The Undergraduate Program in Computer Science

Computer Science Department
University of Rochester
Rochester, NY  14627-0226
(585) 275-4505
http://www.cs.rochester.edu

(Revised 10/1/15)

Contents

1. Introduction
2. Computer Science Advisers
3. Introductory Courses
4. Curriculum for B.S. Degree in Computer Science
   4.1 Admission to the B.S. Program
   4.2 Core Course Requirements for the B.S. Degree
   4.3 Advanced Course Requirements for the B.S. Degree
   4.4 Honors Research in Computer Science
   4.5 Sample B.S. Programs
5. Curriculum for the B.A. Degree
   5.1 Pre-Approved Tracks
   5.2 Creating New Tracks
   5.3. Getting Started in the BA
6. Upper-Level Writing Requirement for Computer Science Majors
7. Computer Science Minor
8. Clusters in Computer Science
9. Opportunities
   9.1 KEY Scholars
   9.2 BS/MS Program
   9.3 GEAR (Graduate Engineering At Rochester)
   9.4 Industry Practicum
   9.5 Departmental Distinction
   9.6 Study Abroad
10. Course Descriptions
1. Introduction

The Department of Computer Science at the University of Rochester was established in 1974 as a graduate research institution. In 1995, the Department began its undergraduate program, offering Bachelor of Science, Bachelor of Arts, and Minor degrees. It participates in the University’s BS/MS program, through which a qualifying student in the B.S. program can also earn a Master of Science degree by one additional year of study.

The B.S. curriculum provides a rigorous background in all core areas of computer science. It is appropriate for students who aspire to achieve a high-level research and development position in the computer industry, who plan to go on to earn an M.S. or PhD. in computer science, or who simply wish to have the broadest and deepest knowledge of the field.

The B.A. curriculum is highly flexible, and can be customized to support students interested in the intersection of computer science with other disciplines, such as mathematics, digital media studies, financial economics or linguistics. The B.A. is good preparation for students aiming for an industrial career or graduate study in a computing-related discipline. Because it has fewer requirements than the B.S., it is also a good option for students who wish to double-major in computer science and another subject, or who wish to specialize in a particular area of computer science.

The Minor in Computer Science can be earned by completing six courses above the 130 level, as described below.

The goal of all of our degree programs is to produce computer scientists, within the context of a liberal arts education. A computer scientist is one who is fluent in algorithmic thought and principles of the design and analysis of computer systems. Our curriculum introduces students to these key concepts and skills early on, and builds on that foundation in subsequent specialized courses. Our advanced courses cover topics such as computer architecture, robotics, operating systems, programming languages, complexity theory, human-computer interaction, artificial intelligence, machine learning, machine vision, and natural language processing. Our students gain the life-long learning skills necessary to stay current in and help shape the rapidly growing world of computing.

Many of our students become intimately involved in the Department's research program. Students find opportunities to work closely with faculty members and their research groups through the Undergraduate Problem Seminar (CSC 200), our Honors degree program, joint undergraduate-graduate courses, summer internships with faculty members for course credit or for pay, and independent study courses.

In 2009, the Department of Computer Science moved from the College of Arts and Sciences to the College of Engineering and Applied Science. While this change opened up new opportunities for our students, such as participation in the GEAR program, it has not diminished the department’s commitment to a liberal-arts education. As described below, our B.S. and B.A. degrees require two clusters, one each in humanities and social sciences. Three course clusters are minimum requirements. More than half of the computer science majors supersede cluster requirements by majoring or minoring in another discipline.

2.2. Computer Science Advisers

Each computer science student can take advantage of four levels of advising: their faculty adviser, the department undergraduate adviser, the Undergraduate Program Director, and the Hajim Dean’s office advising staff. All serve different needs of the students and work together to help the student to complete all necessary requirements and deal with any personal issues that might arise.
Any student choosing computer science as a major will receive a faculty adviser in computer science. Students and faculty advisers meet during Freshman/Transfer orientation and each subsequent semester to discuss course planning and selection. Long term issues about career choices, graduate school and day-to-day needs may also be discussed.

Additionally, the department has an undergraduate coordinator who is available to meet with students to keep them on track to graduation. Most paperwork related to academic life can be obtained from this office.

Students meet with their undergraduate adviser, the undergraduate coordinator or the undergraduate program director to discuss 4-year course plans, declaring majors and minors, drop/add forms, transferring credits, and independent study forms, as well as options related to study abroad, BS/MS program, internships, fellowships, teaching assistantships, cluster exceptions, KEY scholars, honors research, Take 5 applications, etc. The undergraduate program director must review and formally approve transfer courses and major declaration forms. For questions regarding university requirements that are beyond the scope of the department, students meet with advisers in the Hajim School of Engineering and Applied Sciences Dean’s office.

3. Introductory Courses

The department offers a number of introductory courses, where students can start exploring programming and computer science:

CSC 170 – Web Design and Development

This course is an introduction to Internet and Web technologies. Emphasis is placed on front-end web fundamentals, design concepts and industry standards without programming (Fall, Spring and Summer)

CSC 160 - Engineering Computing

This course provides an introduction to MATLAB programming, data collection hardware, and data analysis. Exercises are based on problems from a variety of engineering disciplines, including electrical, mechanical, and chemical engineering. Although this course was designed for freshman engineering majors, it is appropriate for students in any discipline interested in practical scientific programming and data analysis. (Spring)

CSC 161 - The Art of Programming

This course offers a self-contained introduction to the programming language Python, which can be used for practical problem solving in the sciences and humanities. Although Python is significantly easier to learn and use than Java, the concepts learned in this course also provide a good background for students going on to learn Java in CSC 171. This course is a good choice for students who intend to pursue a B.A. in Computer Science, or who are undecided about their major. (Fall and Spring)

CSC 171 - The Science of Programming

This course introduces students to Java, a powerful programming language used in most of our advanced courses. Students who know from the start that they plan to complete a B.S. in Computer Science usually begin with this course. (Fall and Spring)
CSC 172 - The Science of Data Structures

This course introduces notions of abstraction and modularity in programming. Students who have had a strong course in Java programming in high school usually begin with this course. This course is required for both the B.S. and B.A. degrees. (Fall and Spring)

The above courses are appropriate both for students intending to major in Computer Science and students intending to major in other disciplines. The chart below is designed to help choose an appropriate entrance point to our program, and shows the typical movement between courses. The dashed lines indicate that students who do well in CSC 161 or CSC 160 and are willing to learn the basics of the Java language on their own may skip CSC 171 and move directly on to CSC 172. Students interested in doing this must obtain permission from the undergraduate adviser as this is not advisable for everyone.

*Suggested entries and paths for introductory work in computer science.*

In addition to these introductory courses, other Computer Science courses that do not have prerequisites include:

- CSC 190 to 199 - Issues in Computing
- CSC 108 – Technical Computing
- CSC 131 - Recreational Graphics
4. Curriculum for B.S. Degree in Computer Science

The B.S. curriculum is described in terms of qualifying, core, and advanced courses. No more than two core or advanced courses can be completed at other institutions. If transfer courses are taken while in residence, they must be pre-approved by the computer science program undergraduate director.

4.1 Admission to the B.S. Program

Students typically declare a CSC major after successful completion of CSC 172 but all of these courses are required for the B.S.

- MTH 150 - Discrete Mathematics or MTH 150A Discrete Math Module
- MTH 161* - Calculus I
- MTH 162* - Calculus II
- CSC 171 - The Science of Programming
- CSC 172 - The Science of Data Structures

*Calculus sequences of MTH 141-MTH 143, or MTH 171Q & MTH 172Q are also acceptable.

A student must attain a grade of C- or higher in each of the above CSC courses and a GPA of no lower than 2.0 in these courses, which must not be taken on a pass/fail basis. Students typically complete these qualifying courses in the freshman or sophomore year.

4.2 Core Course Requirements for the B.S. Degree

To satisfy the requirements for the B.S., students must take the following core courses:

- CSC 173 - Computation and Formal Systems
- CSC 242 - Artificial Intelligence
- CSC 252 - Computer Organization
- CSC 254 - Programming Language Design & Implementation
- CSC 280 - Computer Models and Limitations
- CSC 282 - Design and Analysis of Efficient Algorithms
- MTH 165 - Linear Algebra with Differential Equations **
- CSC 262 – Introduction to Computational Statistics*

* This requirement can also be fulfilled with STT/MTH201, ECO 230 or STT213.

**Students who take MTH 173Q or BOTH MTH 163 and MTH 235 may count these options as covering the MTH 165 requirement.

Students in 2015-2018 will have the option of doing either CSC200 or the statistics requirement. Students completing the Honors Research degree will need to complete CSC 200H and CSC 262 (or substitute).

4.3 Advanced Course Requirements for the B.S. Degree

In addition to the core courses, the B.S. degree requires three additional advanced courses in computer science (courses numbered above 200, and not included in the core group above, and excluding CSC 390 - Supervised Teaching). Students should consult with their faculty adviser or the department undergraduate coordinator about their advanced course selection. Specialization is encouraged, though not mandatory. Though CSC200 is no longer a B.S. requirement, students interested in pursuing graduate study will benefit from the course. It helps prepare for participation in research and for senior-year independent work. In
particular, students are encouraged to select a related set of courses that will prepare them for participation in one of the department’s research projects and/or to complete a senior project. Areas of interest include the following:

- **Computer Systems**—CSC 253, CSC255, CSC 256, CSC 257, CSC 258, CSC 259
- **Natural Language and Knowledge Representation**—CSC 244, CSC 247, CSC 248
- **Machine Learning, Vision and Robotics**—CSC 240, CSC 246, CSC 249
- **Theory**—CSC 281, CSC283, CSC 284, CSC285, CSC 286, CSC 287
- **Human Computer Interaction and Web**—CSC 210, CSC 212
- **Data Science**—CSC240, CSC246, CSC259, CSC261, CSC265

In addition to the three advanced CSC courses, a B.S. candidate must complete *one* of the following:

- CSC 393 - Senior Project, OR

one additional advanced course in computer science (numbered 200 or higher, excluding CSC 390), OR mathematics (MTH 163,164,173,174, or any additional mathematics course numbered above 200). Especially appropriate are the mathematics courses in probability, linear programming and game theory, chaos and fractals, logic, number theory, cryptography, combinatorics, statistics and graph theory. Courses 200 level or above in other computing related disciplines (e.g., philosophy, linguistics, brain and cognitive science, statistics, electrical engineering, etc) will also be accepted.

### 4.4 Honors Research in Computer Science

The Computer Science Honors Research program is a version of the B.S. degree program in which honors-level coursework and a senior research thesis are required. The degree of honors awarded (“honors in research”, “high honors in research”, or “highest honors in research”) is determined by the Computer Science faculty in consultation with the student's honors committee. The quality of the thesis, its presentation, and other CSC research and publications throughout the student's undergraduate career are all considered. The curricular component of the research honors program consists of the courses CSC 200H, either CSC 391H or 393H, and CSC 395H. Each of these courses is 4.0 credit hours.

Steps to complete the program are as follows:

1. Complete CSC 200H, usually in the sophomore year, as an introduction to computer science research.

2. Find and consult with a Computer Science faculty research adviser. Formulate and refine a research topic. Recruit one other Computer Science professor as a second thesis committee member.

3. Submit a thesis proposal to the thesis committee, and obtain a signed honors thesis approval form after whatever revisions result from the committee's advice. Submit this form to the Undergraduate Coordinator. This phase should be completed by (and preferably before) the Fall semester of the senior year. NOTE: Students must still register for independent study (CSC 391H - Honors Research) or senior project (CSC 393H - Honors Senior Project), and complete coursework.

4. Once the proposal is approved and related relevant coursework (including CSC 391H or CSC 393H) is completed, enroll in CSC 395H (Honors Thesis) by completing an independent study form and submitting it to the registrar. Declare your intent to complete the honors curriculum by submitting the signed thesis approval form to the Undergraduate Coordinator in the Undergraduate
Program Office. (Under special circumstances, CSC 391H or CSC 393H may be taken in the same semester as CSC 395H.)

5. In CSC 395H, write or complete an honors thesis in the style of a scientific journal article. The written thesis must be given to the thesis committee no later than April 15th of the graduating year.

6. Present the thesis in a public seminar and successfully defend it in a private oral examination by (at least) the thesis committee and an additional faculty member chosen by the departmental chair.

7. Maintain a 3.3 GPA over the CSC B.S. concentration courses (the twelve courses the student has formally declared as the BS). Upper level writing courses are not included in this calculation though they appear on the concentration form.

4.5 Sample B.S. Programs

Following are three sample B.S. programs, emphasizing systems, theory, or AI in the advanced course selection (advanced courses are marked “AC” below). Note that these are only samples, and that other paths through the degree are possible.

Sample program with an advanced course emphasis in systems:

1st Year: Fall: CSC 171 (Pre), MTH 150 (Pre), 2 electives; Spring: CSC 172 (Pre), MTH 161 (Pre), 2 electives.

2nd Year: Fall: CSC 173 (Core), MTH 162 (Pre), 2 electives; Spring: CSC 252 (Core), MTH 165 (B.S.), 2 electives.

3rd Year: Fall: CSC 282 (Core), CSC262 (Core), CSC 254 (Core), 1 elective; Spring: CSC 280 (Core), CSC 242 (Core), 2 electives.

4th Year: Fall: CSC 257 (AC1), CSC 256 (AC2), 2 electives; Spring: CSC 258(AC3), CSC 393 (B.S.), 2 electives.

Sample program with an advanced course emphasis in theory:

1st Year: Fall: CSC 171 (Pre), MTH 150 (Pre), 2 electives; Spring: CSC 172 (Pre), MTH 161 (Pre), 2 electives.

2nd Year: Fall: CSC 173 (Pre), MTH 162 (Pre), 2 electives; Spring: CSC280 (Core), CSC 252 (Core), 2 electives.

3rd Year: Fall: CSC282 (Core), CSC 286 (AC1), CSC262 (Core), 1 elective; Spring: CSC 284 (AC2), 2 electives.

4th Year: Fall: CSC 254 (Core), CSC 281 (AC3), 2 electives; Spring: MTH 165 (B.S.), CSC 242 (Core), CSC 393 (B.S.), 1 elective.

Sample program with an advanced course emphasis in artificial intelligence:

1st Year: Fall: CSC 171 (Pre), MTH 150 (Pre), 2 electives;
Spring: CSC 172 (Pre), MTH 161 (Pre), 2 electives.

2nd Year: Fall: CSC 173 (Pre), MTH 162 (Pre), 2 electives;
Spring: CSC 242 (Core), CSC 252 (Core), 2 electives.

3rd Year: Fall: CSC 282 (Core), CSC262 (Core), 2 elective;
Spring: CSC 280 (Core), MTH 165 (B.S.) 2 electives.

4th Year: Fall: CSC 244 (AC1), CSC 254 (Core), 2 electives;
Spring: CSC 246 (AC2), CSC 249 (AC3), CSC 393 (B.S.), 1 elective.

5. Curriculum for the B.A. Degree

The B.A. curriculum is described in terms of tracks. A track is an approved set of at least three related advanced courses. Tracks allow students to focus their interests and to take advantage of the many other disciplines at UR (music, biology, digital media studies, studio art, political science, optics, brain and cognitive sciences, etc.) for which computing is a powerful enabler. In order to earn a B.A., a student must complete:

One approved track;
2 core courses as defined below
All of the prerequisite courses for the track; and
Additional courses as necessary in order to reach a total of 12 major courses. **

The set of 12 major courses includes all Computer Science courses taken by the student, subject to a limit of 2 independent study courses, and including up to 2 courses from other disciplines that appear in the student’s chosen track or track prerequisites. At least two of the twelve major courses taken for the degree must be CSC courses from the list “Core Course Requirements for the B.S. Degree” (currently 173, 242, 252, 254, 262, 280, and 282). No more than two of the major courses in Computer Science may be numbered lower than CSC172. All of the track courses and courses counting toward the 12 major courses must be taken for a grade and not on a pass/fail basis. All 12 will count toward their Computer Science GPA. No more than two of the 12 courses for the B.A. can be completed at other institutions. If transfer courses are taken while in residence, they must be pre-approved by the undergraduate program director.

There are no specific course requirements before entry into the program, but three CSC courses above the level of 130 must be passed with an average of at least 2.0 in order to enjoy the full privileges of a CSC major (non-expiring accounts, lab space, free printing, etc.). Students typically declare a CS major after successful completion of CSC172.

** No more than two non-CSC courses may count toward the 12 courses for the B.A., including prerequisites.

5.1 Pre-Approved Tracks

A student may chose one of the pre-approved tracks described in this section, or propose a new track for approval, as described in Sec 5.1. The prerequisites for each Computer Science track course are noted after the course name below. For prerequisites for non-Computer Science track courses, students should consult the appropriate department offering the course.
Artificial Intelligence and Machine Vision: (Choose three or more)

CSC 231 – Robot Control: MTH 165, CSC171 or ECE114
CSC 232 – Autonomous Mobile Robotics: MTH165, CSC171 or ECE114
CSC 244 - Logical Foundations of AI: CSC 173, CSC 242
CSC 246 – Machine Learning: CSC 242, MTH 165
CSC 249 – Machine Vision: CSC 242, MTH 161

Natural Language Understanding: (Choose three or more, including at least one CSC course)

CSC 244 - Logical Foundations of AI: CSC 173, CSC 242
CSC 246 – Machine Learning: CSC 242, MTH 165
CSC 247 - Natural Language Processing: CSC 242
CSC 248 - Statistical Speech and Language Processing: CSC 172, CSC 242
BCS 152 - Language and Psycholinguistics **
BCS 259 - Language Development **
BCS 261 - Language use and Understanding **
LIN 210 - Introduction to Language Sound Systems **
LIN 220 - Introduction to Grammatical Systems **

Human-Computer Interaction: (Choose three or more)

CSC 131 - Recreational Graphics
CSC 209 – Advanced Front End Web Design: CSC 170
CSC 210 - Web Programming: CSC 172
CSC 212 - Human-Computer Interaction: CSC 172
CSC 261 – Database Systems: Recommended: CSC173 and CSC 252
CSC 296 – Advanced Front-End Web Design

Theory of Computation: (Choose three or more)

CSC 281 – Cryptography (MTH150 or 162) and CSC 171
CSC 283 – Topics in Cryptography (CSC281 recommended)
CSC 284 - Advanced Algorithms: CSC 282
CSC 285 – Algorithms and Elections: (at least one of MTH 150, MTH 143, MTH 162, MTH 172,
PSC 107, CSC 280, CSC 281, CSC 282)
CSC 286 - Computational Complexity: CSC 280
CSC 287 - Advanced Modes of Computation: CSC 284 or CSC 286
MTH 248 – Graph Theory **

Computer Systems: (Choose three or more)

CSC 255 - Advanced Programming Systems: CSC 254
CSC 256 - Operating Systems: CSC 252
CSC 257 - Computer Networks: CSC 252
CSC 258 - Parallel and Distributed Systems: CSC 252, 254, 256
CSC259 – Big Data Computer Systems: CSC 252
ECE 201 - Advanced Computer Architecture **

Computer Security: (Choose three or more)
Computational Science: (Choose three or more, including at least one CSC course)

CSC 258 - Parallel and Distributed Computing: CSC 252, CSC 254, CSC 256
CSC 261 - Database Systems: Recommended: CSC 173 and CSC 252
BIO 266 - Tree of Life **
PHY 256 - Computational Physics **
OPT 211 - Computational Methods in Optics **
ME 211 - Computational Methods in Engineering **
BME 221 - Biomedical Computation **
CHE 242 - Introduction to Molecular Simulation **

5.2 Creating New Tracks

In consultation with a faculty member or the department’s undergraduate adviser, a student may design and propose a new track for approval by the undergraduate program director. The general requirements for a new track are that it reflects a coherent area of study and contains at least three advanced courses. The advanced courses that are in Computer Science must be outside our introductory courses (Sec. 2) and also outside our B.S. core courses (Sec. 3.2). Note that the prerequisites for the advanced courses will, in general, involve some of our core courses. Advanced courses for a new track from disciplines other than Computer Science should in general be at the 200 level or higher. Students together with the undergraduate department adviser may design and propose a new track for approval by the undergraduate program director.

5.3. Getting Started in the B.A.

Students begin the B.A. by taking an introductory sequence of Computer Science courses through CSC 172. Students should plan their B.A. for an existing or new track in consultation with the department’s undergraduate adviser. After choosing a track, make a list of all of its courses and prerequisite courses in Computer Science and other departments. Determine the semester you will take each required course during your time in the major. Finally, determine when you will have time to take any remaining major courses (needed to reach the total of 12 major courses) and electives. Note that the special topics courses, CSC 190-199 and CSC 290-299, may be taken for credit more than once if the topics of the offerings differ. New special topics offerings appear almost every semester.

6. Upper-Level Writing Requirement for Computer Science Majors

Every Computer Science major must develop, in consultation with his or her faculty adviser, a plan that includes two upper-level writing “experiences.” Each experience must generate at least 25 pages of expository prose, with substantial feedback on content and form, and revision of the work. (The 25 pages may be in the form of a single major paper or a series of smaller papers in a coherent context, e.g., a course.) The plan must be described in writing, on a form signed by both the student and the adviser. Acceptable writing experiences include (but at the adviser’s discretion are not limited to) the following:

- WRT 273 “Communicating Your Professional Identity” should be used to fulfill one of the two upper level writing requirements. This course should be taken in your junior year. (2 credits)
• Courses in other departments with a level of writing experience meeting the standard described above. Courses that other departments have labeled with a W as satisfying their departmental upper-level writing requirement are likely to be satisfactory, but must be checked for availability and may be vetoed by the adviser on the basis of insufficient or inappropriate content. Courses not labeled by other departments with a W may still be acceptable, but must be approved by the adviser as satisfying the Computer Science requirements.

• Designated courses in Computer Science. The Computer Science Department Curriculum Committee may designate certain courses as satisfying the upper-level writing requirement. All such courses must meet the standard described above; designation by the Computer Science Curriculum Committee will simply relieve the adviser of the need for verification. Students may also register for CSC 391W (Writing in Computer Science), which is a .5 credit course that can be added to other CSC courses, with permission of instructor. This will allow for separate grading of writing for any given course.

• Creation of a research paper. A student may, under the supervision of a faculty member, be an author or a principal co-author of a research paper submitted (with the supervisor’s approval) to a professional journal or conference, or published by the department as a technical report. Such reports of necessity require significant feedback and revision. Given the resulting level of quality, the adviser may waive the 25-page threshold.

For any writing experience other than a course, the student must file evidence of completion with the department’s undergraduate program administrator. The administrator will verify satisfaction of the upper-level writing requirement as part of the pre-graduation program review.

7. Computer Science Minor

The Minor requirements are satisfied by any six CSC courses above the level of 130 (except for CSC390 - Supervised Teaching). Here is a sample minor of six courses with emphasis on Web Design and Programming: (CSC 161 and 162) OR (CSC 171 and 172); CSC 170, CSC 209, CSC 210 & CSC 212.

8. Clusters in Computer Science

The Rochester curriculum includes clusters of three related courses in a discipline. Computer Science currently offers many such clusters. All courses in the college fall into the categories of Natural Science, Humanities, and Social Science. If your major is Computer Science (Natural Science), you will need a 3-course cluster in both the Humanities and Social Science areas. If you need a Natural Science cluster, the following CSC clusters will help you fulfill this requirement.

• Foundations of Computer Science (N4CSC001): CSC 171, CSC 172, CSC 173. This is the main course sequence leading into the Computer Science B.S. degree. It provides a thorough overview of foundational computer science techniques and issues. Prerequisite: none.

• Computer Systems (N4CSC004): Either CSC 171 or CSC 173, plus CSC 172 and CSC 252. Covers the internal organization of computers and its relation to recent computer hardware developments as well as to classical topics in computer software such as compilers and operating systems.

• Business Computing (N4CSC002): CSC 108 is required; Choose one: (MTH210 or ECO230); and choose one: (CSC 161, CSC 170 or CSC 171). An introduction to software packages, computing, and computerized business systems analysis. The mixture of programming skills and powerful analysis packages like EXCEL is a strong foundation for serious applications.
• Computer Science and Art (N4CSC009): Required: CSC108; choose one: (CSC131, CSC166, or CSC209), and also choose one: (CSC161, CSC170 or CSC171). This cluster introduces students to the use of computers in visual art and music.

• Computing for the Social Sciences (N4CSC010): Choose one: (CSC108, CSC199 – Social Implications of Computing), and choose one: (CSC161, CSC170, CSC171) and choose one (PSC200, PSC201, STT211, STT212, STT213, PSY 211, CSC 262 or ECO230). This cluster introduces students to powerful software packages and fundamentals of computer programming. There is a special emphasis on computation done in the context of the social sciences.

• Algorithms (N4CSC011): Choose one: (MTH150, MTH162 or MTH172Q) and choose one: (CSC161, CSC171, CSC 160) and choose one: (CSC 281, CSC 282). Emphasizes algorithmic thought, use of data structures and the design and analysis of efficient algorithms, including cryptography.

• Human Computer Interaction (N4CSC013): Required: Human Computer Interaction (CSC212): Choose either (CSC161 and CSC 162) OR (CSC 171 and CSC172). This cluster teaches the program and design tools for good user interfaces in modern technology.


• Web Design (N4CSC018): Required: CSC 170 & CSC 209. Choose either CSC 161 or CSC 171. This cluster will provide basic web design skills, teaching fundamentals of good web design.

• Web Development (N4CSC017): Required: CSC 210; Choose either (CSC161 & CSC 162) OR (CSC171 & CSC 172). This cluster gives students more depth into client side/back end web development, languages and current techniques.

9. Opportunities

9.1 KEY Scholars

KEY (Kauffman Entrepreneurial Year) Scholars: The Kauffman Entrepreneurial Year (KEY) Program offers selected students a fifth, tuition-free year of college to pursue entrepreneurial endeavors. Qualified UR students may propose to devote as much as an entire academic year to internships, special projects, business plan development, research into various facets of entrepreneurship, or analysis of how culture and public policy influence entrepreneurial activity. Students may apply from the time that they have been accepted into a major through the first semester of their senior year.

9.2 BS/MS Program

The BS/MS Program allows undergraduate students to complete both the Bachelors and Masters Degree in Computer Science in five years. The program assumes the completion of all undergraduate B.S. requirements, and possibly some of the graduate course requirements, within the first 4 years. The fifth year is devoted to graduate courses, research, and a final exam, thesis, or equivalent.

9.3 GEAR (Graduate Engineering at Rochester)
The GEAR program provides selected students with an assurance of admission into one of eight engineering master's programs at the Hajim School of Engineering and Applied Sciences: Biomedical Engineering, Chemical Engineering, Computer Science, Electrical and Computer Engineering, Materials Science, Mechanical Engineering, Optics, or Technical Entrepreneurship and Management (TEAM).

9.4 Industry Practicum

An elective industrial partnership program has been developed that allows students to spend six to eight months (usually a summer and an adjacent semester) working in an industrial setting allowing students to integrate education with paid experience before graduation. Graduation thus is delayed one semester. Interested students should meet with the department’s undergraduate adviser to plan their studies to ensure that all their academic program requirements are met despite the semester away.

9.5 Departmental Distinction

Departmental distinction in Computer Science, for both the B.A. and B.S. degrees, will be determined by the student’s GPA on the courses that constitute the program of study for the concentration. The minimum scores for the three levels of distinction will be 3.3 (Distinction), 3.5 (High Distinction), and 3.7 (Highest Distinction).

9.6 Study Abroad

Students are encouraged to plan to study abroad during their course of study. Making a four year course plan can identify which semesters would be optimum for this opportunity. The majority of students study abroad during their fifth, sixth or seventh semesters. Students are not allowed to study abroad in their final semester.

Many schools that teach courses in English have been identified by the college and have increased the number of destination schools for students without strong second language backgrounds. Computer Science has had students study in Europe, Asia, Oceania, and the Middle East. For more information about programs, please see the Center for Study Abroad, the Hajim adviser for SAB or your department’s undergraduate coordinator. Completing an individualized four-year course plan will help you in determining when and where you may be able to study.

Here are some sample programs:

**Sample BS program with study abroad in fall of junior year:**

<table>
<thead>
<tr>
<th>Year</th>
<th>Fall Courses</th>
<th>Spring Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>CSC 171 (Pre), MTH 150 (Pre), WRT 105, one cluster course</td>
<td>CSC 172 (Pre), MTH 161 (Pre), 2 cluster courses</td>
</tr>
<tr>
<td>2nd</td>
<td>CSC 173 (Core), MTH 162 (Pre) CSC212 (AC1), 1 cluster course</td>
<td>CSC 252 (Core), CSC 242 (Core), MTH 165 (BS), 1 cluster course</td>
</tr>
<tr>
<td>3rd</td>
<td>STUDY ABROAD</td>
<td>CSC 280 (core), CSC 261 (AC2), 1 cluster course, 1 elective</td>
</tr>
<tr>
<td>4th</td>
<td>CSC 282 (Core), CSC 254 (Core), CSC 262 or STT choice (BS), 1 elective</td>
<td>CSC 393 (BS), CSC 246 (AC3), 2 electives</td>
</tr>
</tbody>
</table>

**Sample BS program with study abroad spring of junior year**

<table>
<thead>
<tr>
<th>Year</th>
<th>Fall Courses</th>
<th>Spring Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>CSC 171 (Pre), MTH 150 (Pre), WRT 105, one cluster course</td>
<td>CSC 172 (Pre), MTH 161 (Pre), 2 cluster courses</td>
</tr>
</tbody>
</table>
2nd Year: Fall: CSC 173 (Core), MTH 162 (Pre) 2 cluster courses  
Spring: CSC 252 (Core), CSC 242 (Core), MTH 165 (BS), 1 cluster course
3rd Year: Fall: CSC 254 (Core), CSC 282 (Core), CSC 256 (AC1), 1 elective  
Spring: STUDY ABROAD
4th Year: Fall: CSC 262 (BS), CSC244 (AC2), 2 electives  
Spring: CSC 280 (Core), CSC 246 (AC3), CSC393 (BS)

Study abroad opportunities aren’t limited to school year or academic experiences only. Many students do programs in the summer, or find exciting internships abroad. All of these opportunities broaden your knowledge, enhance your career and enable personal growth.

10. Course Descriptions

CSC 108. Technical Literacy. An introduction to computer applications in business and graphic design. Students will begin by learning the basics and some advanced functions of Microsoft Word, Excel, and Powerpoint. The class then progresses through the Adobe graphic design applications Photoshop, After Effects, and Flash. In learning these applications, students are introduced to topics such as computer graphics, file compression, and animation. Not open to officially declared CSC majors. No prerequisites (4 hours; Fall/Spring).

CSC131. Recreational Graphics. A hands-on introduction to 3D computer graphics and animation techniques taught from a user point of view. Topics include 3D modeling, animation, and simulation. Assessment based on projects. No written exams. No previous programming or graphics experience required. (4 hours, Spring).

CSC160. Engineering Computing. This course provides an introduction to MATLAB programming, data collection hardware, and data analysis. Exercises are based on problems from a variety of engineering disciplines, including electrical, mechanical, and chemical engineering. Although this course was designed for freshman engineering majors, it is appropriate for students in any discipline interested in practical scientific programming and data analysis. (Spring)

CSC161. The Art of Programming. Organized thinking, creative problem solving, and the precise description of solutions are valuable skills in academia and life. The formulation and solution of problems using computers is increasingly important in all artistic and scholarly fields. We introduce core concepts and techniques of programming as a way to develop these skills, as basis for further CS study, and for application to other fields. Lab required. No prerequisites (Fall and Spring).

CSC 170. Web Design and Development. An introduction to Internet and web technologies. Topics include Internet transport protocols, HTML5 and CSS3, web page design and website publishing. Emphasis is placed on fundamentals, design concepts and industry standards. Additional topics include the user experience, mobile design issues, and copyright/intellectual property considerations. (4 hours; Fall, Spring)

CSC171. The Science of Programming. Discovering, formulating, and exploiting the structure of problems to aid in their solution by computer -- an introduction to algorithmic problem solving and computer programming in Java. Lab required. No prerequisites. (Fall, Spring)

CSC172. The Science of Data Structures. Abstract data types (e.g., sets, mappings, and graphs) and their implementation as concrete data structures in Java. Analysis of the running times of programs operating on such data structures, and basic techniques for program design, analysis, and proof of correctness (e.g., induction and recursion). Lab required. Prerequisite: CSC 171 or equivalent; MTH 150 is
CSC 173: Computation and Formal Systems. We investigate formal systems integral to computer science (including Turing machines and simpler automata, the Chomsky hierarchy of formal grammars, lambda calculus, propositional and predicate logic, logical circuits, and some practical programming languages, such as C, Scheme, Prolog, and SQL), their relations to each other (including the Church-Turing thesis), and some of their applications (in scanners, parsers, and data-base access, for example). Prereq.: CSC 172. (Fall)

CSC191-199. Issues in Computing. Rotating topics in computer science that do not require prior computing experience. This course may be repeated for credit for different topics. Prerequisite: none (Fall and Spring;) Courses have been separated by broad subject area and will be assigned individual course titles when offered:

- 191: Issues in Cognitive Science
- 192: Issues in Programming Languages
- 193: Issues in Programming Systems
- 194: Issues in Theory of Computation
- 195: Issues in Numerical Analysis
- 196: Issues in Applications of Computer Science
- 197: Issues in Artificial Intelligence
- 198: Issues in Digital Media
- 199: Issues in Computer Science (Other)

Fall 2015

(CSC 199) Creative Computing. Quick! How much would a tunnel under Lake Ontario cost? How many people probably touched that orange you just bought at Wegmans? Can the military’s satellites really read your license plate from orbit? Explores the creative use computational mechanisms and information sources to obtain rough estimates and feasibility analyses for interesting questions practical problems, and looks at the technological basis of the art of measurement. No prerequisites.

Spring 2016

(CSC199) Social Implications of Computing. Computers and the Internet, perhaps more than any other technology, have transformed society over the past 50 years, with dramatic increases in human productivity; an explosion of options for news, entertainment, and communication; and fundamental breakthroughs in almost every branch of science and engineering. At the same time, they have contributed to unprecedented threats to privacy; whole new categories of crime and anti-social behavior; major disruptions in the job market; and the large-scale concentration of risk into systems capable of catastrophic failure. In this discussion- and writing-oriented class, we will consider all of this and more, with the goal of better understanding how to shape technological change in ways that maximize the benefits and minimize the costs. Offers upper-level writing credit in Computer Science. No prerequisites.

CSC 200/200H: Undergraduate Problem Seminar. Intensive seminar on cooperative problem solving. Overview of the subdisciplines and the research of the University of Rochester’s computer science faculty. 200H required for the Honors B.S. in Computer Science; Students considering graduate school would benefit from this course. Prerequisites: CSC171, CSC172, MTH 150, MTH 161, MTH 162.. (Fall & Spring)

CSC 209: Front-End Web Design and Development. "Front-end" is an industry term that refers to the focus on HTML, CSS and JavaScript, which differentiates this course from the formal programming courses.
- Topics will include Information Architecture, visual design, use of client libraries (mostly JS), and asset management strategies; we will also cover Content Management Systems and introduce web databases using PHP and MySQL. Prerequisite: CSC 170.

**CSC 210: Principles of Advanced Web Development** The World Wide Web was born around 1990, so it is not much older than most of you. In this course, we will follow the growth of the Web from its toddler years, to early childhood, to its turbulent pre-teen and teenage years, and finally as it begins to mature as a young adult. Along this journey, you will learn influential Web technologies such as HTTP, HTML, JavaScript, CSS, the LAMP stack, XML, JSON, Ajax, WebSockets, and modern MVC frameworks. Even though you will be doing a lot of programming in this course, its purpose is not to teach you to become an expert in any particular language or framework. Web technologies change at a blistering pace, so specifics quickly get outdated. However, once you take this course and understand the fundamentals, you will be able to easily pick up new technologies on the fly. Prerequisite: CSC172 or permission of the instructor. (Fall)

**CSC212/412: Human Computer Interaction.** This course will explore the design, implementation, and evaluation of user interfaces. Students will study the theoretical methods for interface design and evaluation, including requirements gathering, usability heuristics, user interface inspections, usability studies, information visualization, and prototyping. Case studies of interface successes and failures will augment theory with practical experiences. Students will apply this methodology to assignments in the design, implementation, and evaluation cycle. Students taking this course at the graduate level will have additional readings and assignments. Prerequisite: CSC 172 or permission of instructor. Programming experience is assumed. (Fall)

**CSC231 Robot Control.** This course covers control and planning algorithms with applications in robotics. Topics include transfer function models, state-space models, root-locus analysis, frequency-response analysis, Bode diagrams, controllability, observability, PID control, linear quadratic optimal control, model-predictive control, stochastic control, forward and inverse kinematics, dynamics, joint space control, operational space control, and robot trajectory planning. Proficiency with Matlab/C++ is recommended. Prerequisites: MTH 165 and ECE114, CSC 160 or CSC 171. (Fall)

**CSC232 – Autonomous Mobile Robots.** This course covers models and algorithms for autonomous mobile robots. Topics include sensors, perception, state estimation, mapping, planning, control, and human-robot interaction. Proficiency with Matlab/C++ is recommended. Lab required. Prerequisites: MTH 165 and ECE 114, CS 160, or CSC 171. (Spring)

**CSC 240 Data Mining.** Fundamental concepts and techniques of data mining, including data attributes, data visualization, data pre-processing, mining frequent patterns, association and correlation, classification methods, and cluster analysis. Advanced topics include outlier detection, stream mining, and social media data mining. CSC 440, a graduate-level course, requires additional readings and a course project. Prerequisites will be strictly enforced: CSC171, CSC 172 and MTH 161. Recommended: CSC 242 or CSC262; MTH165. (Fall)

**CSC 242: Artificial Intelligence.** Introduces fundamental principles and key applications of artificial intelligence, including heuristic search, automated reasoning, machine learning, neural networks and machine perception. Programming project include building autonomous software agent in a virtual world. This course is a prerequisite for advanced AI courses. Prerequisites: MTH 150 and CSC 172. Same as BCS 232. (Spring)

**CSC 244/444: Logical Foundations of Artificial Intelligence.** An introduction to the logical foundations of AI, including first-order logic, search, knowledge representation, planning. Students taking this course at the
400 level will be required to complete additional readings and/or assignments, including a significant project or essay. Prerequisites: CSC 173 and 242. (Fall)

**CSC 246/446: Machine Learning.** This course presents the mathematical foundations of AI, including probability, decision theory and machine learning. Prerequisites: CSC 242 and MTH 165. (Spring)

**CSC 247/447: Natural Language Processing.** An introduction to natural language processing: constructing computer programs that understand natural language. Topics include parsing, semantic analysis, and knowledge representation. CSC 447, a graduate-level course, requires additional readings and assignments. Prerequisite: CSC 242. (Spring or Fall; alternating years with CSC 248/448; cross-listed as BCS 235/535, LIN 247/447)

**CSC 248/448: Statistical Speech and Language Processing.** An introduction to statistical natural language processing and automatic speech recognition techniques. This course presents the theory and practice behind the recently developed language processing technologies that enable applications such as speech-driven dictation systems, document search engines (e.g., finding web pages) and automatic machine translation. Students taking this course at the 400 level will be required to complete additional readings and/or assignments. Prerequisites: CSC 172 and CSC 242. (Fall or Spring; Cross-listed with BCS 233/ BCS 533, LIN 248/448; alternating years with CSC 247/447)

**CSC 249/449: Machine Vision.** Introduction to computer vision, including camera models, basic image processing, pattern and object recognition, and elements of human vision. Specific topics include geometric issues, statistical models, Hough transforms, color theory, texture, and optic flow. CSC 449, a graduate-level course, requires additional readings and assignments. Prerequisites: MTH 161 and CSC 242. (Fall or Spring; may not be offered every year; cross-listed as BCS 236/536)

**CSC 252: Computer Organization.** Introduction to computer architecture and the layering of hardware/software systems. Topics include instruction set design; logical building blocks; computer arithmetic; processor organization; the memory hierarchy (registers, caches, main memory, and secondary storage); I/O—buses, devices, and interrupts; microcode and assembly language; virtual machines; the roles of the assembler, linker, compiler, and operating system; technological trends and the future of computing hardware. Several programming assignments required. Prerequisites: MTH 150 and CSC 172. (Spring)

**CSC 253/453 Dynamic Languages and Software Development.** This course explores unique aspects of dynamically-typed programming languages, which are now pervasive in domains such as scientific research, Web application development, gaming, and user interface design. The lessons you will learn here complement those in traditional compilers and programming languages courses, which focus mainly on statically-typed languages. We will use the Python language as a case study. In the first half of this course, we will study the internals of the Python interpreter, which is implemented in C. In the second half, we will build analysis and debugging tools for Python, potentially extending open-source tools with large user bases. Familiarity with Python and C are beneficial but not strictly required. Prerequisites: CSC 252, CSC 254. Offer alternate years with CSC 255/455. (Fall or Spring)

**CSC 254/454: Programming Language Design & Implementation.** Design and implementation of programming languages, with an emphasis on imperative languages and on implementation tradeoffs. In-depth examination of "how programming languages work." Topics include fundamental language concepts (names, values, types, abstraction, control flow); compilation and interpretation (syntactic and semantic analysis, code generation and optimization); major language paradigms (imperative, object-oriented, functional, logic-based, concurrent). Course projects include assignments in several different languages, with
an emphasis on compilation issues. Meets jointly with CSC 454, a graduate-level course that requires additional readings and assignments. Prerequisite: CSC 173; CSC 252 recommended. (Fall)

**CSC 255/455: Software Analysis and Improvement.** Programming is the automation of information processing. Program analysis and transformation is the automation of programming itself—how much a program can understand and improve other programs. Because of the diversity and complexity of computer hardware, programmers increasingly depend on automation in compilers and other tools to deliver efficient and reliable software. This course combines fundamental principles and (hands-on) practical applications. Specific topics include data flow and dependence theories; static and dynamic program transformation including parallelization; memory and cache management; type checking and program verification; and performance analysis and modeling. The knowledge and practice will help students to become experts in software performance and correctness. Course projects include the design and implementation of program analysis and improvement tools. Meets jointly with CSC 455, a graduate-level course that requires additional readings and assignments. Prerequisite: CSC 254; CSC 252 recommended. (Fall)

**CSC 256/456: Operating Systems** Principles of operating system design, explored within the practical context of traditional, embedded, distributed, and real-time operating systems. Topics include device management, process management, scheduling, synchronization principles, memory management and virtual memory, file management and remote files, protection and security, fault tolerance, networks, and distributed computing. CSC 456, a graduate-level course, requires additional readings and assignments. Prerequisite: CSC 252. (4 hours; Spring)

**CSC 257/457: Computer Networks.** Introduction to computer networks and computer communication: Architecture and Protocols: Design of protocols for error recovery, reliable delivery, routing and congestion control. Store-and-forward networks, satellite networks, local area networks and locally distributed systems. Case studies of networks, protocols and protocol families. Emphasis on software design issues in computer communication. Prerequisite: CSC 252 (Fall or Spring)

**CSC 258/458: Parallel and Distributed Systems.** Principles of parallel and distributed systems, and the associated implementation and performance issues. Topics covered will include programming interfaces to parallel and distributed computing, interprocess communication, synchronization, and consistency models, fault tolerance and reliability, distributed process management, distributed file systems, multiprocessor architectures, parallel program optimization, and parallelizing compilers. Students taking this course at the 400 level will be required to complete additional readings and/or assignments. Prerequisites: CSC 254, CSC 256, and consent of instructor. (Fall or Spring; may not be offered every year)

**CSC 259 Big Data Computer Systems.** Computer systems are increasingly driven by large, diverse data sets in a variety of domains including the web, health care, social networking, civil engineering, and earth and environmental sciences. This seminar course will discuss and explore computer systems issues for the big data era. Tentative topics will include the big data applications, big data programming (such as mapreduce), warehouse-scale data centers, big data storage, sustainable (energy-efficient) data processing, data reliability, durability, security, and privacy, as well as data-driven science and engineering in the wild. (pre-requisite: CSC 172, CSC 252 (May not be offered every year)

**CSC 260/460: Topics in Natural Language Dialog Systems.** This course will examine recent research in computational linguistics and artificial intelligence on natural language dialog systems. Students will take turns leading the discussion of current research papers. Undergraduates taking the course for credit will also be required to prepare a written review of one of the papers. Graduates taking the course may have additional readings or assignments. It may be repeated for credit with permission of the instructor. (Spring) Prereq.: CSC 244 and CSC 247.
CSC 261/461 Database Systems. This course presents the fundamental concepts of database design and use. It provides a study of data models, data description languages, and query facilities including relational algebra and SQL, data normalization, transactions and their properties, physical data organization and indexing, security issues and object databases. It also looks at the new trends in databases. The knowledge of the above topics will be applied in the design and implementation of a database application using a target database management system as part of a semester-long group project. Pre-requisite: CSC 172. Recommended: CSC 173 and CSC 252. (Spring).

CSC262/462 Computational Introduction to Statistics. This course will cover foundational concepts in probability and statistical inference, with an emphasis on topics of interest to computer scientists. Following an introduction to elementary probability theory, topics will include applications of combinatorics; Markov chains; principles of statistical classification (Bayes' rule, sensitivity and specificity, ROC curves) and random number generation. The theory of statistical estimation and hypothesis testing will be introduced, and applied to one and two sample inference for population means, proportions, variances and correlations. Nonparametric procedures will be discussed. Topics also include statistical modeling (ANOVA, simple and multiple regression), and computational methods. Students will be introduced to the R statistical computing environment. A background in calculus and linear algebra will be assumed.

CSC 263/463 Computational Models of Music. We will explore various computational approaches to musical problems (rule-based approaches, connectionism, dynamical systems, and probabilistic models), focusing on two main areas: 1) models of musical processing and information retrieval; 2) models of musical styles. Our focus will be on the symbolic level of music representation rather than on the signal level (there will be no signal processing in the course). Most assignments will consist of reading articles and answering questions about them. There will be some programming assignments, with other options for students without programming ability. Students should understand music notation and have knowledge of basic musical concepts such as key and meter. (not offered every year)

CSC 266/466: CPU Parallel Programming Using C/C++. GPU micro-architecture, including global memory, constant memory, texture memory, SP, SM, scratchpad memory, L1 and L2 cache memory, multi-ported memory, register file, and task scheduler. Parallel programming applications to parallel sorting, reduction, numeric iterations, fundamental graphics operations such as ray tracing. Desktop GPU programming using Nvidia's CUDA (Compute-Uniform Device Architecture). CPU/GPU cooperative scheduling of partially serial/partially parallel tasks. No midterms or written exams. Course consists of seven hands-on projects using CUDA. (Fall; Cross-listed as ECE 206/406) Prerequisites: ECE 200, or ECE 216, or ECE 201/401, or equivalent. Familiarity with assembly language and C programming language. Instructor approval.

CSC 280: Computer Models and Limitations. This course studies fundamental computer models and their computational limitations. Finite-state machines and pumping lemmas, the Chomsky hierarchy, Turing machines and algorithmic universality, noncomputability and undecidability, tradeoffs between power and formal tractability. Prerequisites: MTH 150 and CSC173. (Spring)

CSC 281/481: Cryptography. The modern study of cryptography investigates techniques for facilitating interactions between distrustful entities. With the advent of large-scale networked systems such as the Internet, such techniques have become indispensable—enabling, for instance, electronic voting, privacy-preserving auctions, internet banking, satellite radio/television and more. In this course we introduce some of the fundamental concepts of this study. Emphasis will be placed on the foundations of cryptography and in particular on precise definitions and proof techniques. Prerequisites: (MTH150 OR MTH162) AND (CSC171 or programming experience)
CSC 282: Design and Analysis of Efficient Algorithms. How does one design programs and ascertain their efficiency? Divide-and-conquer techniques, string processing, graph algorithms, mathematical algorithms. Advanced data structures such as balanced tree schemes. Introduction to NP-completeness and intractable combinatorial search, optimization, and decision problems. Prerequisites (MTH150 and CSC 172) OR MTH 172. (Fall)

CSC 283: Topics in Cryptography. This will be a seminar-style course in which students will read and present papers on current research in cryptography. Potential topics include lattice-based cryptography, concurrency and protocol security, database privacy, cryptographic game theory and interplay of cryptography with other fields. The course will build on material covered in the introductory course (281/481) but is not a required prerequisite. (May not be offered every year).

CSC 284/484: Advanced Algorithms. Advanced study of design and analysis of algorithms. Topics typically include: growth of functions; recurrences; probabilistic analysis and randomized algorithms; maximum flow; sorting networks; expander graphs; matrix operations; linear programming; discrete Fourier transform; number-theoretic algorithms; string matching; computational geometry; NP-completeness; approximation algorithms. Students taking this course at the 400 level may be required to complete additional tests, readings or assignments. Prerequisite: CSC 282. (Spring)

CSC 285/485: Algorithms and Elections. The focus of this course is on using algorithms to manipulate elections and on using complexity to protect elections from such manipulative attacks. Among the attacks we will study are manipulation, bribery and control. Students taking this course at the 400 level may be required to complete additional tests, readings, or assignments. Prerequisites: At least one of the following: MTH 150, MTH 143, MTH162, MTH172, PSC107, CSC280, CSC 281, CSC 282.

CSC 286/486: Computational Complexity. The difference between computable and uncomputable problems and between feasible and infeasible problems. Regarding the latter, what properties of a problem make it computationally simple? What properties of a problem may preclude its having efficient algorithms? How computationally hard are problems? Complete sets and low information content; P=NP?; unambiguous computation and one-way functions; reductions relating the complexity of problems; complexity classes and hierarchies. Prerequisite: CSC280.

CSC 287/487: Advanced Modes of Computation. Advanced modes of computation such as probabilistic computation, counting-based computation, semi-feasible computation, nondeterminism, computation trees, and parallel access. CSC 487, a graduate-level course, requires additional readings and assignments. Prerequisite: CSC 284 or CSC 286. (Fall or Spring; may not be offered every year)

CSC 291-299: Topics in Computer Science. This course covers special topics of current interest and usually differs each time it is offered. Courses have been separated by broad subject area and will be assigned individual course titles when offered: Prerequisite: varies with topic. (4 hours; Fall and/or Spring; may not be offered every semester)

• 291: Topics in Cognitive Science
• 292: Topics in Programming Languages
• 293: Topics in Programming Systems
• 294: Topics in Theory of Computation
• 295: Topics in Human Computer Interaction
• 296: Topics in Applications of Computer Science
• 297: Topics in Artificial Intelligence
Fall 2015


(CSC 299) Creative Computing. Quick! How much would a tunnel under Lake Ontario cost? How many people probably touched that orange you just bought at Wegmans? Can the military's satellites really read your license plate from orbit? Explores the creative use computational mechanisms and information sources to obtain rough estimates and feasibility analyses for interesting questions practical problems, and looks at the technological basis of the art of measurement.

Spring 2016

(CSC292) Computer Security Foundations. This course will teach students the foundations of computer security. Students will learn what security is, the design principles of secure systems, how security is enforced, and how security is compromised. Topics include access controls, information flow, basic applications of cryptography, buffer overflow attacks, and malware. Prerequisite: CSC252.

(CSC 297) Robot Construction. A robot can be defined as any artificial system that moves by itself under some sort of adaptive control. This course is intended to be a practical exercise in robot construction, with emphasis on creation of working robotic artifacts. The class will be divided into teams of 3-5, each of which will work to produce a separate mechanism. The robots to be built will be determined through discussion in the first couple of weeks. Prerequisite: CSC 172.

(CSC299) Social Implications of Computing. Computers and the Internet, perhaps more than any other technology, have transformed society over the past 50 years, with dramatic increases in human productivity; an explosion of options for news, entertainment, and communication; and fundamental breakthroughs in almost every branch of science and engineering. At the same time, they have contributed to unprecedented threats to privacy; whole new categories of crime and anti-social behavior; major disruptions in the job market; and the large-scale concentration of risk into systems capable of catastrophic failure. In this discussion- and writing-oriented class, we will consider all of this and more, with the goal of better understanding how to shape technological change in ways that maximize the benefits and minimize the costs. Offers upper-level writing credit in Computer Science. No prerequisites.

CSC 390: Supervised Teaching.

CSC 391: Independent Study in Computer Science. Special work arranged individually with a faculty member. (Fall and Spring)

CSC 391H: Honors Independent Study in Computer Science. Special work for Honors B.S. arranged individually with a faculty member. (Fall and Spring)

CSC 393: Senior Project. A one-semester senior project for computer science majors. Each project is arranged individually with a faculty adviser. Prerequisite: consent of the adviser.
CSC393H: Honors Senior Project. A one-semester senior project for computer science majors completing the Honors B.S.. Each project is arranged individually with a faculty adviser. Prerequisite: consent of the adviser.

CSC 394: Internship.

CSC 395: Research in Computer Science. Special problems may be arranged for advanced students wishing to do individual research. Requires consent of the Department.

CSC 395H: Honors Thesis Writing.

CSC 396: Research in Computer Science.