Computer Science

**Chair:** Professor V. Barr

**Faculty:** Professor J. Spinelli; Associate Professors A. Cass, C. Fernandes; Assistant Professors J. Rieffel, K. Striegnitz; Visiting Assistant Professor N. Webb

**Administration:** L. Spallholz (Lab Manager)

**Staff:** L. Bremigen (Administrative Assistant)

The department offers a B. S. in computer science, and collaborates with the Electrical and Computer Engineering Department to offer a B. S. in computer engineering, and with the Visual Arts department to offer a program in digital art.

The department also participates in offering three minors: a traditional minor in computer science (described below), a minor in Computational Methods, and a minor in Digital Media. The latter two are described in their own entries in this catalog.

**Requirements for the Major in Computer Science:** Ten computer science courses including one 100-level introductory course, 150, 250, 260, 270, five electives numbered 110 or higher; plus the project sequence 497–499. Of the five electives, four must be at least 300-level, with one from the Theory group and one from the Systems group. Also required: Math 197; Math 110-112 or 113; one Math elective numbered above 113, chosen in consultation with the advisor; one major-level lab science course from Biology, Chemistry, Geology, Physics, Environmental Science, or ESC-100, and one non-CS course meeting the Core Curriculum science and technology requirement (courses cross-listed with CSC are not acceptable). A typical first year major program includes a 100-level introductory course and CSC-150, Math 197, and first year Core Curriculum courses.

The Theory group: CSC-350 Theory of Computing; CSC-370 Programming Languages.

The Systems group: CSC-333 Introduction to Parallel Computing, CSC-335 Operating Systems, CSC-483 Topic: Compilers

The senior writing requirement is satisfied by CSC 498 and CSC 499.

**Requirements for the Major in Computer Engineering:** Refer to the Computer Engineering section.

**Requirements for the Interdepartmental Major:** the following eight courses in the department: a 100-level introductory course, CSC 150, 250, 260, 497, one course above introductory level (above 110), one course of at least 300-level, and one other CSC course. Only one CSC course numbered below 100 may be included, and only one 100-level introductory CSC course may be included. Also required: MTH-197 and the project sequence IDM-498 and IDM-499. The ID senior project must be designed to integrate the fields composing the major.

**Requirements for Honors in Computer Science:** Candidates for honors in computer science must have a minimum overall grade point average of 3.3, a minimum grade point average of 3.3 in the major with at least 3 grades of A- or better in full credit CS courses numbered 100 or above, a grade of at least A- for CSC-499 or IDM-499, and must present the senior project at the Steinmetz Symposium.

**Requirements for the Minor in Computer Science:** Six computer science courses including a 100-level introductory course, CSC-150 and four additional CSC courses chosen with the approval of an advisor from computer
Course Selection Guidelines

Placement: A score of 4 or 5 on the AP Computer Science “A” exam will count as having satisfied one of the introductory courses (CSC 103-109).

Prerequisite: For prerequisite structure of all courses in the CS major, please visit cs.union.edu/media/csmajorgraph.pdf

General Interest Courses

CSC-050. Computers and Computing (not offered, 2011-2012). Introduction to spreadsheet and database applications, computer hardware and programming. Not open to computer science or engineering majors, or to students with credit for a computer science course numbered 100 or higher.

CSC-055. Working with the Web (Fall, Fernandes). Design, writing, and publishing of WWW pages; creation of graphical images; study of the underlying Web technologies such as communication protocols, digital encoding and compression; programming of Web pages.

CSC-080. History of Computing (Cross-listed as HST 292) (Fall, Webb). A survey of tools for computation, from number systems and the abacus to contemporary digital computers. The course focuses on the development of modern electronic computers from ENIAC to the present. Study of hardware, software, and the societal effects of computing.

Introductory Courses

Each CS major or minor program includes one course from the following list. Each course focuses on a distinct application area. The courses all cover the same basic computer science concepts and programming skills and only one may be counted toward a major or minor. These courses are open to non-majors and are prerequisite to certain intermediate courses that are also available to and suitable for non-majors. A grade of C- or better is required in order to take any course that requires an introductory course as prerequisite.

CSC-103. Taming Big Data: Introduction to Computer Science (Fall, Barr). Introduction to the field of computer science with the theme of natural and social science applications. Introduces students to algorithms, basic data structures, and programming techniques. Includes development of programs and use of existing applications and tools for computational applications including simulation, data analysis, visualization, and other computational experiments. Includes a laboratory.

CSC-104. Robots Rule! Introduction to Computer Science (Fall, Webb). Introduction to the field of computer science with a robotics theme. Introduces students to algorithms, basic data structures, and programming techniques. Students will build and program robots, exploring mobility, navigation, sensing, and inter-robot communication. Additional class topics include: history of robotics, social and ethical issues, emotionally intelligent behavior and other current topics in robotics. Includes a laboratory.

CSC-105. Game Development: Introduction to Computer Science (Spring, Striegnitz). Introduction to the field of computer science with a computer games theme. Introduces students to algorithms, basic data structures, and
programming techniques. Computer game development is used as an example application area and students implement their own games throughout the course. Includes a laboratory.

**CSC-106. Can Computers Think? Introduction to Computer Science** (Winter, Fernandes; Spring, Fernandes). Introduction to the field of computer science with an artificial intelligence theme. Introduces algorithms, basic data structures, programming techniques, and basic methods from artificial intelligence. Includes discussion of questions in the philosophy of artificial intelligence. Includes a laboratory.

**CSC-107. Creative Computing: Introduction to Computer Science** (Winter, Webb). Introduction to the field of computer science with a media computation theme. Introduces students to algorithms, basic data structures, and programming techniques. Media computation is used as an application area, focusing on image manipulation, sound splicing, animations, HTML generation and automated reading of web pages. Includes a laboratory.

**CSC-109. (070) Computer Programming for Engineers** (Fall, Rieffel; Winter, Almstead; Spring, Webb). Introduction to the field of computer science with an engineering applications theme. Topics include math and logical operations, data types, matrices, conditions and decisions, looping, subroutines, numerical methods, and plotting.

**Intermediate and Advanced Courses**

**CSC-118. Introduction to Computer and Logic Design** (Fall, Cotter). See ECE-118.

**CSC-150. Data Structures** (Winter, Cass; Spring, Fernandes). Basic concepts of data organization and abstraction, software design, stacks, queues, trees, and their implementation with linked structures. Programming in Java. Prerequisites: one course from CSC-103 to CSC-109. A grade of C- or better is required in order to continue with any course that requires CSC-150 as a prerequisite.

**CSC-206. Natural Language Processing** (Spring, Striegnitz). This course studies computational techniques for processing human languages. It will introduce data structures and algorithms for various natural language processing tasks and applications, presenting statistically motivated as well as linguistically and psycholinguistically motivated methods. Prerequisite: one course from CSC-103 to CSC-109.


**CSC-237. Data Communications and Networks** (Fall, Spinelli). (See ECE-337).

**CSC-240. Web Programming** (to be offered 2012-2013). This course addresses the standards in programming applications for the Web. Covers the client-side technologies XHTML, CSS, JavaScript and Dynamic HTML as well as server-side technologies PHP, MySQL, and CGI programming in Perl or Python. Prerequisite: one course from CSC-103 to CSC-109.

**CSC-243 (283). Bioinformatics: Information Technology in the Life Sciences.** (Cross-listed as BIO 243) (to be offered 2012-2013). Biology and computer science students will gain a working knowledge of the basic principles of the others’ discipline, and will collaborate together on bioinformatics projects. Topics include pairwise and multiple sequence alignments, phylogenetic trees, gene expression analysis, and protein structure prediction. Additional topics will be presented by invited speakers. Prerequisites: BIO-225 or one course from CSC-103 to CSC-109.

**CSC-245. The Computer Science of Computer Games** (Fall, Striegnitz). This course surveys the field of computer science from the perspective of computer games. Topics explored include: rendering of graphics to a screen, implementation of realistic simulation, use of artificial intelligence in games, handling user input, game physics, collaborative development. Final course project is a complete computer game. Prerequisite: one course from CSC-103 to CSC-109.

**CSC-250. Algorithm Design and Analysis** (Spring, Cass). Fundamental algorithms used in a variety of applications. Includes algorithms on list processing, string processing, geometric algorithms, and graph algorithms.
Prerequisites: CSC-150 and MTH-197, or permission of the instructor. A grade of C- or better is required in order to continue with any course that requires CSC-250 as a prerequisite.

**CSC-260. Large-Scale Software Development** (Winter, Cass). Strategies for the systematic design, implementation, and testing of large software systems. Design notations, tools, and techniques. Design patterns and implementation idioms. Implementation, debugging, and testing. Includes team and individual software development projects. Prerequisites: CSC-150 and MTH-197, or permission of the instructor.

**CSC-270. Computer Organization** (Winter; Rieffel). The architecture and operation of the digital computer. CPU design, input/output, computer arithmetic, assembly language. Prerequisite: CSC-150. Includes a laboratory.

**CSC-280. User Interfaces** (to be offered 2012-2013). Introduction to the field of human-computer interaction (HCI) through the study of user interfaces. Theory and application of what makes an interface usable. Design principles, empirical studies, and statistical analyses will be employed in team-based projects. Students will make extensive use of equipment for recording and analyzing participants in both laboratory and field settings. Prerequisite: CSC-150.


**CSC-320. Artificial Intelligence** (Fall, Rieffel). Fundamental concepts used in creating “intelligent” computer systems; semantic representation, logical deduction, natural language processing, and game playing; expert systems, knowledge-based systems, and elementary robotics. Prerequisite: CSC-250.

**CSC-325. Robotics** (to be offered 2012-2013). The course will cover basic algorithms necessary for motor control. Building on these methods we will discuss higher level navigation for mobile robots, as well as the sensing necessary for localization of the robot in its environment. Finally we will also examine the challenges of motion planning for jointed robots with many degrees of freedom. Prerequisite: CSC-250 or permission of the instructor.

**CSC-329. Neural Networks** (not offered 2011-2012). (See ECE-329).

**CSC-333. Introduction to Parallel Computing** (Spring, Rieffel). Synchronization and communication in concurrent programs. Parallel computing with libraries for shared-memory programming and for cluster computing. Introduction to algorithms for parallel scientific computing. Prerequisite: CSC-250.

**CSC-335. Operating Systems** (to be offered 2012-2013). Selected topics in operating system development including process and thread management, concurrency, memory and file system management, resource allocation, job scheduling, and security. Prerequisites: CSC-270 and junior standing.

**CSC-340. Introduction to Databases** (to be offered 2012-2013). Introduction to data models and database design. Coverage of network, hierarchical, and relational architectures with emphasis on the latter. Study of relational algebra, entity-relationship modeling, and data normalization. Study of fourth generation query languages including SQL. Introduction to centralized, distributed, federated, and mediated systems. Prerequisite: CSC-150 and MTH-197.


**CSC-360. Software Engineering** (Fall, Cass). Strategies for the specification, design, production, testing, and support of computer programs; software development models; programming team structures; documentation; and maintenance. Prerequisite: CSC-260.

**CSC-370. Programming Languages** (to be offered 2012-2013). An introduction to issues in programming language design and implementation. Major programming language paradigms: functional, logic, and object-oriented, and their use. Prerequisite: CSC-150 and junior standing.

CSC-483. Selected Topics in Computer Science (Spring, Webb).
CSC-490, 491, 492. Independent Study (Fall, Winter, Spring: Staff). Prerequisite: Permission of department chair.
CSC-497. Computer Science Capstone Seminar (Spring; 0.5 credit, Rieffel) Development of the skills necessary for independent research: Reading scholarly works, designing experiments and empirically evaluating their results. Development of a comprehensive senior capstone project proposal. Investigation of professional ethics, skills and responsibilities. Prerequisite: CSC-260. Normally taken in spring of the junior year.
CSC-498, 499. Computer Science Capstone Project (Fall, Winter, Spring: .75 credit, Staff). Design, implementation, and evaluation of the capstone project. Normally taken during the senior year.