

Computer Science Make a Gift

[Home](#) >

Program Description

The computer science curriculum, offered by the Department of Computer Science, enables a student to study systematic approaches to problem solving within a computing environment. In the course of this study, the student develops the practice of clear thinking and logical reasoning, while learning to analyze information processing tools and systems. Within this study there is an emphasis on the human factors associated with computing in the modern world.

The available courses cover a wide area of computer science. In addition, upper-division students may, in collaboration with faculty members, be involved in on-going research projects or study topics not covered by the regular course offerings.

The major is designed to give students a broad understanding of both the theoretical and applied aspects of the discipline. As such, it provides a firm foundation for those intending to do graduate work as well as for those interested in pursuing a career in computer science.

- [Program Requirements](#)
- [Course Listing](#)
- [Computer Science website](#)

Program Requirements

Requirements for the Computer Science major

The requirements for a Bachelor of Arts in computer science are 10 courses in computer science at the level of Computer Science 107/111 or above. One of the courses may be selected from a list of approved courses offered by other departments.

The 10 courses must include:

- Computer Science 107/111: Computer Science I
- Computer Science 112: Computer Science II
- Computer Science 201: The Mathematics of Computation OR Math 215: Abstract Mathematics I
- Computer Science 216: Data Structures
- Computer Science 221: Computer Organization and Assembly Language Programming
- Computer Science 301: Theory of Computation
- Computer Science 440: Advanced System Design (taken during the senior year)

In addition to the 10 courses in computer science, students must take:

- Equivalent of Mathematics 111 or above (usually taken during the first year)

Approved courses offered by other departments

- Biology 251
- Mathematics 351 and 366
- Physics 240
- Psychology 315, 316 or 338

Requirements for the Computer Science minor

The minor in Computer Science consists of any six Computer Sciences courses numbered 107/111 or above.

Course Listing

Course level:

100 | 200 | 300 | 400

CS-103 Introduction to Computing

Liberal arts introduction to the discipline of computer science and the use of computers in a variety of fields. Topics include a historical survey of technology and the use of computers, computer application, software systems design, programming with scripts, computer hardware and digital logic design, and implications of computing. Course is laboratory-oriented and includes several hands-on laboratory projects. Prerequisite: none.

CS-107 Introduction to Scientific Computation

Introduction to computer science with a scientific computation language (e.g., MATLAB). Emphasis is on fundamental programming constructs, algorithmic thinking and problem-solving, fundamental data structures, and recursion. Student projects include common scientific computing challenges and data visualization. Tailored for students interested in mathematics or the sciences. May be taken as an alternative to Computer Science 111, the usual beginning course for a major or minor in computer science. Prerequisite: none.

CS-111 Computer Science I

Introduction to computer science with an emphasis on fundamental programming constructs, algorithmic thinking and problem-solving, fundamental data structures, and recursion. Students implement projects using the Java programming. This course is the usual beginning course for students planning to pursue a major or minor in computer science. Prerequisite: none.

CS-112 Computer Science II

Second course in the introductory sequence for computer science majors and students interested in the principles of programming. Special attention is given to object-oriented program design methods, algorithms, and elementary data structures. Prerequisite: Computer Science 107 or 111.

[Back to the top of course listing](#)

CS-201 The Mathematics of Computation

Study of the mathematics needed for an understanding of the theoretical foundations of computation. Topics include mathematical logic, set theory, mathematical induction, mathematical definitions and proofs, graph theory, and an introduction to finite-state automata. Applications and illustrative examples are drawn from computer science topics such as digital circuits, analysis and correctness of algorithms, automata, decidable problems, and efficient searching. Prerequisites: Computer Science 107/111 and Mathematics 111.

CS-216 Data Structures

Introduction to major data structures and some of their applications. Topics include sets, queues, stacks, linked lists, string processing, trees, graphs, arrays, tables, files, and an introduction to the analysis of algorithms. Prerequisite: Computer Science 112.

CS-221 Computer Organization and Assembly Language Programming

Programming at the machine level, with emphasis on the logical connection of the basic components of the computer and systems programs. Topics include machine and assembly language programming, basic computer operations, data representation, hardware organization, systems software, and compilers. Prerequisite: Computer Science 112.

CS-251 Introduction to Bioinformatics

Introduction to the emerging field of bioinformatics, where biology and computer science intersect to interpret and use the rapidly expanding volume of biological information produced by genome sequencing and proteomic exploration of life processes. Application of bioinformatic software tools to the analysis of gene sequences and protein structures is emphasized. Students undertake a laboratory project combining *in silico* and *in vitro* approaches to isolate and then analyze a segment of their own DNA. An introduction to computer algorithms used in bioinformatic software is provided. Three class hours and laboratory. Biology 251 and CS 251 are cross-listed.

[Back to the top of course listing](#)

CS-301 The Theory of Computation

Study of the basic theoretical principles of the computational model. Topics include finite automata, regular expressions, context-free grammars, Turing Machines, Church's Thesis, P and NP classes, the halting problem, unsolvability, computational complexity, and program verification. Prerequisite: Computer Science 201.

CS-311 Design and Analysis of Algorithms

Survey of basic principles and techniques for the development of good algorithms. Emphasis is placed on individual development of algorithms and an analysis of the results in terms of usefulness, efficiency, and organization. Topics include design techniques, worst case and average case analysis, searching, sorting, branch and bound, spanning trees, reachability, combinatorial methods, and NP-hard problems. Prerequisites: Computer Science 201, Computer Science 216.

CS-322 Introduction to Computer Networks

Introduction to principles used to analyze and build a network of computers. Course covers concepts and issues relating to low-level communications and protocols of computer networking. Students study formal methods for integrating communication events into normal process cycles of the computer, then concentrate on a study of practices for defining and specifying a formal communications protocol. Throughout the course, students apply principles that they study to existing networks within the department. Prerequisite: Computer Science 216. Offered alternate years.

CS-324 Principles of Operating Systems

Study of fundamental concepts of operating systems. Topics include sequential processes, concurrent processes, processor management, memory management, scheduling algorithms, and computer security. Projects include writing of a program to simulate major components of an operating system. Prerequisite: Computer Science 216. Offered alternate years.

CS-327 Parallel and Distributed Processing

Introduction to techniques used to implement multiple processor problem-solving environments. Investigation includes several different environments for parallel computing, including SIMD (Single Instruction Multiple Data stream), MIMD (Multiple Instructions Multiple Data stream), and computing in a distributed workstation environment. Students work with actual implementations of each of these environments and explore their advantages and design algorithms appropriate for these environments. Prerequisite: Computer Science 216. Offered alternate years.

CS-335 Software Engineering

Introduction to principles used to analyze and specify software systems. Concepts and issues relating to initial stages of the software life cycle are covered. Study examines formal methods for analyzing and investigating environments requiring automation, then studies languages and CASE (Computer-Aided Software Engineering) tools. Throughout the course students apply the principles that they study to situations on campus and within local communities. Prerequisite: Computer Science 216.

CS-341 Survey of Programming Languages

Study of fundamental concepts in the design of programming languages. Concepts include BNF grammar specification, abstract syntax trees, variables, expressions typing, scope, lexical address, procedures, data types, static/dynamic binding, and environment-passing interpreters. Special emphasis is placed on the development of an interpreter using a functional programming language. Other languages are introduced to further illustrate these concepts. Prerequisite: Computer Science 216. Offered alternate years.

CS-360 Principles of Database Systems

Study of fundamental concepts of database systems. Topics include physical organization of databases, indexing techniques, and query processing. Particular models studied include the Entity-Relationship and Relational. Class projects stress design and implementation of a database. Prerequisite: Computer Science 216. Offered alternate years.

CS-371 Introduction to Artificial Intelligence

Study of modern techniques for creating software that behaves intelligently. Topics include uninformed and heuristic search, constraint satisfaction, stochastic optimization, game-tree search, propositional reasoning, probabilistic reasoning, Bayesian networks with Markov chain Monte Carlo techniques, and robotics. Course concludes with a class robotics project. Prerequisite: Computer Science 216. Offered alternate years.

CS-373 Computer Graphics

Study of methods and issues surrounding the construction of graphical images on the computer. Topics include windowing systems and user input, two-dimensional graphics packages, curve drawing techniques, modeling in three dimensions, use of lighting and shading techniques, and the process of rendering images. Student work consists both of using existing packages to create images and of implementing algorithms used in graphical systems. Prerequisite: Computer Science 216. Offered alternate years.

CS-374 Compilers

Introduction to techniques used to translate high-level computer languages into machine code. Course covers current implementation techniques and relevant theory. Topics include lexical scanning, parsing, abstract syntax trees, semantic analysis, intermediate code generation, and code generation. Students complete a major project involving the compilation of a particular computer language. Prerequisite: Computer Science 216. Offered alternate years.

CS-391 Selected Topics

CS-392 Selected Topics

[Back to the top of course listing](#)

CS-440 Advanced Systems Design

Senior capstone course teaching a formal approach to techniques of computer project development. Integral part of course is the involvement of students, working as a team, in the development of a large project. Topics include formal requirement and specification, software testing techniques, written and spoken communication skills in technology, and user interfaces. Research option is available by permission of the instructor with a faculty research sponsor and recommendation from a previous research supervisor. Prerequisites: Computer Science 216, one 300-level computer science course, and senior status or permission of department.

CS-450 Individualized Study-Tutorial

Individualized tutorial counting toward the minimum requirements in a major or minor, graded A-F

CS-460 Individualized Study-Research

Intensive study of a selected topic in computer science or a related area. Research project is completed in collaboration with a faculty member. Prerequisites: Computer Science 216 and permission of department.

CS-477 Half Credit Internship

Half credit internship, graded S/U.

[Back to the top of course listing](#)

[Students](#)

[Faculty](#)

[Employees](#)

[Alumni](#)

[Families](#)

[Visitors](#)

[Employment](#)

[Directory](#)

[News](#)

[Calendars & Events](#)

[myGettysburg](#)

[Giving](#)

[About this Site](#)

[Email via the Web](#)

[Moodle](#)

[CNAV](#)

[Student Center](#)

[Library](#)

[Bookstore](#)

[Mobile](#)



Gettysburg College
300 North Washington Street
Gettysburg, Pennsylvania 17325
717.337.6300

[Back to top](#)