introduction to Computer Science, but at an accelerated pace and without a lab section. Placements are binding and may not be overridden without permission of the department. Students with a placement of CSCI 1103 Accelerated Introduction to Computer Science should note that this course is normally offered only in the fall; CSCI 1101 Introduction to Computer Science and CSCI 2101 Data Structures are offered every semester. Students with a placement of MATH 1050 Quantitative Reasoning should complete MATH 1050 Quantitative Reasoning before taking a computer science course.

Introductory computer science classes regularly fill to capacity, and registration priority is given to first-years. Students are strongly encouraged to take their first computer science course during their first year, when they have priority registration. Students who intend to pursue a computer science major are required to complete CSCI 2101 Data Structures with a letter grade of C- or above by the end of their sophomore year.

Prior knowledge of computer science is used for placement only and does not count as credit towards the major. Students with any questions about appropriate placement should talk to a member of the department prior to registration.

Courses

CSCI 1101 (a, MCSR) Introduction to Computer Science
Kai Presler-Marshall; Jeova Farias; Stephen Houser; Ed Morehouse.
Every Semester. Fall 2023; Spring 2024. 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 experience with the concepts presented in class. Assumes no prior knowledge of computers or programming. Students with prior programming experience may instead be placed into CSCI 1103. Final grade must be C+ or better to serve as a prerequisite for Computer Science 2101.

Prerequisites: MATH 1050 or Placement in MATH 1600 (M) or Placement in MATH 1700 (M) or Placement in MATH 1750 (M) or Placement in MATH 1800 (M) or Placement in MATH 2020 or 2206 (M) or Placement in 2000, 2020, 2206 (M) or Placement in CSCI 1101.

Previous terms offered: Spring 2023, Fall 2022, Spring 2022, Fall 2021, Spring 2021, Fall 2020, Spring 2020, Fall 2019.

CSCI 1103 (a, MCSR) Accelerated Introduction to Computer Science
Stephen Houser.
Every Fall. Fall 2023. Enrollment limit: 30.

Covers the same introductory material as CSCI 1101, but at an accelerated pace and without an associated lab section. Appropriate for students with prior programming experience, though not necessarily in any specific programming language. Final course grade must be C+ or better to serve as a prerequisite for CSCI 2101.

Prerequisites: Two of: CSCI 1055 or either DCS 1100 or DCS 1200 or Placement in CSCI 1103 and MATH 1050 or Placement in MATH 1600 (M) or Placement in MATH 1700 (M) or Placement in MATH 1750 (M) or Placement in MATH 1800 (M) or Placement in MATH 2020 or 2206 (M) or Placement in 2000, 2020, 2206 (M).

Previous terms offered: Fall 2022, Fall 2021, Fall 2020, Fall 2019.

CSCI 2101 (a, MCSR) Data Structures
Sarah Harmon; David Byrd; Kai Presler-Marshall.
Every Semester. Fall 2023; Spring 2024. 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 either CSCI 1101 or CSCI 1103 with a grade of C+ or better.

Prerequisites: Two of: either CSCI 1101 or CSCI 1103 (same as DCS 1300) or Placement in CSCI 2101 and either CSCI 1101 or CSCI 1103 (same as DCS 1300) or Placement in CSCI 2101 and MATH 1600 or Placement in MATH 1600 (M) or Placement in MATH 1700 (M) or Placement in MATH 1750 (M) or Placement in MATH 1800 (M) or Placement in MATH 2020 or 2206 (M) or Placement in 2000, 2020, 2206 (M) and MATH 1600 or Placement in MATH 1600 (M) or Placement in MATH 1700 (M) or Placement in MATH 1750 (M) or Placement in MATH 1800 (M) or Placement in MATH 2020 or 2206 (M) or Placement in 2000, 2020, 2206 (M).

Previous terms offered: Spring 2023, Fall 2022, Spring 2022, Fall 2021, Spring 2021, Fall 2020, Spring 2020, Fall 2019.

CSCI 2200 (a, MCSR) Algorithms
Laura Toma.
Every Semester. Fall 2023; Spring 2024. 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 2023, Fall 2022, Spring 2022, Fall 2021, Spring 2021, Fall 2020, Spring 2020, Fall 2019.

CSCI 2210 (a, MCSR) Theory of Computation
Ed Morehouse.
Non-Standard Rotation. Fall 2023. 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: Fall 2022, Fall 2021, Fall 2020.

CSCI 2320 (a, MCSR) Principles of Programming Languages
Laura Toma; Mohammad Irfan.
Every Year. Fall 2023; Spring 2024. 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 2330.

Previous terms offered: Spring 2022, Spring 2021, Spring 2020.

CSCI 2330 (a, MCSR) Foundations of Computer Systems
Sean Barker.
Every Semester. Fall 2023; Spring 2024. 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 2023, Fall 2022, Spring 2022, Fall 2021, Spring 2021, Fall 2020, Spring 2020, Fall 2019.

CSCI 2335 (a) Software Engineering
Kai Presler-Marshall.
Every Fall. Spring 2024. Enrollment limit: 35.

A Java-based introduction to the processes of software engineering. How to design, implement, and test small-to medium-sized object-oriented systems, including the use of appropriate design notation to create a design before implementation. Closed-box and open-boxed testing, including writing and executing system test plans, writing unit tests, and using code coverage. An introduction to the best practices of software engineering, including pair programming, test-first development, static analysis, version control, and continuous integration. An introduction to project management. Class will feature several multi-week, hands-on projects; some projects will require students to work together in small teams (three to four members). Time permitting, we may also cover an introduction to full-stack web development, including HTML, JavaScript, and Databases.

Prerequisites: CSCI 2101.

Previous terms offered: Spring 2023, Fall 2022.

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 2023. 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: Spring 2022, Fall 2020, Fall 2019.

CSCI 2400 (a, MCSR) Artificial Intelligence
David Byrd.
Every Year. Fall 2023. 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 2022, Fall 2021, Fall 2020.

CSCI 2520 (a, MCSR) Dependently Typed Functional Programming
Ed Morehouse.
Non-Standard Rotation. Spring 2024. Enrollment limit: 30.

An important aspect of the expressiveness of a programming language concerns what kinds of things it can manipulate as data. This course introduces programming in a dependently typed functional language. In such a language both functions and types are ordinary data. This enables features such as higher-order and generic functions, as well as data whose type depends on the value of other data. One benefit of working in this setting is that properties of programs can be both expressed and proved within the language itself, providing much stronger guarantees of correctness than is possible using testing.

CSCI 2715 (a, MCSR) Human-Computer Interaction
Sarah Harmon.
Every Year. Spring 2024. Enrollment limit: 30.

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 2101.

Previous terms offered: Spring 2023.

CSCI 3210 (a) Computational Game Theory
Mohammad Irfan.
Every Year. Spring 2024. 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 2021, Fall 2020, Fall 2019.

CSCI 3225 (a) Algorithms for GIS
Laura Toma.
Every Year. Fall 2023. Enrollment limit: 16.

Studies key algorithms and data structures for geographical data. Topics include modeling processes on terrain such as drainage, watersheds, flooding, sea level rise and visibility, line and terrain simplification, spatial data structures such as B-trees, quad-trees and R-trees, space-filling curves, Delaunay triangulations and Voronoi diagrams. Students gain exposure to algorithmic modeling of spatial processes, transferring algorithms into practice and working with large spatial datasets through programming-intensive projects.

Prerequisites: Two of: CSCI 2330 and CSCI 2330 and CSCI 2200 and CSCI 2200.

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 2023, Fall 2021.

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: Spring 2021, Fall 2019.

CSCI 3310 (a) Operating Systems
Sean Barker.
Every Other Year. Spring 2024. 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: Fall 2022, Spring 2021.

CSCI 3325 (a) Distributed Systems
Sean Barker.
Every Other Year. Fall 2023. 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 2022.

CSCI 3400 (a) Cognitive Architecture
Eric Chown.
Non-Standard Rotation. Spring 2024. 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: Fall 2022.

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: Fall 2021, 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: Spring 2022, Spring 2021, Fall 2019.

CSCI 3465 (a, MCSR) Financial Machine Learning
David Byrd.
Every Spring. Spring 2024. Enrollment limit: 16.

Machine learning (ML) is the study of computer programs that can autonomously improve performance by exposure to additional data, and it is increasingly central to financial research and the finance industry. Both established firms and “FinTech” startups use ML to determine creditworthiness, allocate investment resources, and predict the future of company, sector, or national equities. These efforts raise interesting technical and ethical questions. Course content is divided between advanced technical topics like deep neural networks, deep reinforcement learning, and multi-agent simulation, and financial topics like the efficient markets hypothesis, capital assets pricing model, and mean-variance portfolio optimization. Emphasis is on empirical machine learning applied to financial topics, careful analysis of the results, and the need for responsible approaches to ML. Programming-heavy, project-heavy course. Strong CS background is needed. No prior finance knowledge is expected.

Prerequisites: CSCI 2400.

Previous terms offered: Spring 2023, Spring 2022.

CSCI 3485 (a, MCSR) Deep Learning for Computer Vision
Jeova Farias.
Non-Standard Rotation. Fall 2023. Enrollment limit: 16.

Computer vision has become ubiquitous in our society, from image searches to self-driving cars. On the other hand, deep learning has shaken the world of artificial intelligence in recent years. Most of these developments greatly advanced the performance of state-of-the-art visual recognition systems, which put computer vision at the epicenter of most technological progress from the past decade. In this context, this course aims at providing a consistent exploration of how deep learning began to its most recent achievements, always using computer vision tasks as their main application, historically or practically. During the course, we'll also understand many of the main computer vision problems and use them as cases for the introduction of various deep learning-related problems. Finally, this course hopes to give students working knowledge of one of the main deep learning frameworks, such as PyTorch, and prepare them for future industrial and academic careers in the field.

Prerequisites: CSCI 2101.

CSCI 3500 (a, MCSR) Software and Cognition
Laura Toma.
Non-Standard Rotation. Spring 2024. Enrollment limit: 20.

Does your brain treat writing a program more like writing a poem or more like solving an equation? What does the psychology of perception say about user interface design? Covers the intersection of software topics (e.g., reading, writing and debugging programs) and human factors (e.g., perception, bias, cognition). Will explore classic topics in psychology (e.g., Asch's conformity experiment, zero-risk bias, functional magnetic resonance imaging, mental rotation, Hick's law, etc.) in a manner that is approachable to students without prior psychology experience. Will relate these topics to a activities in computer science and discuss the resulting implications. Group discussions will focus on constructive conversations that summarize the work, analyze its strengths, and brainstorm ways in which it might be improved in the future.

CSCI 3725 (a) Computational Creativity
Sarah Harmon.
Every Year. Fall 2023. 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 2022, Spring 2021, Fall 2020, Fall 2019.