

COMPUTER SCIENCE

Departmental Guidelines

Computer Science is the study of algorithmic problem solving in both theoretical and applied areas. The major in Computer Science is designed to enable students to become well-rounded in these areas, and well-prepared for either graduate study or work in a variety of fields. Emphasis is placed on core concepts, analytical thinking, and problem solving throughout the curriculum.

In addition to a broad complement of introductory courses, the department regularly offers advanced courses in artificial intelligence and robotics, computer systems and networking, algorithm analysis and the theory of computation, software engineering, computer game design, and computational biology. Students have opportunities to conduct research through the Anderson Summer Research program and/or a senior research project.

Students interested in a Computer Science major should take:

Code	Title
An introductory course:	
CS 109	Discovering Computer Science
CS 111	Discovering Computer Science: Scientific Data and Dynamics
CS 112	Discovering Computer Science: Markets, Polls, and Social Networks
Followed by:	
CS 173	Intermediate Computer Science (by the end of the first year.)
In rare circumstances, a student may complete this sequence during the sophomore year.	
Majors should also take:	
CS 234 & MATH 135	Mathematical Foundations of Computer Science and Single Variable Calculus (during their first year.)

Faculty

Professor Ashwin Lall, Chair

Professors Mikey Goldweber, Matt Kretchmar, Ashwin Lall; Assistant Professors Michael Chavrimootoo, Flannery Currin, David Kahn, Matt Law, Stacey Truex; Visiting Instructor Ali Miller.

Academic Administrative Assistant

Dee Ghiloni

Systems Administrator

Tony Silveira

View faculty profiles and contact information (<https://denison.edu/academics/computer-science/contacts/>)

Computer Science Major

Computer Science offers two degrees, a minor, and a concentration. The two majors both require the computer science core curriculum. The core courses in Computer Science are:

Code	Title
An introductory course	
CS 109 or CS 111	Discovering Computer Science Discovering Computer Science: Scientific Data and Dynamics
or CS 112	Discovering Computer Science: Markets, Polls, and Social Networks
or CS 113 or CS 114	Discovering Computer Science: Physical Computing Discovering Computer Science: Computing for the Social Good
CS 173	Intermediate Computer Science
CS 181	Data Systems
CS 234	Mathematical Foundations of Computer Science
CS 271	Data Structures
CS 281	Introduction to Computer Systems
CS 371	Algorithm Design and Analysis
CS 395	Technical Communication I
MATH 135	Single Variable Calculus

Bachelor of Arts Degree

The minimum requirements for a Bachelor of Arts degree in Computer Science are the core courses plus two additional Computer Science courses at the 300 or 400 level (excluding 395/495, 361-362 and 363-364). One of the 300 or 400 level electives must be a Systems course and the other must either be a Theory or Applied elective.

Bachelor of Science Degree

The minimum requirements for a Bachelor of Science degree in Computer Science are the core courses, CS 372 - Operating Systems, and four additional Computer Science courses at the 300 or 400 level (excluding 395/495, 361-362, and 363-364). The electives must include at least one Theory elective and one Applied elective. Students may substitute MATH 220 - Applied Statistics for one of these additional courses. We recommend that Bachelor of Science candidates also take MATH 145 - Multivariable Calculus and MATH 300 - Introduction to Proofs, and one or more of MATH 213 - Linear Algebra and Differential Equations, MATH 220 - Applied Statistics and MATH 435 - Mathematical Modeling.

Students majoring in Mathematics or Computer Science may take up to two cross-listed mathematics or computer science courses to count as requirements in the intended major. These cross listed courses typically satisfy electives in the major. Courses pre-approved for this policy include Math/CS 334 Theory of Computation, Math 415/CS 337 Operations Research, and Math 427/CS 335 Probability, Computing, and Graph Theory. Math 220 (Applied Statistics) is also pre-approved as an elective for the computer science BS major. A year-long senior research project may count as one elective toward the major. Other math and computer science courses must be approved by the department prior to enrollment.

Electives

The three categories of Computer Science electives are Theory, Systems, and Applied courses.

- Theory electives are proof-based and feature a high level of mathematical content.
- Systems electives are concerned with the principles and practice of providing infrastructure that enables other software to be executed.
- Applied electives demonstrate applications of computer science to a variety of other disciplines.

Code	Title
The Applied courses are	
CS 314	Game Design
CS 337	Operations Research
CS 339	Artificial Intelligence
CS 349	Software Engineering
CS 391	Robotics
The Systems courses are	
CS 372	Operating Systems
CS 373	Programming Languages
CS 374	Compilers
CS 375	Computer Networks
CS 377	Database Systems
The Theory courses are	
CS 333	Big Data Algorithms
CS 334	Theory of Computation
CS 335	Probability, Computing and Graph Theory

Computer Science Minor

A minor in computer science consists of:

Code	Title
An introductory course	
CS 109 or CS 111 or CS 112 or CS 113 or CS 114	Discovering Computer Science Discovering Computer Science: Scientific Data and Dynamics Discovering Computer Science: Markets, Polls, and Social Networks Discovering Computer Science: Physical Computing Discovering Computer Science: Computing for the Social Good
CS 173	Intermediate Computer Science
CS 181	Data Systems
CS 234	Mathematical Foundations of Computer Science
CS 271	Data Structures
MATH 135	Single Variable Calculus

Computational Science Concentration

Computational Science is the field of study concerned with constructing mathematical models and numerical solution techniques, and using computer algorithms and simulation to analyze and solve scientific, social scientific, and engineering problems.

The Computational Science concentration consists of four core courses:

Code	Title
MATH 145	Multivariable Calculus
MATH 213	Linear Algebra and Differential Equations
CS 173 or CS 181	Intermediate Computer Science Data Systems
One of	
CS 109	Discovering Computer Science
CS 111	Discovering Computer Science: Scientific Data and Dynamics
CS 112	Discovering Computer Science: Markets, Polls, and Social Networks
CS 113	Discovering Computer Science: Physical Computing
CS 114	Discovering Computer Science: Computing for the Social Good

and an additional course which may be in another department, that must have a strong and persistent mathematical modeling or computing component and must be approved in advance by the Department of Mathematics or the Department of Computer Science.

In addition, students must take a two-semester sequence of courses in a department other than Mathematics and Computer Science. A written plan for completing the concentration must be approved by the Department of Mathematics or the Department of Computer Science before the end of the student's junior year of study and prior to enrollment in the elective courses. In particular, the elective courses and cognate requirements specified above must be chosen consistently with a valid educational plan for the study of Computational Science (as defined above). Any mathematics major who wishes to complete this concentration must choose non-math courses as their elective courses. Any computer science major who wishes to complete this concentration must choose non-computer science courses for their elective courses. A double mathematics and computer science major is not eligible for this concentration.

Additional Points of Interest

Students who intend to continue with graduate study in Computer Science should pursue the Bachelor of Science degree.

Computer Science students should be adept at not only solving problems through the implementation of computer programs, but also in communicating those solutions to a wide variety of audiences. Students should learn the proper use of documentation to share their programs with users and other computer scientists, be prepared to compose proofs and analyses of their algorithms, and have opportunities to write formal papers.

Research at Denison

Denison offers a number of research opportunities, including funding for summer research projects. The Anderson Foundation and the Denison University Research Foundation (DURF) support qualified students conducting summer research. For off-campus research opportunities in Computer Science, see various National Science Foundation Research Experience for Undergraduates (https://www.nsf.gov/crssprgm/reu/list_result.jsp?unitid=5049) experiences. Interested students should consult a faculty member as early as possible in the fall semester.

Off-Campus Study

The Department of Computer Science strongly encourages students to globalize their education by completing some portion of their undergraduate education abroad. A majority of Denison students spend a semester abroad during their junior year and many more spend a summer (or two) abroad. Denison offers a wide range of opportunities to study off-campus that are highly relevant to your Denison experience.

Going abroad allows students to enhance their knowledge while experiencing another culture and way of life. Students gain valuable experience that will benefit future career goals and/or graduate school opportunities. Computer Science majors who are fluent in another language will have special advantages in the job market.

Students may take up to two classes outside the department to transfer towards the major at Denison. Additional courses taken outside Denison may accrue credit hours towards graduation, but will not contribute to requirements in the major. Courses taken outside the university must be **pre-approved** for acceptance towards major requirements. Students should provide the department chair syllabi for the intended courses for department approval. Students may petition the department chair for exceptions to this policy. In particular, transfer students may be eligible to transfer additional courses towards major requirements.

Courses

CS 109 - Discovering Computer Science (4 Credit Hours)

This course is an introduction to computational problem solving. In each instance of the course, students will develop their abilities to abstract and model problems drawn from a particular application domain and generate elegant and efficient solutions. Students will practice these skills by developing computer programs to solve these applied problems. The course will cover programming fundamentals, as well as the development of algorithms and data manipulation techniques related to the chosen application area. Students may earn credit for at most one of CS 109, CS 111, CS 112, CS 113, and CS 114. Absolutely no prior experience is necessary.

CS 111 - Discovering Computer Science: Scientific Data and Dynamics (4 Credit Hours)

This course is an introduction to computational problem solving. Students will develop their abilities to abstract (or model) otherwise complex problems and generate elegant and efficient solutions. Students will practice these skills by developing computer programs that solve problems motivated by research in the sciences. Additional topics may include Monte Carlo methods, data analysis, population dynamics, computational biology, genetic algorithms, cellular automata, networks, data mining, and fractals. Students may earn credit for at most one of CS 109, CS 111, CS 112, CS 113, and CS 114. Absolutely no prior experience is necessary.

CS 112 - Discovering Computer Science: Markets, Polls, and Social Networks (4 Credit Hours)

This course is an introduction to computational problem-solving. Students will develop their abilities to abstract otherwise complex problems and generate elegant and efficient solutions. Students will practice these skills by developing computer programs that solve problems that arise in the social sciences. Topics such as social networks, population modeling in economics, data collection via polling, voting systems, game theory, and Congressional polarization will be discussed in the context of computational problem-solving. Students may earn credit for at most one of the following courses: CS 109, CS 111, CS 112, CS 113, or CS 114. Absolutely no prior experience is necessary.

CS 113 - Discovering Computer Science: Physical Computing (4 Credit Hours)

This course is an introduction to computational problem solving. Students will develop their abilities to abstract otherwise complex problems and generate elegant and efficient solutions. Students will practice these skills by developing computer programs that solve problems related to physical devices that interact with the world, e.g., through sound, light, or motion. Topics may include working with microcontrollers, sensors, and actuators to process and/or transmit information. Students may earn credit for at most one of CS 109, CS 111, CS 112, CS 113, and CS 114. Absolutely no prior experience is necessary.

CS 114 - Discovering Computer Science: Computing for the Social Good (4 Credit Hours)

This course is an introduction to computational problem solving. Students will develop their abilities to abstract otherwise complex problems and generate elegant and efficient solutions. Students will practice these skills by developing computer programs that address issues of social relevance. Topics such as pollution, aquifer depletion, climate change, police use of deadly force, solar energy deployment, and carbon sequestration will be discussed in the context of computational problem-solving. Students may earn credit for at most one of CS 109, CS 111, CS 112, CS 113, and CS 114. Absolutely no prior experience is necessary.

CS 119 - Seminar: Programming Problems (1 Credit Hour)

Students meet weekly to solve a challenging programming problem. Strategies for solving problems will be discussed. Used as a preparation for programming contests.

Prerequisite(s): CS 173.

CS 173 - Intermediate Computer Science (4 Credit Hours)

A study of intermediate level computer science principles and programming techniques with an emphasis on abstract data types and software engineering. Topics include recursion, sorting, dynamic memory allocation, basic data structures, software engineering principles, and modularization.

Prerequisite(s): CS 109 or CS 111 or CS 112 or CS 113 or CS 114.

CS 181 - Data Systems (4 Credit Hours)

This course provides a broad perspective on the access, structure, storage, and representation of data. It encompasses traditional database systems, but extends to other structured and unstructured repositories of data and their access/acquisition in a client-server model of Internet computing. Also developed are an understanding of data representations amenable to structured analysis, and the algorithms and techniques for transforming and restructuring data to allow such analysis.

Prerequisite(s): CS 109 or CS 111 or CS 112 or CS 113 or CS 114.

Crosslisting: DA 210.

CS 199 - Introductory Topics in Computer Science (1-4 Credit Hours)

A general category used only in the evaluation of transfer credit.

CS 200 - Topics in Computer Science (1 Credit Hour)

This is a mini-seminar devoted to a particular application or programming language. Topics have included: relational database and SQL, software engineering, Advanced C++, cryptography, and parallel programming.

Prerequisite(s): CS 173.

CS 234 - Mathematical Foundations of Computer Science (4 Credit Hours)

The goal of this course is to provide an understanding of the mathematical techniques that underlie the discipline of computer science. In this course, students learn mathematical proof techniques, such as induction and proof by contradiction, and how to write rigorous proofs. It also serves as an introduction to the fundamentals of the theory of computation. Models of computation, namely finite automata and Turing machines, are studied with the goal of understanding what tasks computers are and are not capable of performing.

Prerequisite(s): MATH 130 or MATH 135 or MATH 145, and CS 109 or CS 111 or CS 112 or CS 113 or CS 114.

CS 271 - Data Structures (4 Credit Hours)

In this course, students study a variety of data organization methods, and implement and analyze the efficiency of basic algorithms that use these data structures. Course topics include lists, stacks, queues, binary search trees, heaps, priority queues, hash tables, and balanced trees. Students will also be introduced to basic functional programming in LISP. The department strongly recommends that students enrolling in this course have earned a grade of C or higher in Intermediate Computer Science (CS 173) and a grade of C or higher in Mathematical Foundations of Computer Science (CS 234).

Prerequisite(s): CS 173 and MATH 135 or MATH 145 and CS 234 or MATH 300.

CS 281 - Introduction to Computer Systems (4 Credit Hours)

The Introduction to Computer Systems course provides a perspective into how computer systems execute programs, store information, and communicate. It enables students to become better problem solvers, especially in dealing with issues of performance, portability and robustness. It also serves as a foundation for courses on operating systems, networks, and parallel computing, where a deeper understanding of systems-level issues is required. Topics covered include: basic digital logic design and computer organization, machine-level code and its generation by compilers, performance evaluation and optimization, representation and computer arithmetic, and memory organization and management.

Prerequisite(s): CS 173.

CS 299 - Intermediate Topics in Computer Science (1-4 Credit Hours)

A general category used only in the evaluation of transfer credit.

CS 314 - Game Design (4 Credit Hours)

This course is about the computer science and theory of game design as well as practical game development. It covers computer science concepts such as 3D projection and transformation, rasterization, texture-mapping, shading, path-finding, and game theory, as well as game design topics such as mechanics, elements, theme, iteration, balance, documentation, and interest curves. A significant component of the course consists of prototyping computer games. The department strongly recommends that students enrolling in this course have earned a grade of C or higher in Data Structures (CS 271). This course is classified as an applied elective.

Prerequisite(s): CS 271.

CS 323 - Data Privacy (4 Credit Hours)

The explosion of data collection and advances in artificial intelligence and machine learning have motivated a robust economy around data-based services. While such services provide opportunities for a broad array of individuals and companies to leverage the power of modern data analytics and machine learning, this new economy also exposes new vulnerabilities and privacy risks. This course will explore the growing area of data privacy in modern computing systems including formal frameworks such as differential privacy and secure multiparty computation. Students will work to understand techniques, issues, and trade-offs related to data privacy in a computing context. In particular, students will study: definitions of data privacy, techniques for achieving privacy, limitations and trade-offs inherent in various settings, and the relationship between privacy policy and privacy technology. The department strongly recommends that students enrolling in this course have earned a grade of C or higher in Data Structures (CS 271). This course is classified as a theory elective.

Prerequisite(s): CS 271.

CS 333 - Big Data Algorithms (4 Credit Hours)

This course is about the design and analysis of big data algorithms, i.e. algorithms that compute on extremely large datasets. Two frameworks are required to understand big data algorithms: MapReduce algorithms for data stored on a cluster, and streaming algorithms for data too large to store. After introducing these frameworks, the course covers numerous examples of big data algorithms, including hashing, frequency moments, Google's PageRank algorithm, matching algorithms, clustering, the Netflix recommendation algorithm, algorithms on social network graphs, and dimensionality reduction. The analysis of such algorithms requires tools from probability theory and statistics, which will be introduced as needed.

Prerequisite(s): CS 181 or DA 210 and CS 271.

CS 334 - Theory of Computation (4 Credit Hours)

This course will continue from where CS 234 left off in studying computers as mathematical abstractions to understand the limits of computation. In this course, students will learn about topics in computability theory and complexity theory. Topics in computability theory include Turing machines and their variations, the Universal Turing machine, decidability of the halting problem, reductions, and proving the decidability of other problems. Topics in complexity theory include the classes P and NP, NP-completeness, and other fundamental complexity classes. The Department strongly recommends that students enrolling in this course have earned a grade of C or higher in Data Structures (CS 271). This course is classified as a theory elective.

Prerequisite(s): CS 109, CS 111, CS 112, CS 113 or CS 114, and MATH 300 or CS 234 and CS 271.

Crosslisting: MATH 334.

CS 335 - Probability, Computing and Graph Theory (4 Credit Hours)

This course is about the design and analysis of randomized algorithms (i.e. algorithms that compute probabilistically). Such algorithms are often robust and fast, though there is a small probability that they return the wrong answer. Examples include Google's PageRank algorithm, load balancing in computer networks, coping with Big Data via random sampling, navigation of unknown terrains by autonomous mobile entities, and matching medical students to residencies. The analysis of such algorithms requires tools from probability theory, which will be introduced as needed. This course also covers the basics of graph theory, and several randomized algorithms on graphs. Graphs are often used to mathematically model phenomena of interest to computer scientists, including the internet, social networks graphs, and computer networks. Lastly, this course demonstrates the powerful Probabilistic Method to non-constructively prove the existence of certain prescribed graph structures, how to turn such proofs into randomized algorithms, and how to derandomize such algorithms into deterministic algorithms. The department strongly recommends that students enrolling in this course have earned a grade of C or higher in Data Structures (CS 271). This course is classified as a theory elective.

Prerequisite(s): CS 271 or MATH 220 or DA 220, and MATH 300, and one from CS 109 or CS 111 or CS 112 or CS 113 or CS 114.

Crosslisting: MATH 427.

CS 337 - Operations Research (4 Credit Hours)

This course involves mathematical modeling of real-world problems and the development of approaches to find optimal (or nearly optimal) solutions to these problems. Topics may include: modeling, linear programming and the simplex method, the Karush-Kuhn Tucker conditions for optimality, duality, network optimization, and nonlinear programming.

Prerequisite(s): CS 109, CS 111, CS 112, CS 113 or CS 114, and MATH 213, and MATH 220 or DA 220.

Crosslisting: MATH 415.

CS 339 - Artificial Intelligence (4 Credit Hours)

A survey course of topics in Artificial Intelligence including search, formal systems, learning, connectionism, evolutionary computation and computability. A major emphasis is given to the philosophy of Artificial Intelligence. The department strongly recommends that students enrolling in this course have earned a grade of C or higher in Data Structures (CS 271). This course is classified as an applied elective.

Prerequisite(s): CS 271 or MATH 213 or consent.

CS 349 - Software Engineering (4 Credit Hours)

Students will apply their theoretic background, together with current research ideas to solve real problems. They will study principles of requirements analysis, methods of designing solutions to problems, and testing techniques, with special emphasis on documentation. The department strongly recommends that students enrolling in this course have earned a grade of C or higher in Data Structures (CS 271). This course is classified as an applied elective.

Prerequisite(s): CS 271 and CS 281.

CS 361 - Directed Study (1-4 Credit Hours)

A student in good standing may work intensively in areas of special interest under the Directed Study plan. A Directed Study is appropriate when, under the guidance of a faculty member, a student wants to explore a subject more fully than is possible in a regular course or to study a subject not covered in the regular curriculum. A Directed Study should not normally duplicate a course that is regularly offered. Directed Studies are normally taken for 3 or 4 credits. A one-semester Directed Study is limited to a maximum of 4 credit hours. Note: Directed Studies may not be used to fulfill General Education requirements.

CS 362 - Directed Study (1-4 Credit Hours)

A student in good standing may work intensively in areas of special interest under the Directed Study plan. A Directed Study is appropriate when, under the guidance of a faculty member, a student wants to explore a subject more fully than is possible in a regular course or to study a subject not covered in the regular curriculum. A Directed Study should not normally duplicate a course that is regularly offered. Directed Studies are normally taken for 3 or 4 credits. A one-semester Directed Study is limited to a maximum of 4 credit hours. Note: Directed Studies may not be used to fulfill General Education requirements.

CS 363 - Independent Study (1-4 Credit Hours)

Independent Study engages a student in the pursuit of clearly defined goals. In this effort a student may employ skills and information developed in previous course experiences or may develop some mastery of new knowledge or skills. A proposal for an Independent Study project must be approved in advance by the faculty member who agrees to serve as the project advisor. Note: Independent Studies may not be used to fulfill General Education requirements.

CS 364 - Independent Study (1-4 Credit Hours)

Independent Study engages a student in the pursuit of clearly defined goals. In this effort a student may employ skills and information developed in previous course experiences or may develop some mastery of new knowledge or skills. A proposal for an Independent Study project must be approved in advance by the faculty member who agrees to serve as the project advisor. Note: Independent Studies may not be used to fulfill General Education requirements.

CS 371 - Algorithm Design and Analysis (4 Credit Hours)

In this course, students study in depth the design, analysis, and implementation of efficient algorithms to solve a variety of fundamental problems. The limits of tractable computation and techniques that can be used to deal with intractability are also covered. The department strongly recommends that students enrolling in this course have earned a grade of C or higher in Data Structures (CS 271).

Prerequisite(s): CS 234, CS 271, and junior/senior status.

CS 372 - Operating Systems (4 Credit Hours)

A study of the principles of operating systems and the conceptual view of an operating system as a collection of concurrent processes. Topics include process synchronization and scheduling, resource management, memory management and virtual memory, and file systems. The department strongly recommends that students enrolling in this course have earned a grade of C or higher in Data Structures (CS 271). This course is classified as a systems elective.

Prerequisite(s): CS 181 or DA 210, CS 271, and CS 281.

CS 373 - Programming Languages (4 Credit Hours)

A systematic examination of programming language features independent of a particular language. Topics include syntax, semantics, typing, scope, parameter modes, blocking, encapsulation, translation issues, control, inheritance, language design. A variety of languages from different classes are introduced. The department strongly recommends that students enrolling in this course have earned a grade of C or higher in Data Structures (CS 271). This course is classified as a systems elective.

Prerequisite(s): CS 181 or DA 210, CS 271, and CS 281.

CS 374 - Compilers (4 Credit Hours)

A study of regular and context-free languages with the purpose of developing theory to build scanners and parsers. The class will develop its own structured language and construct a working compiler. An examination of compiler construction tools. The department strongly recommends that students enrolling in this course have earned a grade of C or higher in Data Structures (CS 271). This course is classified as a systems elective.

Prerequisite(s): CS 181 or DA 210, CS 271, CS 281, and CS 334.

CS 375 - Computer Networks (4 Credit Hours)

A study of computer network architecture and protocols. Topics include packet and circuit switching, datalink, network and transport layer protocols, reliability, routing, internetworking, and congestion control. The department strongly recommends that students enrolling in this course have earned a grade of C or higher in Data Structures (CS 271). This course is classified as a systems elective.

Prerequisite(s): CS 181 or DA 210, CS 271, and CS 281.

CS 377 - Database Systems (4 Credit Hours)

A study of the design, implementation and application of database management systems. Topics include the relational data model, physical implementation issues, database design and normalization, query processing and concurrency. The department strongly recommends that students enrolling in this course have earned a grade of C or higher in Data Structures (CS 271). This course is classified as a systems elective.

Prerequisite(s): CS 181 or DA 210, CS 271, and CS 281.

CS 391 - Robotics (4 Credit Hours)

An introductory course in both hardware and software aspects of robotics. Students will learn the basics of manipulators, sensors, locomotion, and micro-controllers. Students will also construct a small mobile robot and then program the robot to perform various tasks. The department strongly recommends that students enrolling in this course have earned a grade of C or higher in Data Structures (CS 271). This course is classified as an applied elective.

Prerequisite(s): CS 271 and CS 281.

CS 395 - Technical Communication I (1 Credit Hour)

This course aims to enhance mathematics and computer science students' proficiency and comfort in orally communicating content in their disciplines. Students will develop skills in presenting technical information to a non-technical audience. In particular, students will deliver several presentations during the semester on substantive, well-researched themes appropriate to their status in their major.

Prerequisite(s): MATH 300 or CS 271.

Corequisite(s): a 300-level or higher mathematics or computer science course.

Crosslisting: MATH 395.

CS 399 - Advanced Topics in Computer Science (1-4 Credit Hours)

A general category used only in the evaluation of transfer credit.

CS 401 - Algorithms, Computations and Elections (4 Credit Hours)

Topics may include Computer Graphics, Neural Networks, Advanced Algorithms, Network Security, or other subjects of current interest. Fall 2025 Description: This course introduces students to the computational aspects of collective decision-making, i.e., computational social choice. Students will learn to study and design algorithms or prove the lack thereof, to tackle problems in tournaments, fairness in resource allocation, strategic/manipulative voting, electoral control (e.g., gerrymandering and voter suppression), bribery, and coalition formation, among other topics. Students may elect to take this course as a Theory or Applied elective.

Prerequisite(s): CS 271 or equivalent with instructor permission.

CS 402 - Advanced Topics in Computer Science (4 Credit Hours)

Topics may include Computer Graphics, Neural Networks, Advanced Algorithms, Network Security or other subjects of current interest.

CS 451 - Senior Research (4 Credit Hours)

Students may enroll in Senior Research in their final year at Denison.

Normally, Senior Research requires a major thesis, report, or project in the student's field of concentration and carries eight semester-hours of credit for the year. Typically, a final grade for a year-long Senior Research will not be assigned until the completion of the year-long Senior Research at the end of the second semester. In which case, the first semester Senior Research grade will remain "in progress" (PR) until the completion of the second semester Senior Research. Each semester of Senior Research is limited to a maximum of 4 credit hours. Note: Senior Research may not be used to fulfill General Education requirements.

CS 452 - Senior Research (4 Credit Hours)

Students may enroll in Senior Research in their final year at Denison.

Normally, Senior Research requires a major thesis, report, or project in the student's field of concentration and carries eight semester-hours of credit for the year. Typically, a final grade for a year-long Senior Research will not be assigned until the completion of the year-long Senior Research at the end of the second semester. In which case, the first semester Senior Research grade will remain "in progress" (PR) until the completion of the second semester Senior Research. Each semester of Senior Research is limited to a maximum of 4 credit hours. Note: Senior Research may not be used to fulfill General Education requirements.

CS 495 - Technical Communication II (1 Credit Hour)

This course is a capstone experience in oral and written communication for mathematics and computer science majors. Students will research a substantive topic, write a rigorous expository article, and make a presentation to the department.

Prerequisite(s): MATH/CS 395 and a 300-400 level computer science course or a 400-level mathematical course.