COMPUTER SCIENCE

Ellen Haberoth, Regents Math 307
507-786-3113
habero1@stolaf.edu
wp.stolaf.edu/cs (http://wp.stolaf.edu/cs)

(Mathematics, Statistics, and Computer Science)

Computer science (CS) is the academic discipline that focuses on creative computing-related problem solving. St. Olaf's CS program employs "hands-on" personal experience to build up invaluable technical and analytical skills while learning powerful computing concepts in a liberal arts context. Beginning with the introductory courses, the program's curriculum draws connections with applications in other disciplines ranging from the natural sciences to the humanities. The program offers an authentic and satisfying education in the concepts and practices of computer science; the courses below represent national expectations for an undergraduate computer science curriculum. Undergraduate research appears throughout the program, from foundation courses that develop valuable project skills to advanced courses such as the senior capstone seminar. St. Olaf is a national leader in incorporating instruction in parallel and distributed computing throughout the CS curriculum. The program also incorporates a distinct liberal arts perspective, including emphasis on teamwork and communication skills, examination of ethical and social issues in computing, and collaboration in upper-level interdisciplinary projects.

Although CS differs from other areas of computing, such as systems (IS), the study of computer science serves as excellent preparation for careers in any computing-related field, because the concepts of computer science provide insights into all types of computation. While specific computing systems come and go, the principles of CS endure for the long term, and people with awareness of those principles can perceive them in all forms of computing. The increasing effectiveness of computing in nearly all endeavors makes CS concepts relevant in almost every setting. The CS program emphasizes development of creative problem-solving and analytical thinking abilities, interpersonal skills, ethical analysis and awareness, and realistic applications. Development of these skills enhance any career, whether one becomes a computing professional or an occasional user.

Overview of the Major

The CS program emphasizes the concepts and practices of computer science, as well as applications to other disciplines. The major begins with foundation courses that present the nature of CS through hands-on experience. The foundation and subsequent core courses together span the national expectations for an undergraduate computer science major curriculum, and advanced courses and electives provide options for depth. Several themes appear throughout the major: breadth-first introductory courses; team collaboration (often interdisciplinary) and project-based learning; development of communication skills; thoughtful, structured analysis of ethical and social issues in computing; and undergraduate research, beginning with project-building skills in early courses and continuing through advanced experiences such as CSCI 390 Senior Capstone Seminar.

Intended Learning Outcomes for the Major (http://wp.stolaf.edu/curriculum-committee/computer-science-major-ilos)

Distinction

See Academic Honors (http://catalog.stolaf.edu/academic-regulations-procedures/academic-honors/#distinction)

Other Fields

Certain courses in computer science can count toward other majors and concentrations.

Majors

Code	Title	Credits
Mathematics		
CSCI 315	Bioinformatics	1.00
CSCI 333	Theory of Computation	1.00

Concentrations

Code	Title	Credits
Linguistics		
CSCI 121	Principles of Computer Science	1.00
CSCI 125	Computer Science for Scientists and Mathematicians	1.00
CSCI 251	Software Design and Implementation	1.00
CSCI 276	Programming Languages	1.00
CSCI 333	Theory of Computation	1.00
Mathematical Biology		
CSCI 125	Computer Science for Scientists and Mathematicians	1.00
CSCI 251	Software Design and Implementation	1.00
CSCI 315	Bioinformatics	1.00
Neuroscience		
CSCI 253	Algorithms and Data Structures	1.00
CSCI 315	Bioinformatics	1.00
CSCI 333	Theory of Computation	1.00
Statistics		
CSCI 125	Computer Science for Scientists and Mathematicians	1.00

Check with the relevant programs for limitations and details.

Also certain courses in other departments can count toward a computer science major:

Code	Title	Credits
ID 259	HiPerCiC: Collaborative Web Applications	1.00
MATH 232	Introduction to Mathematical Reasoning	1.00
MATH 244	Real Analysis I	1.00
MATH 252	Abstract Algebra I	1.00

PHYS 130	Analytical Physics I	1.00
PHYS 246	Electronics	1.00

Check with the computer science program for limitations and details

The computer science program offers many opportunities to participate in undergraduate research, often integrated within courses, or through extracurricular activities such as summer research. Ongoing efforts in high-performance cluster and parallel computing, graphics, 3D computer vision, interdisciplinary web applications, and declarative approaches to language design provide a foundation for many student projects.

Collaborative interdisciplinary projects apply computer science to many fields across campus, including environmental studies, archaeology, management studies, music, physics, linguistics, and history.

Recommendations for Graduate Study

Students considering graduate study in computer science should pursue opportunities that add both breadth and depth in their majors. Graduate-school-bound students are strongly encouraged to pursue undergraduate research involving computer science, and to take courses beyond the minimal major requirements.

Requirements

A student arranges for a computer science major by individual contract with the computer science faculty. This provides some latitude for choice according to individual interests and background and allows the computer science faculty to update the curriculum easily as the field of computer science evolves. Most contracts adhere to the guidelines below, which derive from prominent national recommendations for undergraduate computer science majors.

Code	Title	Credits
Foundation courses		
CSCI 121	Principles of Computer Science	1.00
or CSCI 125	Computer Science for Scientists and Mathematicians	
CSCI 241	Hardware Design	1.00
CSCI 251 & CSCI 252	Software Design and Implementation and Software Design and Implementation Lab (0.25)	1.25
Select one of the following proof-writing courses:		
MATH 232	Introduction to Mathematical Reasoning	
MATH 252	Abstract Algebra I	
MATH 244	Real Analysis I	
Core courses		
CSCI 253	Algorithms and Data Structures	1.00
CSCI 263	Ethical Issues in Software Design	1.00
Select one of the following:		
CSCI 276	Programming Languages	
CSCI 333	Theory of Computation	
CSCI 336	Logic Programming	
Select one of the following:		1.00
CSCI 273	Operating Systems	
CSCI 284	Mobile Computing Applications	
CSCI 300	Topics in Computer Science (with parallel and distributed computing)	
Electives and capstone		
CSCI 390	Senior Capstone Seminar	1.00
Two approved electives (possibly including ID 259)		2.00
Total Credits		9.25

These guidelines represent four levels:

- choice of introductory courses (CSCI 121 or CSCI 125) and three “second courses” (CSCI 241, CSCI 251, and a proof-writing course) that may be taken in any order (note that the foundation courses offer non-majors a variety of one- to four-course samples of computer science);
- core courses in standard computer science topics, including algorithms and data structures, ethics of computing, computer languages, and computer systems;
- a deeper exposure to selected aspects of the discipline of computer science through electives; and
- a senior-level capstone integrative experience (CSCI 390).

Courses

Courses in computer science satisfy the following general education requirements: AQR, WRI, ORC, IST, and EIN. See the Class and Lab Schedule (https://www.stolaf.edu/sis/public-aclasslab.cfm) for details. No more than one of the two introductory courses, CSCI 121 and CSCI 125 may be taken for credit toward the major.

CSCI 121: *Principles of Computer Science*
This course introduces students to computer science (CS), a field devoted to creative problem solving with computers, and its applications to other disciplines. Students explore fundamental concepts, including iteration, recursion, object-oriented software design, algorithm efficiency, levels of naming, parallel computing, and computing ethics. Students apply these concepts daily in hands-on homework exercises relevant to fields in the arts, humanities (including digital humanities computations), social sciences, and natural sciences. Includes a team project applying CS to a chosen discipline. No prior experience with programming is expected or required. Offered each semester. One of CSCI 121, CSCI 125, or CSCI 251 counts toward linguistic studies concentration.

CSCI 125: *Computer Science for Scientists and Mathematicians*
This course focuses on handling data: visualization, finding patterns, and communicating with data. The primary tools are Python (for transforming data) and R (for visualization and classification). Students work individually and in teams to apply basic principles and explore real-world datasets with a sustainability theme. Counts toward statistics and mathematical biology concentrations. One of CSCI 121, CSCI 125, or CSCI 251 counts toward linguistic studies concentration. Offered each semester.
Prerequisite: calculus or consent of the instructor.

CSCI 241: *Hardware Design*
This course explores modern computer hardware, emphasizing trade-offs among architectural choices and examining representative examples. Programming exercises explore how these topics relate to five high-level programming languages. Topics include machines, overview of computer organization, the design of processors including hardware, forms of parallelism, machine-level implementation of programming language features, memory organization, digital logic, microprogrammed and RISC architectures, multi-core architectures, performance enhancements, assembly programming, and architecture of networks and their protocols. Offered annually. Counts toward management studies concentration.
Prerequisite: CSCI 121 or CSCI 125 or consent of the instructor.

CSCI 251: *Software Design and Implementation*
This course provides an introduction to the structure and creation of computer software, using the C++ programming language and emphasizing object-oriented programming and structured collaborative software-development methodology. Concepts and skills are applied in a substantial multi-week team project. Topics include object-oriented programming, specification, programmed memory management, indirect addressing, tools including UML, team software process, requirements analysis, software design strategies, and elementary ethical analysis of software systems. Counts toward mathematical biology and management studies concentrations; one of CSCI 121, CSCI 125, CSCI 251 counts toward linguistic studies concentration. Offered each semester.
Prerequisite: CSCI 121 or CSCI 125 or CSCI 241 or PHYS 130 or MATH 252 or permission of instructor; concurrent registration in CSCI 252 is required.

CSCI 252: *Software Design and Implementation Lab (0.25)*
Students investigate the implementation of software using strategies and concepts presented in CSCI 251, explore standard technologies for creation and management of multi-module software systems, and carry out stages of structured team software project, through hands-on computational exercises and with direct support provided in a small group context. Offered each semester.
Prerequisite: concurrent registration in CSCI 251.

CSCI 253: *Algorithms and Data Structures*
This course surveys standard algorithms and data structures with emphasis on implementation experience and complexity analysis. Topics include algorithmic strategies, fundamental computer algorithms, stacks, queues, lists, trees, hash tables, heaps and priority queues, compression, and decompression. Offered annually. Counts toward neuroscience concentration.
Prerequisites: MATH 232 (MATH 244 or MATH 252) and CSCI 251, or consent of the instructor.

ID 259: *HiPerCiC: Collaborative Web Applications*
Computer science (CS) students team with students in non-CS fields in order to create custom web-based software that serves the research needs of particular professors in those non-CS fields of application. Students in a field of application provide disciplinary content, context, and feedback throughout project development, while pursuing relevant independent work. CS students develop software collaboratively and rapidly using the HiPerCiC (High-Performance Computing in Context) framework, while learning current web programming principles and technologies. Counts toward computer science major.
Prerequisite: CSCI 251 or permission of instructor.

CSCI 263: *Ethical Issues in Software Design*
The software we design has real effects in people's lives. This course explores the ethical and social considerations inherent in computer-based systems, develops skills in thinking about those considerations and in collecting data to determine their effects, and expands students' abilities to integrate these issues and skills into software development procedures, largely through an extensive team analysis of a "live" software project. Coursework uses extended case studies and surveys topics such as professional and ethical responsibilities, risk, liability, intellectual property, privacy, and computer crime. Offered annually. Counts toward management studies concentration.
Prerequisites: CSCI 251 and completion of BTS-T, or permission of instructor.

CSCI 273: *Operating Systems*
This course examines the features of modern operating systems, including detailed consideration of Linux and other example systems. Projects range from system-level programming and multithreaded network programming to kernel modifications. Topics include operating system principles, implementation as system calls, process scheduling and dispatch, concurrency, inter-process communication, programming with threads and sockets, low-level memory management, device management, file systems, security and protection mechanisms, virtual machines, and kernel programming. Offered alternate years. Counts toward management studies concentration.
Prerequisites: completion of or concurrent enrollment in CSCI 241 and CSCI 251, or permission of instructor.

CSCI 276: *Programming Languages*
Students study features commonly found in computer programming languages and construct their own interpreters for an example programming language incorporating various language features they study throughout the course. Topics include programming language semantics, programming language translation, parsing, implementation of control structures and memory structures, abstraction mechanisms, and language translation systems and types. Offered alternate years. Counts toward linguistic studies and management studies concentrations.
Prerequisites: CSCI 241 and CSCI 251, or permission of instructor.

CSCI 284: *Mobile Computing Applications*
Mobile devices are actually sophisticated and powerful computers. This course explores mobile computing technology by creating applications for the Android platform, including a final team project. The course introduces Java language and provides exposure to graphics user interfaces (GUIs), event-driven programming, APIs, databases, SQL query language, and agile team programming methodologies. Offered alternate years .Counts toward management studies concentration.
Prerequisite: CSCI 251 or permission of instructor.

CSCI 294: *Academic Internship*
This is an intermediate-level version of CSCI 394.

CSCI 298: *Independent Study*

CSCI 300: *Topics in Computer Science*
Recent and planned topics include parallel and distributed computing, mobile computer graphics, and relational database systems. May be repeated if topic is different. Offered alternate years.

CSCI 315: ***Bioinformatics***
Students study computational problems arising from the need to store, access, transform, and utilize DNA-related data. Topics from computer science include: exhaustive search; algorithms (including dynamic programming, divide-and-conquer, graph and greedy algorithms) for fragment reassembly, sequence alignment, phylogenetic trees; combinatorial pattern matching; clustering and trees; and hidden Markov models. Offered alternate years. Counts toward neuroscience, biomolecular studies, and mathematical biology concentrations and mathematics major.
Prerequisites: CSCI 253, or one of CSCI 121 or CSCI 125, and one of BIO 125 or MATH 220, or permission of instructor.

CSCI 333: ***Theory of Computation***
Students learn about formal languages, automata, and other topics concerned with the theoretical basis and limitations of computation. The course covers automata theory including regular languages and context-free languages, computability theory, complexity theory including classes P and NP, and cryptographic algorithms. Offered alternate years. Counts toward neuroscience and linguistic studies concentrations.
Prerequisite: a proof writing course (such as MATH 244 or MATH 252) or permission of instructor.

CSCI 336: ***Logic Programming***
Students learn a widely-used style of programming based on first order predicate logic. Topics include declarative programming, Horn clauses, declarative and procedural semantics of logic programs, clauses as relations, goals, backtracking, and resolution. Programming projects and exercises use Prolog, the most significant logic programming language. Additional topics include the relationship of Prolog to logic and applications to artificial intelligence. Offered alternate years.
Prerequisite: CSCI 251 or permission of instructor.

MSCS 341: ***Algorithms for Decision Making***
This course introduces students to the subject of machine learning. The primary focus is the development and application of powerful machine learning algorithms applied to complex, real-world data. Topics covered include linear regression, nearest neighbor models, k-means clustering, shrinkage methods, decision trees and forests, boosting, bagging, support vector machines, and hierarchical clustering. Applications are taken from a wide variety of disciplines, including biology, economics, public policy, public health, and sports. Offered on a regular basis. Familiarity with a programming language such as R or Python is highly recommended.
Prerequisite: MATH 220, CSCI 251, or STAT 272 or permission of the instructor.

CSCI 350: ***Advanced Team Project***
This course is devoted to team research/development projects, employing established software development methodologies. Projects frequently have an interdisciplinary nature, involving consultation with faculty and/or students in other fields, and taking advantage of the particular backgrounds of team members. One or more research papers, posters ,etc., on results are ordinarily expected. Offered alternate years.
Prerequisites: one core course with implementation in computer science or permission of instructor.

MSCS 389: ***Math, Statistics, and Computer Science Research Methods (0.50)***
Students focus on writing scientific papers, preparing scientific posters, and giving presentations in the context of a specific, year-long, interdisciplinary research project. In addition, this weekly seminar series builds collaborative research skills such as working in teams, performing reviews of math, statistics, and computer science literature, consulting effectively, and communicating proficiently. Exposure to post-graduate opportunities in math, statistics, and computer science disciplines is also provided. Open to students accepted into the Center for Interdisciplinary Research.

CSCI 390: ***Senior Capstone Seminar***
Class members participate in undergraduate research, including readings from the research literature, team development of project software, ethical analysis of their project applying CSCI 263 principles, documentation practices, and writing a research paper for public presentation. Offered annually.
Prerequisites: major in computer science with senior standing, and completion of or concurrent enrollment in computer science core courses, ordinarily including CSCI 253 and CSCI 263, or permission of instructor.

CSCI 394: ***Academic Internship***
Students gain experience in computer-industry positions. Recent projects have included contributions to team programming, documentation, business applications of computing, modifications of large proprietary code bases using industry-standard software frameworks, test-driven development, creating mobile-computing interfaces for existing software systems, and improving security of company websites. Internship experiences (whether for credit or not) are strongly encouraged for anyone considering a career in computing.

CSCI 396: ***Directed Undergraduate Research***
This course provides a comprehensive research opportunity, including an introduction to relevant background material, technical instruction, identification of a meaningful project, and data collection. The topic is determined by the faculty member in charge of the course and may relate to his/her research interests. Offered based on department decision. May be offered as a 1.00 credit course or .50 credit course.
Prerequisite: determined by individual instructor.

CSCI 398: ***Independent Research***
Recent projects, usually executed by individuals, but occasionally in small groups, and coordinated with ongoing undergraduate research projects, include cluster-assisted computer vision for robots, parallel computing incomputer science education, and middleware for interdisciplinary web applications.

Faculty

Director, 2017-2018
Richard A. Brown
Professor of Mathematics, Statistics, and Computer Science
computer science; parallel/distributed systems

Richard J. Allen
Professor of Mathematics, Statistics, and Computer Science
logic programming; intelligent tutoring systems; computer science; Islamic geometric patterns; bioinformatics

Olaf A. Hall-Holt (on leave)
Associate Professor of Mathematics, Statistics, and Computer Science
computational geometry; computer vision/graphics

Charles Huff
Professor of Psychology
social psychology; moral psychology; science and technology studies; psychology of religion

Steven McKelvey
Professor of Mathematics, Statistics, and Computer Science, Registrar and Assistant Vice President for Academic Affairs
operations research; wildlife modeling

Matthew P. Richey
Professor of Mathematics, Statistics, and Computer Science
applied and computational mathematics