

Computer Science

As the largest computer science department among research universities in the New York metropolitan area, the Department of Computer Science is a bustling stomping ground for students looking to explore computer science on both a technical and liberal arts level.

The department offers B.S., B.A., M.S. and Ph.D. degree programs in computer science and evolving interdisciplinary programs like telecommunication, bioinformatics and computing and business. The B.S. provides a more technical exploration of computer science and is excellent for students planning to pursue advanced study in computer science while the B.A. program is designed for students interested in liberal arts or management. There's also the opportunity to learn beyond the classroom via the Senior Capstone Program, a semester-long course that gives students the chance to draw on their years of studying and apply said knowledge to one of NJIT's 200 industry sponsors and partners, which include Microsoft, Johnson & Johnson and NASA.

NJIT Faculty

B

Basu Roy, Senjuti, Assistant Professor

Borcea, Cristian M., Professor

C

Calvin, James M., Professor

Curtmola, Reza, Associate Professor

D

Dass, Ananya, University Lecturer

Ding, Xiaoning, Assistant Professor

E

Eljabiri, Osama, Senior University Lecturer

G

Gehani, Narain, Professor, Emeritus

Geller, James, Professor

Gerbessiotis, Alexandros, Associate Professor

Gotsman, Craig J., Distinguished Professor and Dean

H

Hung, Daochuan, Associate Professor

I

Itani, Abdul-Rahman M., Faculty

K

Kapleau, Jonathan, J., University Lecturer

Karvelas, Dionissios, Senior University Lecturer

Koutis, Ioannis, Professor

Kwestel, Morty D., Senior University Lecturer

L

Leung, Joseph Y., Distinguished Professor

Li, Jing, Professor

Liu, Chengjun, Professor

M

Mani, Kumar, Professor

McHugh, James, Professor

Mili, Ali, Professor

N

Nakayama, Marvin K., Professor

Nassimi, David, Associate Professor

Neamtiu, Iulian, Associate Professor

Nicholson, Theodore L., Senior University Lecturer

O

Oria, Vincent, Professor

P

Perl, Yehoshua, Professor

Polyakov, Yuriy S., Associate Research Professor

R

Rohloff, Kurt, Associate Professor

Roshan, Usman W., Associate Professor

Rusinkiewicz, Marek E., Professor

Rutkowski, Wallace, Senior University Lecturer

Ryan, Gerard W., Senior University Lecturer

S

Shih, Frank Y., Professor

Sohn, Andrew, Associate Professor

Spirollari, Junilda, Senior University Lecturer

T

Tang, Qiang, Assistant Professor

Theodoratos, Dimitrios, Associate Professor

Thomson, Susan E., Senior University Lecturer

W

Wang, Jason, T., Professor

Wang, Guiling (Grace), Professor

Wei, Zhi, Associate Professor

Wu, Chase Q., Associate Professor

Programs

- Bioinformatics - B.S. (<http://catalog.njit.edu/undergraduate/computing-sciences/computer-science/bioinformatics-bs>)

- Computer Science - B.A. (<http://catalog.njit.edu/undergraduate/computing-sciences/computer-science/ba>)
- Computer Science - B.S. (<http://catalog.njit.edu/undergraduate/computing-sciences/computer-science/bs>)
- Computing and Business - B.S. (<http://catalog.njit.edu/undergraduate/computing-sciences/computer-science/computing-business-bs>)

Accelerated Programs (<http://catalog.njit.edu/undergraduate/academic-policies-procedures/special-degree-options>)

- Bioinformatics for Honors Premed Students - Accelerated B.S. (<http://catalog.njit.edu/undergraduate/computing-sciences/computer-science/accelerated-bioinformatics-honors-premed-bs>)

Double Majors (<http://catalog.njit.edu/undergraduate/academic-policies-procedures/special-degree-options>)

- Computer Science and Applied Physics - B.S. (<http://catalog.njit.edu/undergraduate/computing-sciences/computer-science/cs-applied-physics-bs>)
- Computer Science and Mathematical Sciences - B.S. (<http://catalog.njit.edu/undergraduate/computing-sciences/computer-science/cs-math-bs>)
- Computer Science Minor (<http://catalog.njit.edu/undergraduate/computing-sciences/computer-science/minor>) (not for Computer Engineering majors)
- Computer Science Minor (<http://catalog.njit.edu/undergraduate/computing-sciences/computer-science/minor-computer-engineering>) (for Computer Engineering majors)

Computer Science Courses

BNFO 135. Programming for Bioinformatics. 3 credits, 3 contact hours (3;0;0).

The ability to use existing programs and to write small programs to access bioinformatics information or to combine and manipulate various existing bioinformatics programs has become a valuable part of the skill set of anyone working with biomolecular or genetic data. This course provides an understanding of the architecture of bioinformatics toolkits and experience in writing small bioinformatics programs using one or more of the scripting ("glue") languages frequently employed for such tasks.

BNFO 236. Programming For Bioinfo II. 3 credits, 3 contact hours (3;0;0).

BNFO 330. Princ of Bioinformatics II. 3 credits, 3 contact hours (3;0;0).

BNFO 340. Data Analysis for Bioinformatics II. 3 credits, 3 contact hours (3;0;0).

Prerequisites: BNFO 240 and R120 101 or equivalent or permission of instructor. Advanced data analysis skills with applications to bioinformatics problems.

BNFO 482. Databases and Data Mining in Bioinformatics. 3 credits, 3 contact hours (3;0;0).

Prerequisites: BNFO 240 or equivalent or permission of instructor. Surveys biological databases and tools for managing them. Covers concepts and principles of data mining in bioinformatics. Hands-on experience for mining genomic data using ORACLE and SQL.

BNFO 488. Independent Study. 3 credits, 3 contact hours (0;0;3).

BNFO 491. Computer Science Project. 3 credits, 3 contact hours (0;0;3).

Prerequisites: CS 490. Restriction: Senior standing in the Honors College and project proposal approval. A course similar to CS 491, with a project of greater depth and scope.

CS 100. Roadmap to Computing. 3 credits, 3 contact hours (3;0;0).

An introduction to programming and problem solving skills using Python or other very high level language. Topics include basic strategies for problem solving, constructs that control the flow of execution of a program and the use of high level data types such as lists, strings and dictionaries in problem representation. The course also presents an overview of selected topics in computing, such as networking and databases.

CS 101. Computer Programming and Problem Solving. 3 credits, 3 contact hours (3;0;0).

An introductory course that is designed for engineering freshman. This course introduces students to the engineering problem solving process in the context of MATLAB. The emphasis is on the logical analysis of a problem and the formulation of a computer program leading to its solution. Topics include basic concepts of computer systems, algorithm design, programming languages and data abstraction. At the end of class, a comparison between MATLAB and C/C++ will be discussed in order to provide students a better understand of general concept of computer programming.

CS 103. Computer Science with Business Problems. 3 credits, 3 contact hours (3;0;0).

An introductory course in computer science, with applications to business and managerial decision making. Topics include basic concepts of computer systems, software engineering, algorithm design, programming languages and abstraction, with applications.

CS 104. Computer Programming and Graphics Problems. 3 credits, 3 contact hours (3;0;0).

An introductory course in computer science with applications in computer graphics for architecture. Emphasis on programming methodology using a high level language as the vehicle to illustrate the concepts. Topics include basic concepts of computer systems, software engineering, algorithm design, programming languages and data abstraction, with applications.

CS 105J. Computer Prog-Java. 1 credit, 1 contact hour (1;0;0).**CS 106. Roadmap to Computing Engineers. 3 credits, 3 contact hours (3;0;0).**

An introduction to programming and problem solving skills for engineering majors using Python programming languages. Topics include basic strategies for problem solving, constructs that control the flow execution of a program and the use of high level data types such as lists, strings, and dictionaries in problem representation. The course also presents an overview of selected "big idea" topics in computing.

CS 113. Introduction to Computer Science. 3 credits, 3 contact hours (3;0;0).

Prerequisites: CS 100 with a grade C or better. Intensive introduction to computer science. Problem solving decomposition. Writing, debugging, and analyzing computer programs. Introduction to arrays and lists. Iteration and recursion. The Java language is introduced and used to highlight these concepts. A student receiving degree credit for CS 113 cannot receive degree credit for CS 115.

CS 114. Introduction to Computer Science II. 3 credits, 3 contact hours (3;0;0).

Prerequisites: CS 113 or completion of a required 100 level GUR course in CIS, plus an approved CIS 105. A study of advanced programming topics with logical structures of data, their physical representation, design and analysis of computer algorithms operating on the structures, and techniques for program development and debugging. Course covers program specifications, correctness and efficiency, data abstraction, basic aspects of simple data structures, internal searching and sorting, recursion and string processing. Algorithmic analysis is also discussed. Students receiving degree credit for CS 114 cannot receive degree credit for CIS 335 or CIS 505.

CS 115. Intro. to CS I in C++. 3 credits, 3 contact hours (3;0;0).

Fundamentals of computer science are introduced, with emphasis on programming methodology and problem solving. Topics include basic concepts of computer systems, software engineering, algorithm design, programming languages and data abstraction, with applications. The high level language C++ is fully discussed and serves as the vehicle to illustrate many of the concepts. CIS majors should enroll in CS 113.

CS 116. Intro. to Computer Science II/C++. 3 credits, 3 contact hours (3;0;0).

Prerequisites: CS 115 or completion of a required 100 level GUR course in CS, plus an approved CS 105. A study of advanced programming topics with logical structures of data, their physical representation, design and analysis of computer algorithms operating on the structures, and techniques for program development and debugging. Course covers program specifications, correctness and efficiency, data abstraction, basic aspects of simple data structures, internal searching and sorting, recursion and string processing. Algorithmic analysis is also discussed. Students receiving degree credit for CS 116 cannot receive degree credit for CS 505.

CS 241. Foundations of Computer Science I. 3 credits, 3 contact hours (3;0;0).

Prerequisites: (CS 114 AND MATH 112) OR (CS 114 AND MATH 133). An introduction to the foundations of computer science with emphasis on the development of techniques for the design and proof of correctness of algorithms and the analysis of their computational complexity. Reasoning techniques based on propositional and predicate logic and relational calculus operations with applications to databases will also be introduced. Auxiliary topics such as combinatorics of finite sets, functions and relations, and graph-theory definitions and graph storage alternatives will also be examined.

CS 252. Computer Organization and Architecture. 3 credits, 3 contact hours (3;0;0).

Prerequisite: CS 113. An introduction to the organization and architecture of computer systems, including the standard Von Neumann model and more recent architectural concepts. Among the topics covered are numeric data representation, assembly language organization, memory addressing, memory systems, both real and virtual, coding and compression, input/output structures treated as programmed, interrupt, and direct memory access, and functional organization of the CPU and the computer system.

CS 266. Game Modification Development. 3 credits, 3 contact hours (3;0;0).

Prerequisites: IT 102 OR IT 114 OR CS 116, OR CS 114. This course introduces students to the basic concepts of game programming and development. Students will learn how to reprogram a professional game engine, or Modification (Mod) development as it is referred to in the industry. Students will work with C extensively. Students will work on their own game projects utilizing the professional game engine.

CS 276. 2D Game Development. 3 credits, 3 contact hours (3;0;0).

Prerequisites: CS 265 and CS 266 or IT 265 and IT 266. This course introduces students to the core concepts and skills necessary for the development of games utilizing 2D graphics. Students will learn how to set up and program their own 2D graphics based game engine. The engine will integrate 2D graphics, audio, input handling and network socket programming. Students will learn how to utilize their own custom 2D graphics and sounds into their projects. Once complete, students will have created two fully functional games.

CS 280. Programming Language Concepts. 3 credits, 3 contact hours (3;0;0).

Prerequisite: CS 114 OR CS 116 OR IT 114 OR IT 102. Conceptual study of programming language syntax, semantics and implementation. Course covers language definition structure, data types and structures, control structures and data flow, run-time consideration, and interpretative languages.

CS 288. Intensive Programming in Linux. 3 credits, 3 contact hours (3;0;0).

Prerequisite: CS 114. The course covers Linux programming with Apache Web and MySql database using Php/Python and C as primary languages. It consists of four stages: basic tools such as Bash and C programming; searching trees and matrix computing, end-to-end applications such as one that constantly presents top 100 stocks; and extending the applications to run on multiple machines. The course provides students with hands-on experience for programming relatively large applications.

CS 310. Co-op Work Experience I. 3 credits, 3 contact hours (0;0;3).

Restriction: completion of the sophomore year, approval of the department, and permission of the Office of Cooperative Education and Internships. Students gain major-related work experience and reinforcement of their academic program. Work assignments facilitated and approved by the Co-op office. Mandatory participation in seminars and completion of a report. Note: Normal grading applies to this COOP Experience.

CS 332. Principles of Operating Systems. 3 credits, 3 contact hours (3;0;0).

Prerequisite: CS 114 OR CS 116 OR IT 114 OR IT 102. Organization of operating systems covering structure, process management and scheduling; interaction of concurrent processes; interrupts; I/O, device handling; memory and virtual memory management and file management.

CS 333. Introduction to UNIX Operating Systems. 3 credits, 3 contact hours (3;0;0).

Prerequisite: CS 332 or equivalent and knowledge of C language. The course covers the UNIX system kernel including initialization, scheduling, context switching, process management, memory management, device management, and the file system. The course also includes the organization of shells, editors, utilities, and programming tools of the UNIX operating system.

CS 337. Performance Modeling in Computing. 3 credits, 3 contact hours (3;0;0).

Prerequisites: CS 114 and (MATH 333 or MATH 341). Introduction to probability models and techniques useful in computer science. Performance evaluation, discrete-event simulation, classification and optimization.

CS 341. Foundations of Computer Science II. 3 credits, 3 contact hours (3;0;0).

Prerequisites: (CS 241 or MATH 226) and CS 280. This course provides an introduction to automata theory, computability theory, and complexity theory. Theoretical models such as finite state machines, push-down stack machines, and Turing machines are developed and related to issues in programming language theory. Also, the course covers undecidability and complexity theory, including the classes P and NP.

CS 345. Web Search. 3 credits, 3 contact hours (3;0;0).

Prerequisites: CS 280 and (CS 241 or CS 252). An introductory course on the web searching. The architecture of a search engine. Information vs. data retrieval. Web crawling. Processing text(tokenization, stemming, stopwords, link analysis). The indexing process and inverted indexes. Query processing. Ranking algorithms based on indexes and links (e.g. Kleinberg's HITS, Google's PAGERANK). Retrieval Models. Search engine evaluation. Case studies (e.g. Google cluster architecture).

CS 356. Introduction to Computer Networks. 3 credits, 3 contact hours (3;0;0).

Computer Science students should take CS 288 before taking CS 356. This course provides an introduction to computer networks, with a special focus on the Internet architecture and protocols. Topics include layered network architectures, addressing, naming, forwarding, routing, communication reliability, the client-server model, web and email protocols. Besides the theoretical foundations, students acquire practical experience by programming reduced versions of real Internet protocols.

CS 357. Fundamentals of Network Security. 3 credits, 3 contact hours (3;0;0).

Prerequisite: CS 356 or IT 120. This course is designed for Computer Science and Information Technology students. They must have a networking course before taking CS 357. IT students take IT 120 and Computer Science students take CS 356. This course offers an in depth study of network security issues, types of computer and network attacks, and effective defenses. It provides both a theoretical foundation in the area of security and hands-on experience with various attack tools, firewalls, and intrusion detection systems. Topics include: network scanning, TCP/IP stack fingerprinting, system vulnerability analysis, buffer overflows, password cracking, session hijacking, denial of service attacks, intrusion detection.

CS 366. 3D Game Development. 3 credits, 3 contact hours (3;0;0).

This course introduces students to the core concepts and skills necessary for the development of games utilizing 3D graphics. Students will learn how to set up and program their own 3D graphics based game engine using OpenGL. Students will learn how to load and display custom 3D models created using existing 3D modeling tools. Once complete, students will have created two fully functional 3D games and tools to work with them.

CS 370. Introduction to Artificial Intelligence. 3 credits, 4 contact hours (3;1;0).

Prerequisites: CS 114 and (MATH 226 or CS 241). An exploration of concepts, approaches and techniques of artificial intelligence. Emphasizes both underlying theory and applications. Topics include knowledge representation, parsing language, search, logic, abduction, uncertainty, and learning. LISP and Prolog programming languages used extensively. Students are required to do programming assignments, complete a programming term project and review case studies.

CS 388. Android Application Developmnt. 3 credits, 3 contact hours (3;0;0).

This course introduces mobile application development for the Android platform. Students will learn skills necessary for creating and deploying applications with the Android Software Development Kit (SDK). The course is designed to introduce and familiarize students with programming in the Android environment. It starts with an examination of the basic components and concepts that define the Android platform, and then moves on to cover the specific structure that comprises an Android application. An overview of the most common tools and techniques for writing Android applications is included. The Android approach to user interfaces is described along with a discussion of some of the more common user interface elements. Storage strategies for persistent information are also covered, including the use of the available SQLite Database features. The unique characteristics of programming for a mobile environment are introduced and explained. Hands on experience in the form of exercises and programming projects are included throughout the course to reinforce material that has been presented in lecture form.

CS 408. Cryptography and Internet Security. 3 credits, 3 contact hours (3;0;0).

Prerequisite: MATH 226 or CS 241. Covers security require-ments for telecommunication over the Internet and other communication networks, various conventional and public-key encryption protocols, digital encryption standard, RSA and ElGamal cryptographic systems, digital signature algorithm and analysis of its cryptoimmunity, and access sharing schemes. Students receiving credit for CS 408 may not enroll in CIS 608.

CS 410. Co-op Work Experience II. 3 credits, 3 contact hours (0;0;3).

Prerequisites: CS 310 or its equivalent, approval of the department, and permission of the Office of Cooperative Education and Internships. Provides major-related work experience as co-op/internship. Mandatory participation in seminars and completion of requirements that include a report and/or project. Note: Normal grading applies to this COOP Experience.

CS 431. Database System Design and Management. 3 credits, 3 contact hours (3;0;0).

Prerequisite: CS 114 or equivalent. Database system architecture; data modeling using the entity-relationship model; storage of databases; the hierarchical, network and relational data models; formal and commercial query languages; functional dependencies and normalization for relational database design; relation decomposition; concurrency control and transactions management. Student projects involve the use of a DBMS package.

CS 433. Introduction to Linux Kernel Programming. 3 credits, 3 contact hours (3;0;0).

An introductory study of how the Linux operating system is built from scratch. AS a hands-on course, students will perform intensive programming using Linux kernel. The contents include booting, segmentation and paging, creating and destroying processes, process switching and scheduling, handling exceptions and interrupts, software interrupts, creating system calls, creating file systems, networking with TCP/IP, device driver writing and module programming, etc. At the end of the course, students will be able to modify the Linux operating system to create their own.

CS 434. Advanced Database Systems. 3 credits, 3 contact hours (3;0;0).

Prerequisites: CS 431. The course covers the basic concepts of traditional files and file processing, provides a "classic" introduction to the relational data model and its languages, and discusses database design methodology and application developments. Students are expected to learn the design of database application systems through a small project and to get some practical hands-on experience with commercial database management systems (DBMS) by writing application programs using the commercial DBMS query languages.

CS 435. Advanced Data Structures and Algorithm Design. 3 credits, 4 contact hours (3;1;0).

Prerequisite: CS 241 and CS 288. Advanced topics in data structures and algorithms, involving sequences, sets, and graphs such as searching, sorting, order statistics, balanced search tree operations, hash tables, graph traversals, graph connectivity and path problems. Algebraic and numeric algorithms. Performance measures, analysis techniques, and complexity of such algorithms.

CS 438. Interactive Computer Graphics. 3 credits, 3 contact hours (3;0;0).

Prerequisites: completion of a 100-level course in CIS, plus knowledge of a higher level language. This course introduces fundamental concepts of interactive graphics oriented toward computer-aided design systems. Such systems emerge in engineering, architecture, and manufacturing. Topics include computer data structures for representation of two- and three-dimensional objects and algorithms for definition, modification, and display of these objects in applications. This course will also discuss a selection of special topics in interactive graphics.

CS 439. Image Processing and Analysis. 3 credits, 3 contact hours (3;0;0).

Prerequisites: CS 114 and MATH 333. This course is an intensive study of the fundamentals of image processing, analysis and understanding. Topics to be covered include: a brief review of the necessary mathematical tools, human visual perception, sampling and quantization, image transformation, enhancement, restoration, compression, reconstruction, image geometric transformation, matching, segmentation, feature extraction, representation and description, recognition and interpretation.

CS 440. Computer Vision. 3 credits, 3 contact hours (3;0;0).

Prerequisite: MATH 333. This course introduces basic concepts and methodologies of computer vision, and focuses on material that is fundamental and has a broad scope of applications. Topics include contemporary developments in all mainstream areas of computer vision e.g., Image Formation, Feature Representation, Classification and Recognition, Motion Analysis, Camera Calibration, Stereo Vision, Shape From X (shading, texture, motion, etc.), and typical applications such as Biometrics.

CS 441. Database Programming. 3 credits, 3 contact hours (3;0;0).

Many technologies have been developed due to the interplay between World-Wide Web Development and Databases on one hand and the growth of Database applications in E-Commerce on the other hand. Today, practically every E-Commerce application has at least a Web component and a Database Component. Many languages have been developed in order to deal with these interactions. The proposed course will focus on accessing databases through the web but also mention new developments in the field.

CS 458. Technologies-Network Security. 3 credits, 3 contact hours (3;0;0).

Prerequisite: CS 356 or CS 456 or IT 420. This course provides both an in depth theoretical study and a practical exposure to technologies which are critical in providing secure communication over the Internet. Topics include: remote access security, web security, wireless security, e-mail security, spam and spam filtering techniques, computer viruses and internet worms, honeypots and honeynets, security liability issues and compliance.

CS 482. Data Mining. 3 credits, 3 contact hours (3;0;0).

Prerequisite: CS 431. The course covers the concepts and principles of advanced data mining systems design; presents methods for association and dependency analysis, classification; prediction; and clustering analysis.

CS 485. Special Topics in Computer Science/Information Systems. 3 credits, 3 contact hours (3;0;0).

Restriction: junior standing and/or department approval. The study of new and/or advanced topics in an area of computer science not regularly covered in any other CIS course. The precise topics to be covered in the course, along with prerequisites, will be announced in the semester prior to the offering of the course. A student may register for no more than two semesters of Special Topics.

CS 486. Topics in Computer Science/Information Systems. 3 credits, 3 contact hours (3;0;0).

Restriction: junior standing and/or department approval. A continuation of CS 485.

CS 488. Independent Study in Computer Science. 3 credits, 0 contact hours (0;0;0).

Restriction: open only to students in the Honors Program who are computer science majors and who have the prior approval of the department and the CIS faculty member who will guide the independent study. Independent studies, investigations, research, and reports on advanced topics in computer science. Students must prepare, in collaboration with their faculty mentor and in the semester prior to enrolling in this course, a detailed plan of topics and expected accomplishments for their independent study. This must have the approval of both the department and the faculty mentor. A student may register for no more than one semester of Independent Study.

CS 490. Guided Design in Software Engineering. 3 credits, 3 contact hours (3;0;0).

Prerequisites: CS 280 and CS 288. This course focuses on the methodology for developing software systems. Methods and techniques for functional requirements analysis and specifications, design, coding, testing and proving, integration and maintenance are discussed.

CS 491. Senior Project. 3 credits, 3 contact hours (3;0;0).

Prerequisites: CS 490, senior standing and project proposal approval. An opportunity for the student to integrate the knowledge and skills gained in previous computer science work into a team-based project. The project involves investigation of current literature as well as computer implementation of either a part of a large program or the whole of a small system.