COMPUTER SCIENCE (CSC)

A minimum GPA of 2.0 in the computer science courses that comprise a major or minor in the department is required for graduation.

Students who are enrolled at Wake Forest University may not take courses in computer science at other institutions to satisfy divisional requirements.

Students with a special interest in multidisciplinary work should consider a program of study that combines computer science with another discipline through either a double major or a minor.

Planning for a Major or Minor in Computer Science

Students do **not** need prior computer science experience to major in computer science. While not required, students interested in the major are encouraged to:

- take CSC 111 and MTH 117 in their freshman year. Students with AP scores of 4 or 5 may receive credit for CSC 111 and begin the computer science curriculum with CSC 112.
- take CSC 112 and CSC 201 as early as feasible. Completing these courses in the sophomore year provides the most flexibility in scheduling other required courses and electives. While not necessary, completing CSC 111, CSC 112, CSC 201, and MTH 117 by the end of the fall semester of the sophomore year provides the best flexibility, especially for the BS.
- consider taking another 200-level course in the sophomore year. Reviewing prerequisites for the 300-level electives may help in the decision regarding which 200-level courses to take early. For example, CSC 250 and CSC 251 are prerequisites for CSC 348.

To declare a major or minor in computer science, the student must have a minimum 2.0 GPA in computer science courses required for the major or minor including CSC 112.

Transfer Credit

A maximum of 7 hours at the 200-level or above may be transferred toward the BA or BS in computer science, with a maximum of 3 hours at the 300 level. For the computer science minor, all computer science electives at 193-level and above must be taken at Wake Forest.

Contact Information

Department of Computer Science (http://college.wfu.edu/cs/)
Manchester Hall 233, Box 7311
Phone 336-758-4982

Programs

Majors

- Computer Science, B.A. (https://bulletin.wfu.edu/undergraduate/departments-programs/computer-science/ba-computer-science/)
- Computer Science, B.S. (https://bulletin.wfu.edu/undergraduate/departments-programs/computer-science/bs-computer-science/)

Minor

- Computer Science, Minor (https://bulletin.wfu.edu/undergraduate/departments-programs/computer-science/minor-computer-science/)

Courses

Computer Science (CSC)

CSC 101. Overview of Computer Science. (3 h)
An introduction to fundamental principles of computer science including abstraction, data and information, the World Wide Web, algorithms, programs, creative computer use, and the global impact of computing. Students cultivate their understanding by working with data, logically solving problems individually and collaboratively, and writing simple computer programs. May not be counted towards the computer science major or minor. (D)

CSC 102. Problem Solving with Python. (3 h)
An introduction to fundamental principles of computer science including abstraction, data and information, algorithm development, and computer programming. Students apply computational problem solving skills to representative problems from a range of domains using the Python programming language. Appropriate for students who want computing experience applicable to other disciplines. May not be counted towards the computer science major or minor. (D)

CSC 111. Introduction to Computer Science. (4 h)
Introduction to computer programming and algorithmic problem solving in an object-oriented programming language. Topics include basic control structures, methods, parameters, objects, classes, arrays, and program testing and debugging. Recommended as the first course for students considering a major or minor in computer science. (D)

CSC 112. Fundamentals of Computer Science. (4 h)
A study of algorithm development and problem solving using top-down design, data abstraction, object-oriented programming, and program debugging and testing. Topics include memory allocation, recursion, data structures such as arrays, lists, stacks, queues, and trees, simple sorting and searching algorithms, and algorithm complexity. P-CSC 111 or POI. (D)

CSC 165. Problem Solving Seminar. (1 h)
A weekly seminar designed for students to develop their problem solving skills designing and implementing software. Does not count towards the computer science major or minor. May be taken twice. Pass/Fail. P-CSC 112 or POI.

CSC 181. Introduction to Computer Science Research. (3 h)
Introduction to concepts, conventions, and techniques used in computer science research. Students will learn research methods, evaluation of data, the presentation and dissemination of results, and research ethics within the context of computer science. Students interested in the Honors in Computer Science program are encouraged to take this course. Does not count towards the computer science major or minor. P-CSC 201 or POI.

CSC 191. Special Topics. (1-3 h)
Topics in computer science that are not covered in regular courses or that give special practice in skills used in other courses. Not to be counted toward the bachelor of science in computer science. May be repeated for up to 6 hours if the topic changes.

CSC 192. STEM Incubator. (1 h)
An engaging and relevant introduction to STEM (science, technology, engineering, and mathematics) through creative exploration, collaboration, and computational problem-solving. Pass/Fail. May be repeated once.

CSC 192H. Honors STEM Incubator. (1 h)
Leadership role in developing STEM (science, technology, engineering, and mathematics) ideas and applications through scientific exploration, creative collaboration, and computational problem-solving. For students with some programming experience. Pass/Fail. May be repeated once. P - POI.

CSC 193. Independent Study. (1-3 h)
A course of independent study directed by a faculty adviser, not to be counted toward the bachelor of science in computer science. May be repeated for up to 3 hours. Enrollment requires prearrangement with a computer science faculty member and departmental approval P-POI.

CSC 201. Data Structures and Algorithms. (3 h)
A study of fundamental data structures and the algorithms that act upon them. Data structure topics include the application, implementation, and complexity analysis of trees, hash tables, heaps, maps, sets, and graphs. Algorithmic topics include advanced sorting and searching methods and an introduction to divide-and-conquer and greedy techniques, graph algorithms, backtracking, and dynamic programming. P-CSC 112 and MTH 117. (D)

CSC 211. Computer Organization. (4 h)
Lecture and laboratory. Computer organization from the perspective of instructions, including the central processor, busses, input and output units, and memory units. A weekly two-hour laboratory covers combinational logic, loaders and linkers, assembly language, address computation, and other architecture-related functions. Lab-2 hours. P-CSC 111. (D)

CSC 221. Data Structures and Algorithms I. (3 h)
Analysis, implementation, and application of abstract data structures such as lists, stacks, queues, trees, hash tables, heaps, and grabs. Complexity analysis of algorithms that operate upon these structures. P-CSC 112. P or C-MTH 117. (D)

CSC 222. Data Structures and Algorithms II. (3 h)
Study of algorithms, algorithm design strategies, and the derivation of time complexity bounds. Case studies illustrate greedy algorithms, divide and conquer, backtracking, and dynamic programming techniques. An introduction to the classes P, NP, NP-complete, and Turing decidability is included. P-CSC 221 and MTH 111 or 112. (QR)

CSC 231. Programming Languages. (3 h)
Comparative study of programming language paradigms, including imperative languages, functional programming, logic programming, and object-oriented programming. Syntax, semantics, parsing, grammars, and issues in language design are covered. P-CSC 112 and MTH 117.

CSC 241. Computer Systems. (4 h)
Lecture and laboratory. Introduction to concepts of operating systems and networks including processor and memory management, concurrency, and protocol independent data communications. Lab-2 hours. P-CSC 112 and MTH 117.

CSC 250. Computer Systems I. (4 h)
Study of computer system mechanisms supporting program execution, with a focus on mapping programs to underlying hardware. Topics include data representation, assembly language, processor architecture, and the memory system, with an introduction to system-level programming. P-CSC 112 and MTH 117. (D)

CSC 251. Computer Systems II. (3 h)
Study of computer system mechanisms supporting program execution, with a focus on process and resource management. Topics include process control, virtual memory, concurrency, parallelism, file-based and network-based I/O, and additional coverage of system-level programming. P-CSC 250.

CSC 301. Algorithm Design and Analysis. (3 h)
A study of techniques for designing algorithms, analyzing their time and space complexity, and demonstrating their correctness. The algorithm design techniques include divide-and-conquer, greedy algorithms, dynamic programming, randomized algorithms, string processing algorithms, and parallel algorithms. The algorithm analysis includes computational models, best/average/worst case analysis, and computational complexity (including lower bounds and NP-completeness). P-CSC 201 and MTH 112. (QR)

CSC 311. Computer Architecture. (3 h)
An in-depth study of computer system and architecture design. Topics include processor design, memory hierarchy, external storage devices, interface design, and parallel architectures. P-CSC 211 or 250.

CSC 321. Database Management Systems. (3 h)
Introduction to database management systems. Topics include data independence, database models, query languages, security, integrity, and transactions. P-CSC 201 or 221.

CSC 322. Data Management and Analytics. (3 h)
Management, analysis, and visualization of large-scale data sets. Topics includes key-value databases, distributed file systems, map-reduce techniques, similarity measures, link analysis, and clustering. P - CSC 321.

CSC 331. Software Engineering. (3 h)
Study of fundamental topics in software engineering including software processes, agile software development and project management, requirements engineering, system modeling, design patterns and implementation, and software testing. Students practice software engineering principles through team projects. P-CSC 201 or 221.

CSC 332. Mobile and Pervasive Computing. (3 h)
Study of the fundamental design concepts and software principles underlying mobile and pervasive computing, including mobile interface design, data management, mobile networks, location aware computing, and mobile security. Involves significant programming on modern mobile platforms. P-CSC 201 or 221.

CSC 333. Principles of Translators for Compilers and Interpreters. (3 h)
Study of techniques for translating high-level programming languages to a target language. Typical target languages include Java bytecode and assembly language. Topics include lexical analysis, parsing, intermediate representations, language semantics, code generation, and optimization. P-CSC 231 and either CSC 211 or 250.

CSC 341. Operating Systems. (3 h)
Study of the different modules that compose a modern operating system. In-depth study of concurrency, processor management, memory management, file management, and security. P-CSC 241 or 251.

CSC 343. Internet Protocols. (3 h)
Study of wide area connectivity through interconnection networks. Emphasis is placed on Internet architecture and protocols. Topics include addressing, routing, multicasting, quality of service, and network security. P-CSC 241 or 251.

CSC 346. Parallel Computation. (3 h)
Study of techniques for parallel and high performance computing. Topics include an overview of modern high-performance computer design, pipelining, concurrency, data dependency, shared memory, message passing, and graphics processors. Select parallel algorithms and methods for asymptotic scalability analysis are also presented. Assignments may include coding with OpenMP, MPI, and the CUDA library. P-CSC 201 or 221.

CSC 347. GPU Programming. (3 h)
An introduction to general purpose parallel program development on Graphics Processing Units (GPUs). Topics covered will include data parallelism, memory and data locality, parallel algorithm patterns and performance metrics, and application test studies. P-CSC 201 or 221 and MTH 121 or 205.

CSC 348. Computer Security. (3 h)
Introduction to computer security concepts and associated theory. Detailed coverage of the core concepts of access control, cryptography, trusted computing bases, digital signatures, authentication, network security, and secure architectures. Legal issues, security policies, risk management, certification and accreditation are covered in their supporting roles. Students will learn to analyze, design, and build secure systems of moderate complexity. P-CSC 241 or 251.

CSC 352. Numerical Linear Algebra. (3 h)
Numerical methods for solving matrix and related problems in science and engineering using a high-level matrix-oriented language such as MATLAB. Topics include systems of linear equations, least squares methods, and eigenvalue computations. Special emphasis given to applications. Also listed as MTH 326. P-MTH 112; and MTH 121 or 205. (D)

CSC 355. Introduction to Numerical Methods. (3 h)
Numerical computations on modern computer architectures; floating-point arithmetic and round-off error. Programming in a scientific/engineering language such as MATLAB, C, or FORTRAN. Algorithms and computer techniques for the solution of problems such as roots of functions, approximation, integration, systems of linear equations, and least squares methods. Also listed as MTH 355. P-MTH 112; and MTH 121 or 205. (D)

CSC 361. Digital Media. (3 h)
Study of the mathematics and algorithms underlying digital sounds, image, and video manipulation. Topics may include sampling and quantization, resolution, filters, transforms, data encoding and compression, multimedia file types and transmission, 3D printing, and digital media in multimedia and web programming. P-CSC 201 or 221 and MTH 111 or 112.

CSC 363. Computer Graphics. (3 h)
Study of software and hardware techniques in computer graphics. Topics include line and polygon drawing, hidden line and surface techniques, transformations, and ray tracing. P-CSC 201 or 221; and either MTH 121 or 205.

CSC 365. Image Processing Fundamentals. (3 h)
Study of the basic theory and algorithms for image enhancement, restoration, segmentation, and analysis. P-CSC 112; and MTH 121 or 205.

CSC 371. Artificial Intelligence. (3 h)
An overview of areas of study in artificial intelligence. Topics are chosen from among knowledge representation, formal logic, fuzzy logic, intelligent agents, expert systems, machine learning, robotics, and natural language processing. P-CSC 201 or 221.

CSC 373. Data Mining. (3 h)
An overview of data mining methods and algorithms for classification, association analysis, clustering,and anomaly detection. A major focus will be on the implementation of algorithms for and design and construction of solutions to data mining problems. Applications and ethical considerations of data mining in humanities, arts, and healthcare are discussed. P-CSC 201 or 221.

CSC 374. Machine Learning. (3 h)
An introduction to concepts and application of machine learning algorithms and techniques, focusing on supervised and unsupervised learning. Students learn the theoretical concepts behind several types of machine learning algorithms and gain practical experience applying them. Algorithms covered could include logistic regression, support vector machines, regularization, dimensional reduction, clustering, and neural networks. P-CSC 201 or 221, MTH 112 and MTH 121 or 205.

CSC 375. Neural Networks and Deep Learning. (3 h)
An introduction to concepts and applications of neural networks and deep learning, a branch of machine learning that uses additional layers of high-level representations of data to maximize performance on a given task. The topics covered may include basic neural networks, deep neural networks, and convolutional and recurrent neural networks. Students learn the theoretical concepts behind several types of neural network algorithms and gain practical experience applying them. P-CSC 201 or 221, MTH 112, and MTH 121 or 205.

CSC 385. Bioinformatics. (3 h)
Introduction to computational approaches essential to modern biological inquiry. Approaches may include large biological dataset analyses, sequence similarity and motif searches, and analysis of high-throughput genomic technologies. Emphasizes interdisciplinary interaction and communication. Also listed as BIO 385 and PHY 385. P-CSC 201 or 221 (and BIO 150, 150L, 160, and 160L if taken as BIO 385) or POI.

CSC 387. Computational Systems Biology. (3 h)
Introduction of concepts and development of skills for comprehension of systems biology problems, including both biological and computational aspects. Topics may include genome-wide transcriptomic analysis, protein interaction networks, large-scale proteomics experiments, and computational approaches for modeling, storing, and analyzing the resulting data sets. Emphasizes interdisciplinary interaction and communication. Also listed as BIO 387. P-CSC 201 or 221 (also requires BIO 150, 150L, 160, and 160L if taken as BIO 387) or POI.

CSC 391. Selected Topics. (1-3 h)
Topics in computer science that are not studied in regular courses or which further examine topics covered in regular courses. May be repeated if topic changes. P - any 200-level CSC course and POI.

CSC 393. Individual Study. (1-3 h)
Independent study directed by a faculty adviser. By prearrangement. No more than three hours may be counted toward a computer science major or minor. Enrollment requires prearrangement with a computer science faculty member and departmental approval. P-any 200-level CSC course and POI.

CSC 399. Computer Science Mastery Exam. (0 h)
Evaluation of student mastery of core topics in the computer science discipline through standardized testing. Taken during the senior year. Pass/Fail.

Faculty

Chair William Turkett
Reynolds Professor of Computer Science and Mathematics (Emeritus)
Robert Plemmons

Professors Errin Fulp, V. Paúl Pauca, Peter Santago
Professors Emeritus Jennifer J. Burg, David John
Associate Professors Grey Ballard, Daniel Cañas, Samuel Cho, William Turkett
Associate Professor Emeritus Stan Thomas
Assistant Professors Sarra Alqahtani, Minghan Chen, Aditya Devarakonda, Xueuyan (Michael) Han, Natalia Khuri, Fan Yang
Assistant Teaching Professor William Cochran
Katherine and Dickerson Wright Presidential Chair in Computer Science and Entrepreneurship Ron Doyle
Affiliate Professor Jennifer Erway
Affiliate Associate Professor James Pease
Affiliate Assistant Professor Kyle Luthy
Adjunct Professors Sean Anderson, Bruno Belkhiter, Adam Carlson, Sami Khuri, Kelly Kuykendall, Rob Robless, Sarah Parsons, Cody Stevens