MAJOR

The goal of the major is to provide an understanding of algorithmic problem solving as well as the conceptual organization of computers and complex programs running on them. Emphasis is placed on the fundamental principles of computer science, building upon the mathematical and theoretical ideas underlying these principles. The introductory and core courses build a broad and solid base for understanding computer science. The more advanced courses allow students to sample a variety of specialized areas including graphics, artificial intelligence, computer architecture, networks, compiler design, and operating systems. Independent study and honors work provide opportunities for students to study and conduct research on topics of special interest.

The major in Computer Science equips students to pursue a wide variety of career opportunities. It can be used as preparation for a career in computing, for graduate school, or to provide important background and techniques for the student whose future career will extend outside of computer science.

MAJOR REQUIREMENTS

Required Courses in Computer Science

A minimum of 8 courses is required in Computer Science, including the following:

<table>
<thead>
<tr>
<th>Introductory Courses</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Science 134</td>
<td>Introduction to Computer Science</td>
</tr>
<tr>
<td>Computer Science 136</td>
<td>Data Structures and Advanced Programming</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Core Courses</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Science 237</td>
<td>Computer Organization</td>
</tr>
<tr>
<td>Computer Science 256</td>
<td>Algorithm Design and Analysis</td>
</tr>
<tr>
<td>Computer Science 334</td>
<td>Principles of Programming Languages</td>
</tr>
<tr>
<td>Computer Science 361</td>
<td>Theory of Computation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Electives</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Two or more electives (bringing the total number of Computer Science courses to at least 8) chosen from: 300- or 400-level courses in Computer Science. At least one of these must be a course designated as a PROJECT COURSE. Computer Science courses with 9 as the middle digit (reading, research, and thesis courses) will normally not be used to satisfy the elective requirements. Students may petition the department to waive this restriction with good reason.</td>
<td></td>
</tr>
</tbody>
</table>

Required Courses in Mathematics

Mathematics 200 Discrete Mathematics

and any other Mathematics or Statistics course at the 200-level or higher

Students considering pursuing a major in Computer Science are urged to take Computer Science 134 and to begin satisfying their mathematics requirements early. Note in particular that Discrete Mathematics covers material complementing that in the introductory courses (Computer Science 134 and 136) and is a prerequisite for many advanced courses.

Students who take Computer Science 102T, 107, or 109 may use that course as one of the two electives required for the major in Computer Science. Those who count Computer Science 109 toward the major must select an elective different from Computer Science 371 (Computer Graphics) for their project course. Computer Science 102T, 107, 109, and 134 are not open to students who have taken a Computer Science course numbered 136 or higher.

To be eligible for admission to the major, a student must normally have completed Computer Science 136 as well as Discrete Mathematics by the end of the sophomore year. A second Mathematics course at the 200-level or higher must be completed by the end of the junior year. Students are urged to have completed two of the four core courses (Computer Science 237, 250, 334, and 361) by the end of the sophomore year and must normally have completed at least three out of the four core courses by the end of the junior year.

All senior majors must attend and participate in the Computer Science Colloquium. With the advance permission of the department, two appropriate mathematics or statistics courses may be substituted for one Computer Science elective. Appropriate mathematics classes are those numbered 300 or above, and appropriate statistics courses are those numbered 200 or above. Other variations in the required courses, adapting the requirements to the special needs and interests of the individual student, may be arranged in consultation with the department.

LABORATORY FACILITIES

The Computer Science Department maintains two departmental computer laboratories for students taking Computer Science courses, as well as a lab that can be configured for teaching specialized topics such as robotics. The workstations in these laboratories also support student and faculty research in computer science.

THE DEGREE WITH HONORS IN COMPUTER SCIENCE

The degree with honors in Computer Science is awarded to students who have demonstrated outstanding intellectual achievement in a program of study extending beyond the requirements of the regular major. The principal considerations in recommending a student for the degree with honors will be: mastery of core material, ability to pursue independent study of computer science, originality in methods of investigation, and creativity in research. Honors study is highly recommended for those students with strong academic records in computer science who wish to attend graduate school, pursue high-level industrial positions in computing, or who would simply like to experience research in computer science.

Prospective honors students are urged to consult with their departmental advisor at the time of registration in the spring of the sophomore or at the beginning of the junior year to arrange a program of study that could lead to the degree with honors. Such a program normally consists of Computer Science 493 and 494 and a WSP of independent research under the guidance of a Computer Science faculty member, culminating in a thesis that is judged acceptable by the department. The program produces a significant piece of written work and often includes a major computer program. All honors candidates are required to give an oral presentation of their research in the Computer Science Colloquium in early spring semester.

Students considering honors work should obtain permission from the department before registering in the fall of the senior year. Formal admission to candidacy occurs at the beginning of the spring semester of the senior year and is based on promising performance in the fall semester and winter study units of honors work. Recommendations for the degree with honors will be made for outstanding performance in the three honors courses. Highest honors will be recommended for students who have displayed exceptional ability, achievement, or originality.

INTRODUCTORY COURSES


Computer Science 134 provides an introduction to computer science with a focus on developing computer programming skills. These skills are essential to most upper-level courses in the department. As a result, Computer Science 134 together with Computer Science 136 are required as a prerequisite to most advanced courses in the department.

Those students intending to take several Computer Science courses are urged to take 134 early.

Those students interested in learning more about exciting new ideas in computer science, but not necessarily interested in developing extensive programming skills, should consider Computer Science 102: The Socio-Techno Web, 107: Creating Games, or 109: The Art and Science of Computer Graphics.

Students with significant programming experience should consider electing Computer Science 136 (see “Advanced Placement” below). Students are always welcome to contact a member of the department for guidance in selecting a first course.
STUDY ABROAD

This course introduces many fundamental concepts in computer science by examining the social aspects of computing. As more and more people use the technologies and services available via the Internet, online environments like Facebook, Amazon, Google, Twitter, and blogs are flourishing. However, several of the problems related to security, privacy, and trust that exist in the real world transform and become amplified in the virtual world created by the ubiquity and pervasiveness of the Internet. In this course, we will investigate how the social, technological, and natural worlds are connected, and how the study of networks sheds light on these connections. Topics include the structure of the Social Web and networks in general; issues such as virtual identity, personal and group privacy, trust evaluation and propagation, and online security; and the technology, economics, and politics of Web information and online communities. No background in computer science or programming is required or expected.

Format: tutorial. Requirements: evaluation will be based on tutorial discussions, presentations, problem sets and labs, a midterm exam, and a final project or paper

No prerequisites. Enrollment limit: 10 (expected: 19). Preference given to first-year students and sophomores who have not previously taken a computer science course.

May not be taken on a pass/fail basis.

ALBRECHT

CSCI 102T The Socio-Techno Web (Not offered 2013-2014) (Q)

The course introduces many fundamental concepts in computer science by examining the social aspects of computing. As more and more people use the technologies and services available via the Internet, online environments like Facebook, Amazon, Google, Twitter, and blogs are flourishing. However, several of the problems related to security, privacy, and trust that exist in the real world transform and become amplified in the virtual world created by the ubiquity and pervasiveness of the Internet. In this course, we will investigate how the social, technological, and natural worlds are connected, and how the study of networks sheds light on these connections. Topics include the structure of the Social Web and networks in general; issues such as virtual identity, personal and group privacy, trust evaluation and propagation, and online security; and the technology, economics, and politics of Web information and online communities. No background in computer science or programming is required or expected.

Format: tutorial. Requirements: evaluation will be based on tutorial discussions, presentations, problem sets and labs, a midterm exam, and a final project or paper

No prerequisites. Enrollment limit: 10 (expected: 19). Preference given to first-year students and sophomores who have not previously taken a computer science course.

May not be taken on a pass/fail basis.

ALBRECHT

CSCI 107 Creating Games (Same as ARTS 107) (Not offered 2013-2014) (Q)

The game is unique as the only broadly-successful interactive art form. Games communicate the experience of embodying a role by manipulating the player’s own decisions, abstraction, and discrete planning. Those three elements are the essence of computation, which makes computer science theory integral to game design. Video games also co-opt programming and computer graphics as new tools for the modern artist. As a result, games are collaborative interdisciplinary constructs that use computation as a medium for creative expression.

Students analyze and extend contemporary video and board games using the methodology of science and the language of the arts. They explore how computational concepts like recursion, state, and complexity apply to interactive experiences. They then synthesize new game elements using mathematics, programming, and both digital and traditional art tools. Emphasis is on the theory of design in modern European board games. Topics covered include experiment design, gameplay balance, minimax, color theory, pathfinding, game theory, composition, and computability.

Format: lecture and studio. Requirements: participation, studio work, quizzes.

No prerequisites; not open to students who completed a CSCI course numbered 136 or above; this course does not count toward the Art Major. Enrollment limit: 19 (expected: 19).

Preference given to first-year students.

Lab fee of $25 will be added to the student’s term bill.

MCGUIRE

CSCI 109(S) The Art and Science of Computer Graphics (Q)

This course provides an opportunity to develop an understanding of the theoretical and practical concepts underlying 2- and 3-dimensional computer graphics. The course will emphasize hands-on studio/laboratory experience, with student work focused around completing a series of projects. Students will experiment with modeling, color, lighting, perspective, and simple animation. They will use the computer to create digital art, and the studio to create 2D and 3D computer graphics. Emphasis is on the theory of design in modern European board games. Topics covered include experiment design, gameplay balance, minimax, color theory, pathfinding, game theory, composition, and computability.

Format: lecture and studio. Evaluation will be based on progress in project work and two examinations.

This course is not open to students who have successfully completed a CSCI course numbered 136 or above. No enrollment limit (expected: 20).

Hour: 8:30-9:45 TR Labs: 1:00-2:25 R, 2:35-4:00 R

BAILEY

COURSES INTENDED FOR BOTH NON-MAJORS AND MAJORS

CSCI 134(E) Introduction to Computer Science (Q)

This course introduces fundamental ideas in computer science and builds skills in the design, implementation, and testing of computer programs. Students implement algorithms in the Java programming language with a strong focus on constructing correct, understandable, and efficient programs. Students explore the material through specific application areas. Topics covered include object-oriented programming, control structures, arrays, recursion, and event-driven programming. This course is appropriate for all students who want to create software and have little or no prior computing experience. More details are available on the department website, http://www.cs.williams.edu.
This course builds on the programming skills acquired in Computer Science 134. It couples work on program design, analysis, and verification with an introduction to the study of data structures. Data structures capture common ways in which to store and manipulate data, and they are important in the construction of sophisticated computer programs. Students are introduced to some of the most important and frequently used data structures: lists, stacks, queues, trees, hash tables, and graphs. Students will be expected to write several programs, ranging from very short programs to more elaborate systems. Emphasis will be placed on the development of clear, modular programs that are easy to read, debug, verify, analyze, and modify.

Format: lecture/laboratory. Evaluation will be based on programming assignments and examinations.

Prerequisites: CSCI 134 or equivalent. (Discrete Mathematics is recommended, but not required). Enrollment limit: 36 (expected: 30). Preference: first-year students and sophomores.

Hour: 9:00-9:50 MWF
10:00-10:50 MWF
Lab: 1-4 W
First Semester: BAILEY
Second Semester: ALBRECHT

CSCI 337T(S) Digital Design and Modern Architecture (Q)
This tutorial course considers topics in the low-level design of modern architectures. Course meetings will review problems of designing effective architectures including instruction-level parallelism, branch-prediction, caching strategies, and advanced ALU design. Readings will be taken from recent technical literature. Labs will focus on the development of custom CMOS circuits to implement projects from gates to bit-sliced ALU’s. Final group projects will develop custom logic demonstrating concepts learned in course meetings.

Format: tutorial. Evaluation will be based on microprocessor design projects, participation in tutorial meetings, and examinations.

Prerequisite: CSCI 237. Enrollment limit: 10 (expected: 10). Preference given to seniors, followed by juniors. May not be taken on a pass/fail basis.

Hour: Tutorial meetings to be arranged
First Semester: MURTAGH
Second Semester: BAILEY

CSCI 339(S) Distributed Systems (Q)
This course studies the key design principles of distributed systems, which are collections of independent networked computers that function as single coherent systems. Covered topics include protocols, processes and threads, naming, synchronization, consistency and replication, fault tolerance, and security. Students also examine some specific real-world distributed systems case studies, ranging from the Internet to file systems. Class discussion is based on readings from the textbook and research papers. The goals of this course are to understand how large-scale computational systems are built, and to provide students with the tools necessary to evaluate new technologies after the course ends.

Format: lecture/laboratory. Evaluation will be based on homework assignments, programming projects, and exams.

Prerequisites: CSCI 136 (Data Structures) or equivalent programming experience, and Computer Science 237 (Computer Organization), or permission of the instructor. No enrollment limit (expected: 20).

PROJECT COURSE
Hour: 11:20-12:25 TR
First Semester: MURTAGH
Second Semester: BAILEY

CSCI 356T Advanced Algorithms (Not offered 2013-2014) (Q)
This course explores advances in algorithm design, algorithm analysis and data structures. The primary focus is on randomized and approximation algorithms, randomized and advanced data structures, and algorithmic complexity. Topics include combinatorial algorithms for cut, packing, and covering problems, linear programming algorithms, approximation schemes, hardness of approximation, random search trees, and hashing.

Format: tutorial. Evaluation is based on weekly problem sets, several small programming projects, weekly paper summaries, and a small, final project.

Prerequisites: CSCI 256. Computer Science 361 is recommended but not required. Enrollment limit: 10 (expected: 10) Preference given to Computer Science majors. May not be taken on a pass/fail basis.

HEERINGA

CSCI 361(F) Theory of Computation (Same as MATH 331) (Q)
This course introduces a formal framework for investigating both the computability and complexity of problems. We study several models of computation including finite automata, regular languages, context-free grammars, and Turing machines. These models provide a mathematical basis for the study of computability theory—the examination of what problems can be solved and what problems cannot be solved—and the study of complexity theory—the examination of how efficiently problems can be solved. Topics
include the halting problem and the P versus NP problem.

Format: lecture. Evaluation will be based on problem sets, a midterm examination, and a final examination.
Prerequisites: CSCI 256 or both a 300-level MATH course and permission of instructor. Enrollment limit: 30 (expected: 20).

Hour: 12:00-12:50 MWF

MURTAGH

CSCI 371 Computational Graphics (Not offered 2013-2014) (Q)
PhotoShop, medical MRIs, video games, and movie special effects all programmatically create and manipulate digital images. This course teaches the fundamental techniques behind these applications. We begin by building a mathematical model of the interaction of light with surfaces, lenses, and an imager. We then study the data structures and processor architectures that allow us to efficiently evaluate that physical model.

Students will complete a series of programming assignments for both photorealistic image creation and real-time 3D rendering using C++, OpenGL, and GLSL. These assignments cumulate in a multi-week final project. Topics covered in the course include: projective geometry, ray tracing, bidirectional surface scattering functions, binary space partition trees, matting and compositing, shadow maps, cache management, and parallel processing on GPUs.

Format: lecture, with optics laboratory exercises. Evaluation based on assignments, projects, and exams.
Prerequisites: CSCI 136 and CSCI 237 OR permission of the instructor. No enrollment limit (expected: 20).

PROJECT COURSE

MCGUIRE

CSCI 373(F) Artificial Intelligence (Q)
This course introduces fundamental techniques in the field of Artificial Intelligence, which is concerned with the ability to create machines that perform tasks requiring “intelligence.” The course covers methods for knowledge representation, reasoning, problem solving, and learning. It then explores those further by surveying current applications in selected areas such as game playing and natural language processing.

Format: lecture/laboratory. Several programming projects in the first half of the semester and a larger project spanning most of the second half account for 70% of the student’s final grade. A midterm examination and a six-page survey paper account for the remainder of the student’s grade.
Prerequisites: CSCI 136 and Discrete Mathematics. No enrollment limit (expected: 20).

Hour: 10:00-10:50 MWF

DANYLUK

CSCI 374T Machine Learning (Not offered 2013-2014) (Q)
This tutorial examines the design, implementation, and analysis of machine learning algorithms. Machine Learning is a branch of Artificial Intelligence that aims to develop algorithms that will improve a system’s performance. Improvement might involve acquiring new factual knowledge from data, learning to perform a new task, or learning to perform an old task more efficiently or effectively. This tutorial will cover examples of supervised learning algorithms (including decision tree learning, support vector machines, and neural networks), unsupervised learning algorithms (including k-means and expectation maximization), and possibly reinforcement learning algorithms (such as Q learning and temporal difference learning). It will also introduce methods for the evaluation of learning algorithms, as well as topics in computational learning theory.

Format: tutorial. Evaluation will be based on presentations, problem sets, programming exercises, empirical analyses of algorithms, critical analysis of current literature, and a final exam.
Prerequisites: CSCI 136 and MATH 200 (formerly 251). CSCI 256 is recommended but not required. Enrollment limit: 10 (expected: 10). Preference given to Computer Science majors.

May not be taken on a pass/fail basis.

DANYLUK

CSCI 397(F), 398(S), 497(F), 498(S) Reading
Directed independent reading in Computer Science.
Prerequisites: permission of the department.

Hour: TBA

Members of the Department

CSCI 432 Operating Systems (Not offered 2013-2014) (Q)
This course explores the design and implementation of computer operating systems. Topics include historical aspects of operating systems development, systems programming, process scheduling, synchronization of concurrent processes, virtual machines, memory management and virtual memory, I/O and file systems, system security, os/architecture interaction, and distributed operating systems.

Format: lecture/laboratory. Evaluation will be based on several implementation projects that will include significant programming, as well as written homework and exams.
Prerequisites: CSCI 237 and either CSCI 256 or 334. No enrollment limit (expected: 20). Preference: current or expected Computer Science majors.

PROJECT COURSE

ALBRECHT

CSCI 434T Compiler Design (Q)
This tutorial covers the principles and practices for the design and implementation of compilers and interpreters. Topics include all stages of the compilation and execution process: lexical analysis; parsing; symbol tables; type systems; scope; semantic analysis; intermediate representations; run-time environments and interpreters; code generation; program analysis and optimization; and garbage collection. The course covers both the theoretical and practical implications of these topics. As a project course, students will construct a full compiler for a simple object-oriented language.

Format: tutorial. Evaluation will be based on presentations, problem sets, a substantial implementation project, and two exams.
Prerequisites: CSCI 237 and 361 (concurrent enrollment is acceptable). CSCI 334 is recommended, but not required. Enrollment limit: 10 (expected: 10). Preference given to current or expected Computer Science majors.

PROJECT COURSE

May not be taken on a pass/fail basis.

Tutorial meetings to be arranged

FREUND

CSCI 493(F) Research in Computer Science
This course provides highly-motivated students an opportunity to work independently with faculty on research topics chosen by individual faculty. Students are generally expected to perform a literature review, identify areas of potential contribution, and explore extensions to existing results. The course culminates in a concise, well-written report describing a problem, its background history, any independent results achieved, and directions for future research.

This course (along with CSCI W31 and CSCI 494) is required for students pursuing honors, but enrollment is not limited to students pursuing honors.
Evaluation will be based on class participation, presentations, and the final written report.
Enrollment is limited. Open to senior Computer Science majors with permission of instructor.

Hour: TBA

Members of the Department

CSCI W31-494(S) Senior Thesis
Prerequisites: Computer Science 493.

CSCI 499(FS) Computer Science Colloquium
Required of senior Computer Science majors, and highly recommended for junior Computer Science majors. Meets most weeks for one hour, both fall and spring.

Hour: 2:35-4:00 F

FREUND