COMPUTER SCIENCE 198

Web Site: *http://cs.camden.rutgers.edu*

Major requirements in computer science can be completed either through daytime or evening attendance.

Department of Computer Science

Chairperson: Jean-Camille Birget

Professor:

Michael A. Palis, *B.S.E.E., Philippines; Ph.D., Minnesota*

Associate Professors:

Jean-Camille Birget, *B.S., Ecole Centrale de Paris; M.S., Ph.D., California (Berkeley)*
Guy Kortsarz, *Ph.D., Weizmann*
Sunil M. Shende, *B.Tech., Indian Institute of Technology; Ph.D., Pennsylvania*

Assistant Professors:

Dawei Hong, *Ph.D., Nebraska (Lincoln)*
Patricia Johann, *B.A., Reed; Ph.D., Wesleyan*
Suneeta Ramaswami, *B.A., Wellesley College; Ph.D., Pennsylvania*

The Department of Computer Science offers a comprehensive program that prepares students for professional careers and/or graduate studies in computer science. The curriculum has been developed in accordance with the recommendations of the Computer Science Accreditation Commission (CSAC) of the Computer Science Accreditation Board (CSAB) and the ACM/IEEE-CS Joint Curriculum Task Force.

Students majoring in computer science may choose between two degree programs: the bachelor of science (B.S.) and the bachelor of arts (B.A.). The two programs have identical computer science requirements and differ only in the natural science and mathematics requirements. The B.S. program requires a minimum of 33 credits in science and mathematics and is intended for students planning to pursue careers and/or graduate studies in science and engineering. The B.A. program has fewer science and mathematics requirements (24 credits), thus allowing students greater opportunity to develop breadth of knowledge in other disciplines, such as the arts, humanities, social sciences, and business.

Major Requirements: CCAS and UC–C

Students wishing to pursue either the B.S. or B.A. program must satisfy the following requirements before formally being admitted into the program:

1. A grade of 2.5 *(C+)* or better in 50:198:111 and 113. For transfer students, a grade of 2.5 or better in the first two computer science courses (each at least 3 credits) taken at Rutgers, not including 50:198:110 and 151.
2. Completion or transfer credits for either 50:640:121 or 130.

To continue in the program and graduate with a degree in computer science, a student must achieve a grade of 2.0 *(C)* or better in all computer science courses required for the major.

Bachelor of Science (B.S.) in Computer Science

Students pursuing the B.S. program in computer science must complete the requirements listed below.

1. *General Curricular Requirements*
   All students must satisfy the general curricular requirements listed in the Degree Requirements chapter of this catalog. Some of the general curricular requirements are automatically satisfied by fulfilling the natural science and mathematics requirements of the B.S. program. The following guidelines should be used:
   a. 6 credits of the B.S. natural science requirements may be used to satisfy the general curricular requirement of 6 credits from the offerings of the natural science disciplines.
   b. 6 credits of the B.S. mathematics requirements may be used to satisfy the general curricular requirement of 3 credits in mathematics and 3 credits in mathematics, computer science, or statistics.
   c. At most, 6 additional credits of the B.S. natural science or mathematics requirements, excluding the courses used to satisfy a and b above, may be counted toward fulfillment of the general curricular requirement of 9 credits in free electives outside the major department. Students *must* take at least 3 more credits outside computer science to satisfy the free electives general curricular requirement.
2. *Natural Science Requirements*
   50:750:131 Elements of Physics I (3)
   50:750:132 Elements of Physics II (3)
   50:750:133 Elements of Physics Laboratory I (1)
   50:750:134 Elements of Physics Laboratory II (1)
   Minimum of 4 additional credits of natural science electives in the biological sciences, chemistry, or physics; courses designed for nonscience majors may *not* be used to satisfy this requirement.
3. *Mathematics Requirements*
   50:640:121, 122, 221 Unified Calculus I,II,III (4,4,4)
   50:640:237 Discrete Mathematics (3)
   50:960:336 Applied Statistics (3) *or* 50:960:283, 284 Introduction to Statistics I,II (3,3)
   Minimum of 3 additional credits in mathematics or statistics at the 200 level or higher.
4. *Computer Science Core*
   50:198:111 Introduction to Computer Science (3)
   50:198:112 Software Laboratory I (1)
   50:198:113 Programming with Data Structures (3)
   50:198:114 Software Laboratory II (1)
   50:198:221 Programming Language Concepts (3)
   50:198:231 Computer Organization and Assembly Language Programming (3)
   50:198:271 Design and Analysis of Algorithms (3)
   50:198:323 Software Methodology and Engineering (3)
   50:198:333 *or* 50:750:308 Computer Hardware and Interfacing (3)
   50:198:334 *or* 50:750:312 Computer Hardware and Interfacing Laboratory (1)
   50:198:341 Principles of Operating Systems (3)
   50:198:376 Introduction to the Theory of Computation (3)
   50:198:493 Senior Design Project (3)
5. *Computer Science Electives*
   At least 15 credits of computer science electives at the 300 or 400 level. At most, 3 credits of 50:198:494 Independent Study and, at most, 3 credits of 50:198:497 Computer Science Internship may be counted toward fulfillment of this requirement.

Bachelor of Arts (B.A.) in Computer Science

Students pursuing the B.A. program in computer science must complete the requirements listed below.

1. *General Curricular Requirements*
   All students must satisfy the general curricular requirements listed in the Degree Requirements chapter of this catalog. Some of the general curricular requirements are automatically satisfied by fulfilling the natural science and mathematics requirements of the B.A. program. The following guidelines should be used:
   a. 6 credits of the B.A. natural science requirements may be used to satisfy the general curricular requirement of 6 credits from the offerings of the natural science disciplines.
   b. 6 credits of the B.A. mathematics requirements may be used to satisfy the general curricular requirement of 3 credits in mathematics and 3 credits in mathematics, computer science, or statistics.
   c. Students *must* take at least 9 additional credits in courses outside computer science to satisfy the free electives general curricular requirement. These courses should be different from the courses used to satisfy the natural science or mathematics requirements given below.
2. *Natural Science Requirements*
   Minimum of 9 credits of natural science electives in the biological sciences, chemistry, or physics (including astronomy and geology).
3. *Mathematics Requirements*

50:640:129	Linear Mathematics for Business and Economics (3)
50:640:130	Calculus for Business, Economics, and Life Sciences (3)
50:640:237	Discrete Mathematics (3)
50:960:336	Applied Statistics (3) *or* 50:960:283,284 Introduction to Statistics I,II (3,3)

   Minimum of 3 additional credits in mathematics or statistics at the 200 level or higher.
4. *Computer Science Core*
   Identical to the B.S. in computer science core requirements.
5. *Computer Science Electives*
   Identical to the B.S. in computer science electives requirements.

Technical Tracks

In lieu of upper-division technical electives, students may pursue one of two technical tracks: the Scientific Computing Track or the Information Systems Track.

Scientific Computing Track

The Scientific Computing Track teaches students the mathematical foundations, methods, and tools of scientific computing and how they can be applied toward the solution of scientific and engineering problems. Students gain experience in the use of high-performance computing equipment and state-of-the-art scientific software and visualization tools of the Science Vision Laboratories. This track is open only to students pursuing the B.S. program and may be taken only through daytime attendance.

1. Required track courses:

50:198:316	Parallel Programming (3)
50:198:381	Introduction to Numerical Methods (3)
50:198:458	Scientific Visualization (3)
50:198:481	Advanced Numerical Methods (3)

   One 300- or 400-level department-approved course in computer science, mathematics, or natural science
2. Required math courses:

50:640:250	Linear Algebra (3)
50:640:314	Elementary Differential Equations (3)

Information Systems Track

The Information Systems Track is intended for students who wish to pursue careers in information systems practice in industry or government. It provides students with the requisite knowledge and skills to carry out such responsibilities as development of systems architecture, systems standards, database design and implementation, network design, integration, testing, and operational management of information systems. This track is open to students pursuing either the B.S. or the B.A. program in computer science.

1. Required track courses:

50:198:346	Computer Networks (3)
50:198:347	Computer Systems Administration (3)
50:198:351	Database Systems (3)
50:198:426	Information Systems Analysis and Design (3)

   One 300- or 400-level computer science elective
2. Business electives:
   At least two courses from the following list:

52:010:101	Introduction to Financial Accounting (3)
52:010:202	Management Accounting (3)
50:220:105	Microeconomic Principles (3)
50:220:106	Macroeconomic Principles (3)
52:390:301	Principles of Finance (3)
52:623:334	Management Information Systems (3)

Minor Requirements: CCAS and UC–C

Students desiring a minor in computer science must complete the following courses with a grade of C or better:

50:198:111	Introduction to Computer Science (3)
50:198:112	Software Laboratory I (1)
50:198:113	Programming with Data Structures (3)
50:198:114	Software Laboratory II (1)
50:198:221	Programming Language Concepts (3)
50:198:231	Computer Organization and Assembly Language Programming (3)

At least two additional computer science courses numbered 200 or above.

Departmental Honors Program

In lieu of 50:198:493 Senior Design Project, a student may complete an undergraduate thesis embodying original research work through the Honors Program. Eligibility in the program is judged by the student's academic performance and the availability of an appropriate supervising professor. Approval must be obtained from the department chair and the faculty member who is to serve as the student's adviser. Students admitted to the program must complete a minimum of 6 credits of the two-term sequence 50:198:495-496 Honors Program in Computer Science. A grade is not given until completion of the sequence and department approval of an undergraduate thesis embodying the project. Students who successfully complete the Honors Program are given the distinction of graduating with *Honors in Computer Science.*

Courses

50:198:110. Introduction to Computing (R) (3)
Prerequisite: Satisfy mathematics requirement for admission to the colleges.
Introduction to computers and information technology, with an emphasis on personal computers and their applications. Understanding of and hands-on experience with application software including word processors, spreadsheets, database systems, electronic mail, and web browsers. Introductory problem solving and computer programming.

50:198:111. Introduction to Computer Science (R) (3)
Corequisites: 50:640:121 or 129 or 130, and 50:198:112.
The first course for computer science majors. Basic concepts of computing, fundamental problem-solving techniques, and principles of structured programming. Programming in a modern high-level language (such as C/C++ or Java). Must be taken concurrently with 50:198:112 Software Laboratory I.

50:198:112. Software Laboratory I (R) (1)
Corequisite: 50:198:111.
Formal laboratory that provides practical programming experience based on the lecture material of 50:198:111. Introduction to the Unix system, including the file system; programming tools such as editors, compilers, and debuggers; libraries; and other utilities.

50:198:113. Programming with Data Structures (R) (3)
Prerequisites: 50:198:111 and 50:640:121 or 129 or 130. Corequisite: 50:198:114.
Abstract data types and elementary data structures, including stacks, queues, linked lists, tree-based structures, and hash tables. Algorithmic analysis, recursion, searching and sorting, and basic algorithms. Principles of object-oriented programming and design illustrated through a modern object-oriented language (such as C++ or Java).

50:198:114. Software Laboratory II (R) (1)
Prerequisite: 50:198:112. Corequisite: 50:198:113.
Formal laboratory that provides practical programming experience based on the lecture material of 50:198:113. Introduction to Unix systems programming, Unix systems calls, shell scripts, and graphical user interface (GUI) libraries and tools.

50:198:151. Introduction to Programming Methods Using FORTRAN (R) (3)
Corequisite: 50:640:121 or 130.
Intended for science and engineering majors. Basic problem-solving and programming techniques, basic data structures, and fundamental numerical algorithms. Programming in FORTRAN.

50:198:221. Programming Language Concepts (3)
Prerequisite: 50:198:113.
Design issues relevant to the implementation of programming languages: sequence control, data control, type checking, run-time storage management, language translation, and semantics. Comparison of major programming paradigms: procedural, functional, logic, and object-oriented. Language features that support parallel and distributed computing. Introduction to the relational data model and relational database query languages.

50:198:231. Computer Organization and Assembly Language Programming (3)
(Formerly 50:198:211 Introduction to Computer Architecture and Assembly Language Programming)
Prerequisite: 50:198:111.
Machine representation of data and instructions, instruction set architecture, memory organization, input/output, and interrupt processing. Assembly language and machine language, relation to high-level language, and operating system interface. Assembly language programming, including subroutine construction, macros, debugging, linking, and loading.

50:198:271. Design and Analysis of Algorithms (3)
Prerequisites: 50:198:113 and 50:640:237.
Algorithm design techniques: divide-and-conquer, greedy method, dynamic programming, backtracking, and branch-and-bound. Advanced data structures, graph algorithms, algebraic algorithms. Complexity analysis, complexity classes, and NP-completeness. Introduction to approximation algorithms and parallel algorithms.

50:198:316. Parallel Programming (3)
Prerequisites: 50:198:113 or advanced programming experience and permission of instructor.
Fundamental issues in the design and development of parallel programs for various types of parallel computers. Various programming models according to both machine type and application area. Cost models, debugging, and performance evaluation of parallel programs with actual application examples. Programming techniques and optimization. Programming exercises on a contemporary parallel machine.

50:198:323. Software Methodology and Engineering (3)
Prerequisite: 50:198:221.
Principles and techniques for the design and construction of reliable, maintainable, and useful software systems. Software life cycle, requirements specifications, and verification and validation issues. Implementation strategies (e.g., top-down, bottom-up, teams), support for reuse, performance improvement. A treatment of human factors and user interfaces included.

50:198:333. Computer Hardware and Interfacing (3)
Prerequisites: 50:198:231 and 50:640:237. Corequisite: 50:198:334. Credit not given for both this course and 50:750:308.
Introduction to digital logic, combinational circuits, sequential circuits. Introduction to microprocessor architecture and organization, operation and programming, interfacing, and application of microprocessors.

50:198:334. Computer Hardware and Interfacing Laboratory (1)
Corequisite: 50:198:333. Credit not given for both this course and 50:750:312.
Provides hands-on experience in digital design using PLA/PLD devices, EEPROM, and MSI/LSI circuits, and interfacing of microprocessors to memory and peripherals.

50:198:341. Principles of Operating Systems (3)
Prerequisites: 50:198:113 and 231.
Fundamental concepts of operating systems. Process management, memory management, device management, file systems, resource allocation, security and protection. Introduction to network and distributed operating systems.

50:198:346. Computer Networks (3)
Prerequisite: 50:198:113.
Introduction to computer communication networks, including physical and architectural components, communication protocols, switching, network routing, congestion control, and flow control. End-to-end transport services, network security and privacy. Networking software and applications. Network installation, testing, and maintenance.

50:198:347. Computer Systems Administration (3)
Prerequisites: 50:198:113 and 343.
Basic administration of networked computer systems (such as Unix and/or Windows NT). Installing and configuring the operating system, upgrading software and hardware, installing patches, system backups, security issues, account creation and deletion, system accounting and log files, job scheduling, performance monitoring, tcp/ip and networking, client/server file sharing and printing, file layout and organization, disk and tape administration, and a look at several administrative tools.

50:198:351. Database Systems (3)
Prerequisites: 50:198:113 and 50:640:237.
Relational database theory and practice, including database design. Database concepts, relational algebra, data integrity, query languages, views. Introduction to object-oriented databases. Application project with a practical database management system.

50:198:356. Computer Graphics (3)
Prerequisite: 50:198:113.
Characteristics of graphics display devices and systems; representation, manipulation, and display of two- and three-dimensional objects; curve and surface modeling; two- and three-dimensional transformations; hidden lines and surfaces; shading and coloring; interactive graphics and user interfaces; animation techniques.

50:198:361. Artificial Intelligence (3)
Prerequisites: 50:198:113 and 50:640:237.
Techniques and applications of artificial intelligence: search, rule-based reasoning, statistical reasoning, game playing, machine learning, knowledge representation. The use of heuristics to obtain satisfactory solutions to intractable problems.

50:198:376. Introduction to the Theory of Computation (3)
Prerequisite: 50:640:237.
Introduction to the theory of computability, including important results from the study of automata and formal languages. Automata and their relationship to regular, context-free, and context-sensitive languages. General theories of computability, including Turing machines, recursive functions, and lambda calculus. Notions of decidability and undecidability, complexity classes, and complexity analysis.

50:198:381. Introduction to Numerical Methods (3)
Prerequisites: 50:198:111 and 50:640:221.
Methods of finding roots, interpolation, curve fitting, integration, differentiation, and minimization; estimation and control of various computational errors.

50:198:421. Compiler Construction (3)
Prerequisites: 50:198:221, 231, and 376.
Introduction to compiler design and implementation, including lexical analysis, formal syntax specification, parsing techniques, syntax-directed translation, semantic analysis, execution environment, storage management, code generation, and optimization techniques.

50:198:426. Information Systems Analysis and Design (3)
Prerequisite: 50:198:351.
Overview of information systems; analysis of existing systems; requirements determination; design of information systems including interface, inputs, outputs, database. Group analysis and design project. Alternative development methodologies. Modern tools for analysis and design.

50:198:431. High-Performance Computer Architectures (3)
Prerequisites: 50:198:231 and 333.
Architecture of pipelined and superscalar processors, vector supercomputers, multiprocessors, parallel computers, multithreaded processors, and dataflow processors. System interconnect architectures, memory hierarchy design, and input/output subsystems. System design issues such as shared memory and message-passing communication models, cache coherence and synchronization mechanisms, latency-hiding techniques, virtual memory management, and task scheduling.

50:198:441. Distributed Systems (3)
Prerequisite: 50:198:341.
Models of distributed systems, distributed algorithms and protocols, operating systems support, programming paradigms. Case studies of experimental and commercial systems.

50:198:458. Scientific Visualization (3)
Prerequisites: 50:198:113, 381, and 50:640:250.
Study of visualization techniques useful for the analysis and interpretation of scientific and engineering data. Topics include two- and three-dimensional data types; visual representation schemes for scalar, vector, and tensor data; isosurface and volume visualization methods; animation; and interactive manipulation of data.

50:198:471. Advanced Algorithms (3)
Prerequisite: 50:198:271.
Advanced and specialized topics in algorithms, selected from parallel algorithms, randomized algorithms, combinatorial optimization, or other.

50:198:473. Introduction to Computational Geometry (3)
Prerequisite: 50:198:271.
Algorithms and data structures for geometric problems that arise in various applications, such as computer graphics, CAD/CAM, robotics, and geographical information systems (GIS). Topics include point location, range searching, intersection, decomposition of polygons, convex hulls, and Voronoi diagrams.

50:198:475. Cryptography and Computer Security (3)
Prerequisite: 50:198:271.
Secret-key cryptography, public-key cryptography, key arrangement, secret sharing, digital signatures, message and user authentication, one-way functions, key management; attacks; practical applications to computer and communication security.

50:198:481. Advanced Numerical Methods (3)
Prerequisites: 50:198:381, 50:640:250 and 314.
Numerical techniques for solving linear algebraic systems, eigenvalue problems, least squares, quadrature. Numerical solution of initial and boundary value problems for ordinary and partial differential equations.

50:198:483. Linear Programming (3)
Prerequisites: 50:640:221 and 250.
Linear programming using the simplex method and dual linear programming. Applications of linear programming to network flows and other problems. Convex sets and elements of convex programming. Combinatorial optimization. Integer programming.

50:198:485. Computational Mathematics (3)
Prerequisite: 50:640:250. Credit not given for both this course and 50:640:498.
Computational aspects of number theory, with a brief introduction to underlying theories. Topics include prime numbers, pseudo primes, and their applications especially in cryptography; prime factorization of composite numbers via several different methods. Computer simulation emphasized.

50:198:487. Visualizing Mathematics by Computer (3)
Prerequisite: 50:640:221. Credit not given for both this course and 50:640:497.
A comprehensive introduction to symbolic computational packages and scientific visualization through examples from calculus and geometry. Covers 2-D, 3-D, and animated computer graphics using Maple, Mathematica, and Geomview.

50:198:491,492. Special Topics in Computer Science (3,3)
Prerequisite: As announced or permission of instructor.
In-depth study of areas not covered in regular curriculum. Topics vary from term to term.

50:198:493. Senior Design Project (3)
Prerequisite: Senior standing or permission of instructor.
Design, implementation, and demonstration of a significant software and/or hardware project. Project proposals must be submitted and approved by instructor. Part of the lecture time used to discuss such issues as the historical and social context of computing, responsibilities of the computing professional, risks and liabilities, and intellectual property.

50:198:494. Independent Study (BA)
Prerequisite: Permission of instructor.
Individual study under the supervision of a computer science faculty member; intended to provide an opportunity to investigate areas not covered in regular courses.

50:198:495-496. Honors Program in Computer Science (BA,BA)

Prerequisite: Approval by department.

A program of readings and guided research in a topic proposed by the student, culminating in an honors thesis presented to the departmental faculty for approval.

50:198:497. Computer Science Internship (BA)

Prerequisite: Approval by department.

The practical application of computer science knowledge and skills through an approved internship in a sponsoring organization. Arrangements for the internship must be agreed upon by the sponsoring organization and approved by the department before the beginning of the term. Students should consult the department for detailed instructions before registering for this course.