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 MST 117 in their freshman year. Students with appropriate experience or AP scores may skip CSC 111 and go straight to CSC 112. Credit is given for CSC 111 with an AP score of 5. With an AP score of 4, students may skip CSC 111 but will not receive credit.
- take CSC 112 and CSC 221 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 221, and MST 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 241 is a prerequisite for CSC 348.

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

- B.S. in Computer Science
- B.A. in Computer Science

Minor

- Minor in Computer Science

Courses

Computer Science (CSC)

CSC 101. Overview of Computer Science. (4 h)
Lecture and laboratory. Overview of computer science. Students are introduced to the core areas of computer science. Topics include data representation, logic, computer organization, pseudo-code, machine/assembly code, higher-level language, algorithms, abstract data types, operating systems, and networks. Algorithms and programming are introduced through analysis of existing code. Lab-2 hours. (D, QR)

CSC 111. Introduction to Computer Science. (4 h)
Lecture and laboratory. Introduction to the basic concepts of computer programming and algorithmic problem solving for students with little or no programming experience. Recommended as the first course for students considering a major or minor in computer science, also appropriate for students who want computing experience applicable to other disciplines. Lab—2 hours. (D)

CSC 112. Fundamentals of Computer Science. (4 h)
Lecture and laboratory. Problem solving and program construction using top-down design, data abstraction, and object-oriented programming. Memory addressing, dynamic memory allocation, and linear data structures are introduced. Lab-2 hours. 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 111 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) ideas and applications through scientific exploration, creative exploration, 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 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 and MST 117. (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-MST 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 MST 111 or 112. (QR)

CSC 231. Programming Languages. (4 h)
Lecture and laboratory. A 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. Lab-2
hours. P-CSC 112 and MST 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 MST 117.

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.

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 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 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 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 211 and 231.

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.

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.

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 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 221 and either
MST 121, 205 or 206.

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

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 MST 326. P-MST 112; and MST 121, 205 or
206. (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 MST 355. P-MST 112; and
MST 121, 205 or 206. (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 112 and
MST 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 221; and MST 121, 205 or 206.

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 MST 121, 205 or
206.
CSC 371. Artificial Intelligence. (3 h)
Introduction to problems in artificial intelligence. Topics include
knowledge representation, heuristic search, formal logic, planning,
robotics, machine learning, intelligent agents, and pattern recognition. P-
CSC 221.

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 221 or
BIO 213 or POI.

CSC 387. Computational Systems Biology. (3 h)
Introduction of concepts and development of skills necessary for
comprehension of systems biology research 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. P - CSC 221 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 Peter Santago
Research Professor Robert J. Plemmons
Professors Jennifer J. Burg, Errin W. Fulp, David J. John, V. Paúl Pauca,
Peter Santago
Associate Professors Daniel A. Cañas, Samuel Cho, Stan J. Thomas, Todd
Assistant Professor Grey Ballard
Visiting Assistant Professors Rongzhong Li, Sai Mukkavilli