COMPUTER SCIENCE

Melissa Schori, Regents Math 307
507-786-3113
schori1@stolaf.edu
wp.stolaf.edu/cs

The Department of Mathematics, Statistics, and Computer Science offers majors in all three disciplines, including mathematics, computer science, and statistics and data science.

Computer science is the academic discipline that focuses on creative computing-related problem solving. St. Olaf's Computer Science 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 Computer Science 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 Computer Science differs from other areas of computing, such as Information 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 Computer Science 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 Computer Science concepts relevant in almost every setting. The Computer Science 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 Computer Science 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 Computer Science 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.

Intended Learning Outcomes for the Major

Distinction

See Academic Honors

Other Fields

Certain courses in computer science (as shown below) can count toward other majors and concentrations.

Majors

Code	Title	Credits
Mathematics		
CSCI 333	Theory of Computation	1.00
CSCI 353	Analysis of Algorithms	1.00
Statistics and Data Science		
CSCI 379	Foundations of Artificial Intelligence	1.00

Concentrations

Code	Title	Credits
Linguistics		
CSCI 121	Principles of Computer Science	1.00
CSCI 251	Software Design and Implementation	1.00
CSCI 276	Programming Languages	1.00
CSCI 333	Theory of Computation	1.00
CSCI 379	Foundations of Artificial Intelligence	1.00
Mathematical Biology		
CSCI 251	Software Design and Implementation	1.00
Neuroscience		
CSCI 121	Principles of Computer Science	1.00
CSCI 353	Analysis of Algorithms	1.00

Check with the relevant programs for limitations and details.

Also, certain courses in other programs/departments are required or can count toward a computer science major:

Code	Title	Credits
MATH 220	Elementary Linear Algebra	1.00
MATH 234	Discrete Mathematical Reasoning	1.00
MATH 261	Computational Geometry	1.00
PHYS 130	Analytical Physics I	1.00
PHYS 246	Electronics	1.00
SDS 333	High-Dimensional Data Analysis	1.00
SDS 341	Algorithms for Decision Making	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, business and management studies, music, physics, applied 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

Requirements for the Major

A student arranges for a computer science major by individual contract (called a CSMaP) with a computer science faculty member. 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. Students are strongly advised to complete at least one of CSCI 241 or 251 no later than the spring of their sophomore year.

Code	Title	Credits
Required Computer Science courses:		
CSCI 121	Principles of Computer Science	1.00
or PHYS 130	Analytical Physics I	
or prior experience approved by the program director		
CSCI 221	Introduction to Data Structures in C++	1.00
CSCI 241	Hardware Design	1.00
CSCI 251	Software Design and Implementation	1.00
CSCI 263	Ethical Issues in Software Design	1.00
CSCI 353	Analysis of Algorithms	1.00
Required Mathematics courses:		
MATH 220	Elementary Linear Algebra	1.00
MATH 234	Discrete Mathematical Reasoning	1.00
Three elective upper-level courses [1]		**3**
Select at least two of the following designated courses		
CSCI 273	Operating Systems	
CSCI 276	Programming Languages	
CSCI 333	Theory of Computation	
CSCI 379	Foundations of Artificial Intelligence	
Other electives include		
CSCI 200	Topics in Computer Science	
CSCI 284	Mobile Computing Applications	
CSCI 300	Topics in Computer Science	
CSCI 356	Parallel and Distributed Computing	
CSCI 390	Senior Capstone Seminar	
or CSCI 391	Ghana International Capstone (study abroad)	
MATH 261	Computational Geometry	
PHYS 246	Electronics	
SDS 333	High-Dimensional Data Analysis	
SDS 341	Algorithms for Decision Making	
Total Credits		**11**

1

At least one elective must be at level 300.

**By completing this major, the student also satisfies the OLE Core Writing in the Major requirement.*

Courses

Courses in computer science satisfy the following OLE Core curriculum requirements: Quantitative and Computational Reasoning and Ethical Reasoning in Context. See the Class and Lab Schedule for details.

No more than one of the two introductory courses, CSCI 121 and the previously offered 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, 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 applied linguistics concentration.

CSCI 200: *Topics in Computer Science*
Students explore special topics in computer science. Topics may vary from year-to-year. May be repeated if topics are different. Offered periodically.
Prerequisite: permission of instructor.

CSCI 201: *Topics in Computer Science (0.50)*
Students explore special topics in computer science. Topics may vary from year-to-year. The workload in this course is commensurate with a 0.50-credit course. May be repeated if topics are different. Offered periodically.
Prerequisite: permission of instructor.

CSCI 202: *Topics in Computer Science (0.25)*
Students explore special topics in computer science. Topics may vary from year-to-year. The workload in this course is commensurate with a 0.25-credit course. May be repeated if topics are different. Offered periodically.
Prerequisite: permission of instructor.

CSCI 221: *Introduction to Data Structures in C++*
This course introduces students to the C++ programming language and common data structures. Students develop their understanding of how machines store and operate on data, down to the individual bits. Students learn about storage space and run-time requirements of common data structures, including stacks, queues, linked lists, and trees. Combined with their deeper understanding of how the computer processes data, students analyze programs for efficiency and to apply the right data structure for specific scenarios. Offered each semester.
Prerequisite: CSCI 121 or CSCI 125 or PHYS 130 or permission of instructor.

CSCI 241: *Hardware Design*
This course explores computer hardware and how the code we write uses it, taught through in-class labs and daily assignments. Topics include an introduction to computer systems, computer organization and architecture, the component-level design of computer hardware, forms of parallelism, machine-level implementation of programming language features, memory organization, logic circuits, data representation, assembly programming, and a brief introduction to networking and cloud computing. Offered each semester.
Prerequisite: CSCI 221 or permission of 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 team software process, software design strategies, and elementary ethical analysis of software systems. Offered each semester.
Prerequisite: CSCI 221 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. Coursework uses case studies and surveys topics such as professional and ethical responsibilities, risk, liability, intellectual property, privacy, computer crime, and AI ethics. Offered each semester.
Prerequisite: CSCI 251 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 periodically.
Prerequisites: completion of or concurrent enrollment in CSCI 241 and CSCI 251, or permission of instructor.

CSCI 276: *Programming Languages*
The course begins with a survey of several popular programming languages, learning to write some code in each of them. We then consider important language features that have been used to describe entire categories of languages. Topics include programming language semantics, programming language translation, parsing, memory structures, abstraction mechanisms, and language translation systems and types. Also counts toward applied linguistics concentration.
Prerequisites: 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 periodically.
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 periodically.

CSCI 301: *Advanced Topics in Computer Science (0.50)*
Students explore special topics in computer science at a level commensurate with other Level III courses in Computer Science, and with appropriate prior experience in Computer Science topics. The workload in this course is commensurate with a 0.50-credit course. Topics may vary from year-to-year. May be repeated if topics are different. Offered periodically.
Prerequisite: 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 periodically.
Prerequisite: a proof writing course (such as MATH 234, MATH 244, or MATH 252) and computer science major or permission of instructor.

CSCI 353: *Analysis of Algorithms*
This course surveys standard topics in the study of algorithms, with an emphasis on complexity analysis and implementation experience. Topics include asymptotic analysis, searching and sorting, divide and conquer, basic graph algorithms, greedy algorithms, dynamic programming, and NP-complete problems. Offered each semester.
Prerequisites: MATH 234 and CSCI 251, or consent of the instructor.

CSCI 356: *Parallel and Distributed Computing*
Parallel and Distributed Computing (PDC) is all around us. Virtually all computers, from cell phones to powerful servers, feature multicore Parallel Computing; and clouds of networked remote Distributed Computing systems provide familiar services from web search to email to online shopping. This course presents PDC concepts and applications through hands-on experiences with widely used PDC technologies including OpenMP, C++-11 threads, MPI, and Hadoop, and develops effective PDC problem-solving skills through industry-proven parallel programming patterns. Offered periodically.
Prerequisite: CSCI 241 and CSCI 251.

CSCI 379: *Foundations of Artificial Intelligence*
Artificial intelligence is an extremely broad field in which the overarching goal is the creation of an autonomous agent with human-level capabilities. Students study the fundamental algorithms and techniques used to create agents with varying degrees of autonomy and function, including search algorithms, first-order logic, Bayesian networks and simple neural networks. Students learn how to represent problems for specific techniques, and how to select the best technique for a given problem. Offered annually.
Prerequisites: CSCI 251 and MATH 234, or permission of instructor.

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 353 and CSCI 263, or permission of instructor.

CSCI 391: *Ghana International Capstone (study abroad)*
Class members participate in a team research project abroad. The central activity of the project teams is to implement an applied research project in collaboration with a host community in West Africa. This research project drives the other elements of the course, which include readings from the research literature, ethical analysis of their project applying CSCI 263 principles, documentation practices, and writing a research paper for public presentation. Offered periodically during January Term. Apply through the Smith Center for Global Engagement.
Prerequisites: computer science major with senior standing and completion of or concurrent enrollment in computer science core courses, ordinarily including CSCI 353 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 their 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 in computer science education, and middleware for interdisciplinary web applications.

SDS 333: *High-Dimensional Data Analysis*
Data with thousands of dimensions or columns is commonplace across multiple disciplines, including image analysis, natural language processing, genomics, time series, and recommendation systems. In this class students learn, understand, and apply principles and techniques to gain insights from high-dimensional data. Topics include clustering, principal component analysis, multidimensional scaling, non-negative matrix factorization, missing-value imputation for recommendation systems, topic modeling, word embeddings, object recognition using "eigenfaces", non-linear embedding techniques, multiple hypothesis testing, and false discovery rate adjustments. Offered periodically during fall and spring semesters.
Prerequisites: MATH 220 and SDS 164.

SDS 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 annually during fall and spring semester.
Prerequisite: SDS 164 or SDS 264 or permission of the instructor.

Plan of Study

Computer Science Major - Plan of Study

This is a sample plan that meets the prescribed requirements for the Computer Science major at St. Olaf. This tool is meant as a guide and does not replace working closely with the student's academic advisor.

Course	Title	Credits
First Year		
Fall Semester		
FYS 120 or WRIT 120	First-Year Seminar (or Conversation Program) or Writing and Rhetoric	1.00
CSCI 121 or PHYS 130	Principles of Computer Science or Analytical Physics I	1.00
MATH 220	Elementary Linear Algebra (if credit already given for Calculus I)	1.00
World Language		1.00
	Credits	**4**
Spring Semester		
WRIT 120 or FYS 120	Writing and Rhetoric (or Conversation Program) or First-Year Seminar	1.00
CSCI 221	Introduction to Data Structures in C++	1.00

World Language		1.00
	Credits	**3**
Sophomore Year		
Fall Semester		
CSCI 251	Software Design and Implementation	1.00
MATH 220	Elementary Linear Algebra (if not already taken in first year)	1.00
	Credits	**2**
Spring Semester		
CSCI 241	Hardware Design	1.00
MATH 234	Discrete Mathematical Reasoning	1.00
	Credits	**2**
Junior Year		
Fall Semester		
CSCI 263	Ethical Issues in Software Design	1.00
	Credits	**1**
Spring Semester		
CSCI 353	Analysis of Algorithms	1.00
	Credits	**1**
Senior Year		
Fall Semester		
CSCI 273 or CSCI 276 or CSCI 333 or CSCI 379	Operating Systems (take two of the following if possible in the fall semester) or Programming Languages or Theory of Computation or Foundations of Artificial Intelligence	1.00
	Credits	**1**
Spring Semester		
CSCI 273 or CSCI 276 or CSCI 333 or CSCI 379	Operating Systems or Programming Languages or Theory of Computation or Foundations of Artificial Intelligence	1.00
CSCI 300-Level Elective Course		1.00
	Credits	**2**
	Total Credits	**16**

Students must successfully complete the equivalent of 35 St. Olaf credits through a combination of full-credit and fractional-credit courses to earn the Bachelor of Arts.

Visit the Computer Science program webpage for more information.

Faculty

Program Director, 2025-2026
Ryota Matsuura
Professor of Mathematics, Statistics, and Computer Science
mathematics education; algebraic number theory

Jaime I. Davila
Associate Professor of Mathematics, Statistics, and Computer Science

Charles Fyfe
Adjunct Assistant Professor of Mathematics, Statistics and Computer Science

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

Michael Haydock
Adjunct Professor of Mathematics, Statistics, and Computer Science.

Barbara Johnson
Visiting Assistant Professor of Mathematics, Statistics and Computer Science

William (Will) Leeson
Assistant Professor of Mathematics, Statistics, and Computer Science

Melissa Lynn
Assistant Professor of Mathematics, Statistics, and Computer Science

Kimberly (Kim) Mandery
Computer Scientist in Residence

Matthew Wright
Associate Professor of Mathematics, Statistics, and Computer Science; Assistant Director of External Fellowships
applied and computational topology