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 6/13/2011)

Contents

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 3-2 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 computational linguistics or computational biology. 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, within the context of a liberal arts education, computer scientists. 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, 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 opens 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 Advisors

Each computer science student has four levels of advising they can take advantage of at any time: their faculty advisor, the department undergraduate advisor, 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 indicating that computer science will be his or her major will receive a faculty advisor in computer science. Students and faculty advisors 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 advisor 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 the undergraduate advisor 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, 3-2 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 advisors 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 - Introduction to Programming and the Web

> This course shows students what goes on behind the scenes to make web pages appear in your browser, and offers a taste of web programming. This course is a good choice for any student who wants to do more than simply use computer applications, and come to understand how computers and software actually work. (Fall)

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. (Spring)

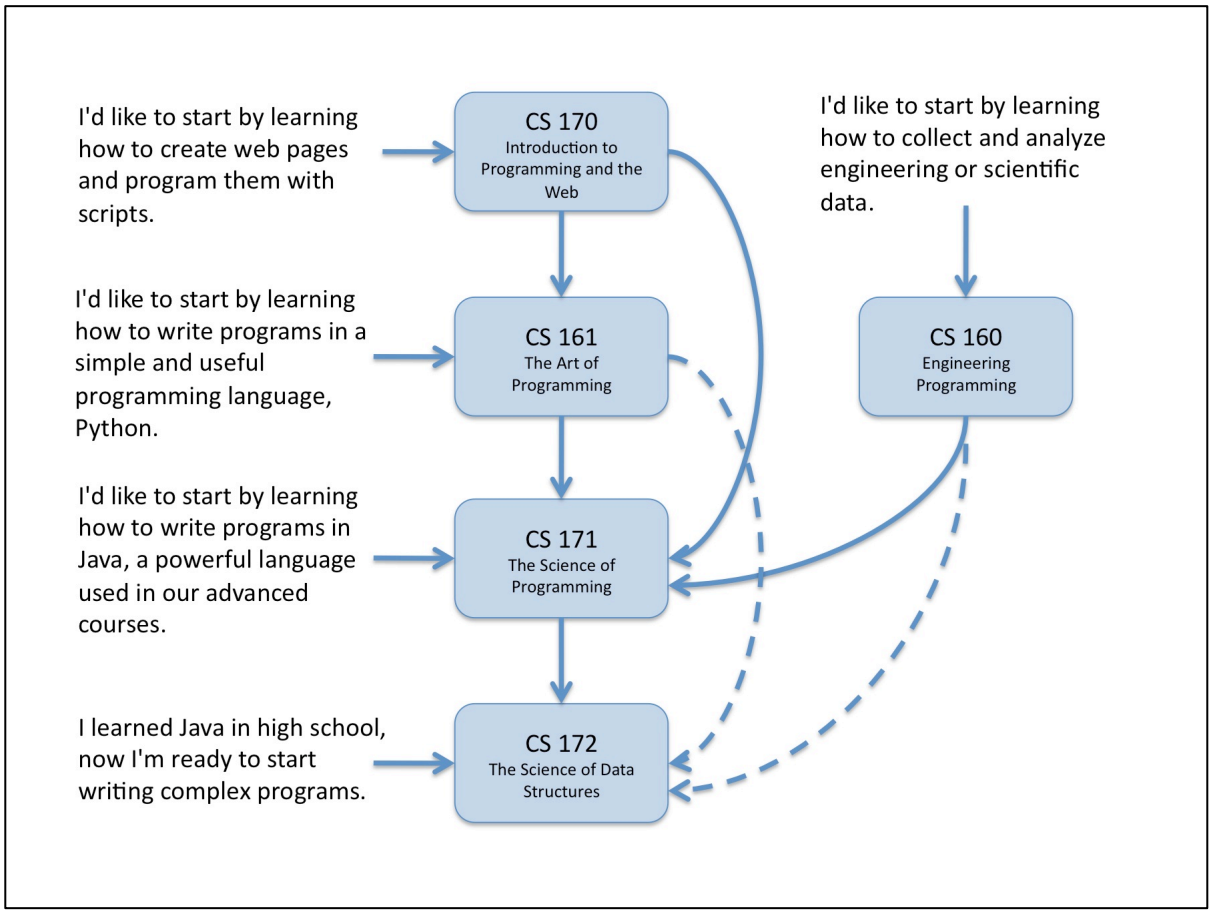
CSC 171 - The Science of Programming

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 will earn a B.S. in Computer Science usually begin with this course. (Fall)

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. (Spring)

These 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 a student 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 should obtain permission from the undergraduate advisor.



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 - Computer Applications

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 qualify for entering the B.S. program by completing the following courses:

MTH 150 - Discrete Mathematics or MTH 150A Discrete Math Module

MTH 161* - Calculus I

MTH 162* - Calculus II

CSC 171 - The Science of Programming (or AP credit, or programming experience with approval of the undergraduate program director)

CSC 172 - The Science of Data Structures

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

To be admitted into the B.S. program, a student must attain a grade of C- or higher in each of the above courses and a GPA of no lower than 2.0 in these courses, which must not be taken on a pass/fail basis. After the qualifying requirements are met, these courses are not included in the final GPA calculation for the B.S. 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

CSC 200/200H - Undergraduate Problem Seminar

MTH 165 - Linear Algebra with Differential Equations

Students who take both MTH 163 and MTH 235 or both MTH 173Q and MTH 174Q may count this two-course sequence as covering the MTH 165 requirement.

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 advisor or the department undergraduate advisor in CSB 735 about their advanced course selection. Specialization is encouraged, though not mandatory. 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 255, CSC 256, CSC 257, CSC 258
- Natural Language and Knowledge Representation—CSC 244, CSC 247, CSC 248
- Vision and Robotics— CSC 246, CSC 249
- Theory— CSC 281, CSC 284, CSC 286, CSC 287
- Human Computer Interaction and Web — CSC 210, CSC 212

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

CSC 393 - Senior Project, OR

A substitute for the senior project, which may include one of the following:

MTH 163, 164, 173, 174, or any additional Mathematics course numbered above 200

Courses 200 level or above in Electrical & Computer Engineering, Philosophy, Linguistics, Brain & Cognitive Sciences, or other computing-related discipline with advanced permission of the undergraduate program director. CIS 225 - Database Management may be used. CIS 215 may not be used.

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 advisor. 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. This phase should be completed by (and preferably before) the Fall semester of the senior year. Register for independent study (CSC 391 - 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 Liaison in the Undergraduate Program Office (CSB 735). (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 (Pre), MTH 162 (Pre), 2 electives; Spring: CSC 252 (Core), CSC280 (Core) or CSC200 (B.S.), 2 electives.
3rd Year:	Fall: CSC 282 (Core), MTH 165 (B.S.), CSC 254 (Core), 1 elective; Spring: CSC 200 (B.S.) or 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: CSC 252 (Core), CSC280 (Core) or CSC200 (B.S.), 2 electives.

3rd Year: Fall: CSC282 (Core), CSC 286 (AC1), 2 elective;
Spring: CSC 252 (Core) or CSC200 (B.S.), CSC 284 (AC2), 2 electives.

4th Year: Fall: CSC 254 (Core), CSC 287 (AC3), 2 electives;
Spring: MTH 165 (B.S.), CSC 242 (Core), CSC 393 (B.S.), 1elective.

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 200 (B.S.) or CSC 280 (Core), 2 electives.

3rd Year: Fall: CSC 282 (Core), MTH 165 (B.S.). 2 electives;
Spring: CSC 280 (Core) or CSC 200 (B.S.), CSC 252 (Core), 2 electives.

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

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, 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;

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. 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 course requirements before entry into the program, but two CSC courses above the level of 130 must be passed in order to enjoy the full privileges of a CSC major (non-expiring accounts, lab space, free printing, *etc*.).

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 5.2. 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 244 - Logical Foundations of AI: CSC 173, CSC 242

CSC 245 - Foundations of Vision (Cross-listed from BCS 222): MTH 162

CSC 246 - Mathematical Foundations of AI: CSC 242, MTH 165

CSC 249 – Machine Vision: CSC 242, MTH 161

CSC 29X - Robotics: CSC 242

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 - Mathematical Foundations of AI: 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 **

Graphics and Human-Computer Interaction: (Choose three or more)

CSC 131 - Recreational Graphics I

CSC 132 - Recreational Graphics II

CSC 212 - Human-Computer Interaction: CSC 161 or CSC 171

CSC 210 - Web Programming: CSC 161 or CSC 171

BCS 228 - The Human-Machine Interface **

MTH 215 - Fractals and Computer Graphics **

Theory of Computation: (Choose three or more)

CSC 284 - Advanced Algorithms: CSC 282

CSC 286 - Computational Complexity: CSC 280

CSC 287 - Advanced Modes in Computation: CSC 286

MTH 248 - Theory of Graphs **

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

ECE 201 - Advanced Computer Architecture **

Computer Security: (Choose three or more)

CSC 257 - Computer Networks: CSC 252

CSC 281 - Cryptography: CSC 161 or CSC 171, CSC 280

CSC 29x - Introduction to Computer Security: CSC 172

CSC 29x - Social Implications of Computing

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

CSC 258 - Parallel and Distributed Computing: CSC 252

CSC 29X - Introduction to Database Systems: CSC 161 or CSC 171 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 **

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

5.2 Creating New Tracks

In consultation with a faculty member or the department's undergraduate advisor, 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 advisor may design and propose a new track for approval by the undergraduate program director.

5.3. Getting Started in the BA

Students begin the B.A. by taking an introductory sequence of Computer Science courses through CSC 172.

We recommend that B.A. students take the following the three courses in order to obtain breadth in computer science, and because these courses appear in the prerequisites of many tracks:

CSC 242 - Artificial Intelligence

CSC 252 - Computer Organization

CSC 282 - Design and Analysis of Efficient Algorithms

Beyond these recommendations, students should plan their B.A. for an existing or new track in consultation with the department's undergraduate advisor. 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 courses 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 advisor, 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 advisor. Acceptable writing experiences include (but at the advisor's discretion are not limited to) the following:

- 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 advisor 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 advisor as satisfying the Computer Science requirements.
- Designated courses in Computer Science. Each semester the Computer Science Department Curriculum Committee may designate certain courses as satisfying the upper-level writing requirement. The set of such courses may vary from semester to semester, and from instructor to instructor. All such courses must meet the standard described above; designation by the Computer Science Curriculum Committee will simply relieve the advisor of the need for verification.
- 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 advisor 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).

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, CIS 215 or CSC 225, and either 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): CSC 108, AH100 or CSC 131, and either CSC 161, CSC 170 or CSC 171. This cluster introduces students to the use of computers in visual art.

- Computing for the Social Sciences (N4CSC010): CSC 108, one of the following: STT 211, STT 212, STT 213, PSC 200 or PSC 201, and either CSC161, CSC 170 or CSC171. 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): CSC172, CSC282 and either CSC171 or CSC173. Emphasizes algorithmic thought, use of data structures and the design and analysis of efficient algorithms.

- Human Computer Interaction (N4CSC013): Choose one of two: CSC 108, CSC 131; and choose one of two: CSC 161 or CSC 171; and CSC 212 Human Computer Interaction. A broad sampling of common computer applications in the sciences and humanities, with the possibility of proceeding more deeply into the principles of HCI through programming and introductory HCI.

- Computer Graphics (N4CSC014): EAS 106, CSC 108, and CSC 131. Teaches the use of modern computational systems as a tool for creative artistic expression. Focuses on 3D computer graphics and animation techniques. Suitable for students with an interest in creating animated video or interaction digital media (computer games).

- Computational Problem Solving (N4CSC015): CSC 108; either CSC 161 or CSC171; CSC 172. An introduction to problem solving through standard computer applications and computer programming. Suitable for students in a wide variety of disciplines.

- Management of Data (N4CSC016): Either CSC 161 or CSC171; CSC172; choose one: CSC 29X Intro to Database Systems. CIS 215 or CIS 225). Computer programming and software packages with an emphasis on the application of databases and information systems to problems in business and commerce.

9. Opportunities

There are several opportunities available for students in Computer Science

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 3-2 Program

The 3-2 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 advisor 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).

10. Course Descriptions

CSC 108. Computer Applications. 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 I. 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 166. Video Game Programming. Do you like to play video games? Why not learn how to make one? This course is a hands-on lab based introduction to software engineering and computer programming using the development of computer/video games as the application area. The course is taught using the C# language with the XNA game development framework targeting applications for the XBOX360 game platform. Students will learn the basics of computer programming and the basics of the management and development processes of software engineering. This course is intended for students with little or no previous programming experience.

CSC 170. Introductory Computer Programming. The course is taught using the Javascript programming language and HTML, but it emphasizes algorithmic thinking and creative problem solving over language specifics. Grades are based on projects and exams. Lab required. Not open to officially declared CSC majors. No prerequisites. (4 hours; Fall)

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 only)

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 recommended. (Fall only)

CSC 173: Computation and Formal Systems. We investigate several formal systems influential in computer science, and also some of their applications (e.g. inspiring and providing the foundation for a computer programming style, or providing the basis for solving important practical problems like communications protocols, compiling, systems analysis, graphics ...) In more detail, we study: propositional and predicate logic and applications like the Prolog language and circuit design; formal languages and automata theory (FLAT) and applications like scanners and parsers, using the C Language; lambda calculus and the Scheme language with an AI application; matrices and the Matlab language, with applications in robotics or graphics. Prerequisite: 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 2011

- **Computational Models of Music. (CSC198)** 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. Prerequisites: Students should understand music notation and have knowledge of basic musical concepts such as key and meter. (Fall).
- **Creative Computing. (CSC 199)** 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 2012

- **Social Implications of Computing. (CSC199)** 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; 200 required for the B.S. Students taking CSC 200H may have additional reading, assignments or projects. Prerequisites: all pre-major requirements. (4 hours; Spring)

CSC 210: Web Programming An introduction to the technology, design and science of web programming. This course will cover the base material needed to create and deploy secure, usable database-driven web applications - including topics selected from programming, networking, databases, security, and usability. Specific technologies and languages covered will include HTML, Javascript, Document Object Model (DOM), PHP, MySQL, Ruby on Rails, XML, AJAX, and Flash. Prerequisite: CSC 161 or CSC 171 or permission of the instructor. (4 hours, Spring

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 171 or permission of instructor. Programming experience is assumed.

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 agenst 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. (4 hours; Fall)

CSC 246/446: Mathematical Foundations of Artificial Intelligence. 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. (4 hours; 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. (4 hours; 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. (4 hours; 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. (4 hours; Spring)

CSC 254: 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.. Prerequisite: CSC 173; CSC 252 recommended. (4 hours; Fall)

CSC 255/455: Advanced Programming Systems. With the increasing diversity and complexity of computers and their applications, the development of efficient, reliable software has become increasingly dependent on automatic support from compilers & other program analysis and translation tools. This course covers principal topics in understanding and transforming programs at the assembly, function, and program levels. Specific techniques for imperative languages include data flow, dependence, and inter-procedural analyses; resource allocation; and program transformation for locality and parallelism. The course will also touch on theoretical issues in program semantics for higher order languages. Course projects include a program analyzer and optimizer for a subset of the C programming language. Meets jointly with CSC 455, a graduate-level course that requires additional readings and assignments. Prerequisite: CSC 254; CSC 252 recommended. (4 hours; 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 (4 hours; 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. (4 hours; Fall or Spring; 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 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 Crosslisted 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: CSC 162 OR CSC 172 OR MTH 172. (4 hours; Spring)

CSC 281/481: Intro to 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 CSC 162 OR CSC 172 OR MTH 172. (4 hours; Fall)

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. (4 hours; Spring)

CSC 286/486: Computational Complexity. This course continues the development of the theory of computing begun in CSC 280. Topics include the formal characterization of computational hardness; one-way functions and cryptography; the complexity hierarchy; and information theory. Prerequisite: CSC 280. (Fall)

CSC 287/487: Randomized, Parallel, and Other 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. (4 hours; 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 Numerical Analysis
- 296: Topics in Applications of Computer Science
- 297: Topics in Artificial Intelligence
- 298: Topics in Digital Media
- 299: Topics in Computer Science (Other)

Fall 2011

(297): Pervasive Computing Pervasive (or ubiquitous) computing systems integrate information processing into everyday objects and activities. Examples include smart phones, intelligent homes, tangible and ambient interfaces, sensor dust, and augmented reality systems. In this reading and project-based class, we will explore recent research in pervasive computing, and brainstorm, design, and prototype novel pervasive computing systems. Students will work in groups to research topics and report back to the class. Programming projects will also be group-based and open-ended, and will in general require significant independent work in learning to program and/or hack devices such as the Xbox Kinect, Wii controller, iPhone, etc. Prerequisites: 172 required; 210 and 252 recommended.

(298) 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.

(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 2012

(291) Machines and Consciousness. An exploration of the possibility of consciousness in machines, both in the sense of perceptual awareness and self-awareness. Readings are from the AI literature as well as from philosophy and cognitive science, with emphasis on scientific and computational aspects. The course will be suitable for writing credit. Prerequisites: CSC 173 and one or more of 240, 242, 280 , or 282. (4 hours, Spring)

(293) Advanced Topics in Memory Systems. Advanced topics in the organization, architecture, and implementation of modern memory subsystems. Topics include power, performance, reliability, and QoS issues in DRAM memory systems and Flash-based SSDs; high-performance memory controllers and interfaces; memory system design for datacenters and enterprise systems; and an introduction to emerging resistive memory technologies. Prerequisites: CSC 252 and ECE 201/401, or permission of instructor.

(296) 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.

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 advisor. Prerequisite: consent of the advisor.

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 advisor. Prerequisite: consent of the advisor.

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. Fellowship program for summer research, typically off-campus. Special application required; deadline February 15. Permission of instructor required.