Computer Science

Reza Rejaie, Department Head
541-346-4408
541-346-5373 fax
120 Deschutes Hall
1202 University of Oregon
Eugene OR 97403-1202

Computer science, the study of computation, offers students the challenge and excitement of a dynamically evolving science whose discoveries and applications affect every aspect of modern life. Computer science is a rich intellectual field where practitioners apply a computational approach to address a wide variety of interesting and challenging problems. Computer scientists are engaged in research in core areas of theoretical computer science, computer systems design, algorithms, and programming languages, as well as more application-oriented areas such as databases and networking.

The Department of Computer Science (CS) is committed to a strong research program and a rewarding educational experience for undergraduate and graduate students.

The department offers instruction and opportunities for research in the following areas:

- Artificial Intelligence
- Assistive Technology
- Computational Science
- Computer Vision
- Cyber Security and Privacy
- High-Performance Computing
- Human-Computer Interaction
- Machine Learning
- Natural Language Processing
- Networking and Systems
- Parallel and Distributed Computing
- Performance Analysis
- Programming Languages and Compilers
- Scientific Visualization
- Software Engineering
- Theoretical Computer Science

The department offers bachelor's, master's and doctoral degrees; in addition, two undergraduate minors and a selection of service courses are offered for students who want introductory exposure to computers and computer applications. The computer science programs at the university are continually evolving as the discipline matures and as students' needs change.

Facilities

The Department of Computer Science is housed in Deschutes Hall, which holds faculty and graduate student offices and extensive laboratory space for research and instruction.

Undergraduate majors may use campus computing labs staffed by CS undergraduate tutors and lab assistants. Undergraduate majors taking upper-division courses and graduate students share a collaborative computing lab for the exclusive use of CS students. Graduate and undergraduate students engaged in active research also have access to the computing facilities of the associated research lab.

The cognitive modeling and eye-tracking laboratory features multiple Eyegaze eye trackers and a Tobii eye tracker, used to collect and analyze the eye movements people make during human-computer interactions, and to develop eye-controlled user interfaces for people with disabilities.

Research in high-performance computing and computational science is supported by resources in the Oregon Advanced Computing Institute for Science and Society (OACISS) and the University of Oregon's Research Advanced Computing Services. The Talapas cluster provides a large-scale computational and storage resource to support research computing across the university.

The Advanced Integration and Mining Laboratory fosters research on discovering useful patterns from the mountain of data on biology, health, medicine, neuroscience, physiology, and social networks and on integrating data from structurally and semantically heterogeneous resources such as databases, online social networks, and the World Wide Web.

Oregon Networking Research Group (ONRG) conducts cutting-edge research in several areas of computer networking and networked systems including Internet measurements, multi-cloud computing, network telemetry systems, machine learning for networking, network security, social computing, and programmable optics. Research at ONRG is currently funded by federal agencies (e.g., NSF, NIH, etc.), industry (e.g., Ripple, Broadcom, Cisco, etc.), and foundations (e.g., Internet Society Foundation).

The Network & Security Research Laboratory features hardware and software facilities devoted to experimentation, simulation, and analysis of various computer networking techniques (such as Internet routing, software-defined networking, online social networking, and Internet of things), malicious network attacks (such as distributed denial-of-service attacks, traffic hijacking, Internet worms, botnets, social bots, phishing), and network defense solutions (such as firewalls, anti-phishing solutions, distributed denial-of-service defense, IP spoofing prevention, Internet routing security, Internet privacy protection, and Internet of things security and privacy).

The Research Group on Computing and Data-Understanding at Extreme Scale (CDUX) pursues problems in scientific visualization, high-performance computing, scientific computing, and computer graphics, and especially focuses on problems where these areas intersect. The group performs research for the Department of Energy, the National Science Foundation, and private companies, delivered in widely used software tools such as the VisIt visualization tool, and helps develop new tools, like VTK-m, a library for many-core visualization and analysis.

The High-Performance Computing Laboratory conducts research in several areas, including optimizing compilers, performance modeling and optimization, parallel algorithms, and software engineering. Example projects include static and dynamic analysis of software for building application performance models, ensuring software quality, or detecting security vulnerabilities; using machine learning and other approaches to model run-time characteristics of software; developing data mining techniques to study and improve HPC software engineering processes; applying natural language processing methods to study and improve HPC software developer productivity; designing new algorithms or

improving existing ones in several application areas, including large-scale dynamic graphs, computational physics, and computational biology.

In addition, the university is a member of Internet2, a high-speed network connecting major research institutions.

Honors Program

The Computer Science department offers an honors program to their undergraduate majors. After obtaining advance approval, students in the degree program are eligible to attain honors in computer science by meeting the honors requirements of the department, including writing a thesis.

Careers

The undergraduate program is designed to prepare students for professional careers or graduate study. The field of computer science, which has become increasingly interdisciplinary over the past decade, offers a rich array of opportunities in fields as disparate as medicine, manufacturing, and the media as well as the computer industry.

Graduates come away with confidence that they can specify, design, and build large software systems; analyze the effectiveness of computing techniques for a specific problem; and work effectively in problem-solving teams. The master of arts (MA) and master of science (MS) degree programs prepare students for higher-level positions in the areas described above as well as for teaching positions in community colleges. The PhD degree program trains students as scientists for advanced research in specialized areas of computer science and for teaching in universities.

Faculty

Zena M. Ariola, professor (programming languages, lambda calculus, logic). BS, 1980, Pisa; PhD, 1992, Harvard. (1992)

Hank Childs, professor (scientific visualization, high-performance computing, computer graphics). BS, 1999, PhD, 2006, California, Davis. (2013)

Jee W. Choi, assistant professor (high-performance computing, scientific computation, data analytics). BS, 2000, PhD, 2015, Georgia Institute of Technology. (2019)

Phil Colbert, instructor (education, healthcare, environmental sciences, IoT, full stack development, CIT Minor). BS, 1990 CSU Chico; MS, 2008, CSU Chico. (2014)

Dejing Dou, professor (artificial intelligence, data mining, natural language processing). BE, 1996, Tsinghua; MS, 2000, PhD, 2004, Yale. (2004)

Brittany Erickson, assistant professor (scientific computing, computational seismology, high performance computing). BS, 2004, MS, 2006, PhD, 2010, California, Santa Barbara. (2018)

Ramakrishnan Durairajan, assistant professor (computer networking, multi-cloud computing, Internet data science, programmable optics, cybersecurity). BS, 2010, College of Engineering, Guindy; MS, 2014, PhD, 2017, Wisconsin, Madison. (2017)

Stephen F. Fickas, professor (software engineering, formal modeling of distributes systems, digital humanities). BS, 1971, Oregon State; MS, 1973, Massachusetts; PhD, 1983, California, Irvine. (1983)

Kathleen Freeman Hennessy, senior instructor; director of undergraduate studies. BS, 1982, Bucknell; PhD, 1993, Oregon. (2011)

Anthony J. Hornof, professor (human-computer interaction, assistive technology, cognitive modeling). BA, 1988, Columbia; MS, 1996, PhD, 1999, Michigan. (1999)

Lei Jiao, assistant professor (networking and distributed computing, performance modeling and evaluation, algorithm design and application). BS, 2007, MS, 2010, Northwestern Polytechnical; PhD, 2014, Göttingen. (2016)

Jun Li, professor (computer and network security, network architectures and protocols, distributed systems). BS, 1992, Peking; ME, 1995, Chinese Academy of Sciences; MS, 1998, PhD, 2002, California, Los Angeles. (2002)

Yingjiu (Joe) Li, professor (cyber security and privacy). BS, 1990, Dalian University of Technology; PhD, 2003, George Mason University. (2019)

Daniel Lowd, associate professor (machine learning, data mining, artificial intelligence). BS, 2003, Harvey Mudd College; MS, 2005, PhD, 2010, Washington (Seattle). (2009)

Allen D. Malony, professor (parallel processing, performance evaluation, neuroinformatics). BS, 1980, MS, 1982, California, Los Angeles; PhD, 1990, Illinois, Urbana-Champaign. (1991)

Thanh H. Nguyen, assistant professor (artificial intelligence, multi-agent systems, machine learning). BS, 2010, Hanoi University of Science and Technology; PhD, 2016, University of Southern California. (2018)

Thien Huu Nguyen, assistant professor (natural language processing, data mining, machine learning, artificial intelligence). BE, 2011, HUST, Hanoi; MS, 2014, PhD, 2018, NYU, New York. (2018)

Boyana Norris, associate professor (high-performance computing, compilers, performance analysis and optimization, software engineering, data science). BS, 1995, Wake Forest; PhD, 2000, Illinois, Urbana-Champaign. (2013)

Reza Rejaie, professor (computer networks, networked systems, Internet measurement). BS, 1991, Sharif University of Technology; MS, 1996, PhD 1999, Southern California. (2002)

Humphrey Shi, assistant professor (Computer Vision, Machine Learning, AI Systems & Applications). BS, 2005, Tsinghua University; PhD, 2017, University of Illinois at Urbana-Champaign. (2019)

Joseph Sventek, professor (complex event processing, Internet of Things). BS, 1973, Rochester; PhD, 1979, California, Berkeley. (2014)

Dave Wilkins, instructor. BA, 1965, Whitman College; MS, 1971, Oregon. (2010)

Eric D. Wills, senior instructor. BS, 2000, MS, 2002, PhD, 2008, Oregon. (2010)

Christopher B. Wilson, associate professor (computational complexity, models of computation). BS, 1978, Oregon; MS, 1980, PhD, 1984, Toronto. (1984)

Michal Young, associate professor (software engineering, software test and analysis). BA, 1983, Oregon; MS, 1985, PhD, 1989, California, Irvine. (1997)

Emeriti

Sarah A. Douglas, professor emerita. AB, 1966, California, Berkeley; MS, 1979, PhD, 1983, Stanford. (1983)

Arthur M. Farley, professor emeritus. BS, 1968, Rensselaer Polytechnic Institute; PhD, 1974, Carnegie-Mellon. (1974)

Michael Hennessy, senior instructor. BS, 1982, MS, 1984, Oregon. (1984)

Virginia M. Lo, associate professor emerita. BA, 1969, Michigan; MS, 1977, Pennsylvania State; PhD, 1983, Illinois, Urbana-Champaign. (1984)

Eugene M. Luks, professor emeritus. BS, 1960, City University of New York, City College; PhD, 1966, Massachusetts Institute of Technology. (1983)

Andrzej Proskurowski, professor emeritus. MS, 1967, Warsaw University of Technology; PhD, 1974, Royal Institute of Technology, Stockholm. (1975)

Kent A. Stevens, professor emeritus. BS, 1969, MS, 1971, California, Los Angeles; PhD, 1979, Massachusetts Institute of Technology. (1982)

The date in parentheses at the end of each entry is the first year on the University of Oregon faculty.

Affiliated Faculty

Sameer Shende, research associate professor and director, Performance Research Laboratory (performance evaluation tools, programming models, software stack for HPC and AI/ML). BTech, 1991, IIT Bombay; MS, 1996, PhD 2001, University of Oregon. (2001)

Participating

Heidi Kaufman, English

Amy K. Lobben, geography

Marc Schlossberg, planning, public policy and management

McKay M. Sohlberg, special education and clinical sciences

Sanjay Srivastava, psychology

Jeffrey Stolet, music

Don M. Tucker, psychology

- **Bachelor of Arts**
- **Bachelor of Science**
- **Computer Science Minor**
- **Computer Information Technology Minor**

Undergraduate Studies

The Department of Computer Science offers a major and a minor in Computer Science (CS), a major in Mathematics and Computer science (MACS), and a minor in Computer Information Technology (CIT).

The Computer Science major is intended for students who want to study computers and computation with strong mathematical and scientific foundations. The Mathematics and Computer science major emphasizes formal and abstract problem solving complemented by computational methods and computer technologies. This program, administered jointly with the Department of Mathematics, is described in the **Mathematics and Computer Science** section of this catalog. Both of these majors lead to the Bachelor of Arts (BA) or Bachelor of Science (BS) degrees.

Students majoring in Computer Science may choose to focus their studies in one of several areas of specialization, or concentrations, which build on the standard Computer Science core requirements. Each concentration specifies a set of coordinated choices for fulfilling upper-division computer science and other elective requirements. Current Computer Science concentrations include AI and Machine Learning, Computer Networks, High Performance Computing/Computational Science, Security, and Software Development.

Preparation

High school students who plan to major in Computer Science should pursue a strong academic program, including substantial work in mathematics, the sciences, and writing. Courses in algebra, geometry, trigonometry, and more advanced topics should be included. Courses in computer programming or computer technology are useful but not required. Students who have taken AP or IB Computer Science courses should check the UO TES® Public View for equivalent transfer credits and score minimums. Upon arrival at the university, students should consult with an advisor to determine the entry-level course best suited to the student's background.

Transfer and Second Baccalaureate Students

Transfer and second baccalaureate students should consult the online Interactive Transfer Catalog as well as an advisor to determine whether computer science, mathematics, and science courses they have taken fulfill the major requirements. Completing only general-university requirements prior to transferring to the University of Oregon will not be sufficient preparation to complete a CS degree in two years.

Students attending community college in Oregon are encouraged to follow the Oregon Computer Science Major Transfer Map (MTM) or obtain the associate of arts Oregon transfer degree or the associate of science Oregon transfer degree in computer science before entering the University of Oregon. While earning an associate's degree, community college transfer students should take discrete mathematics and computer science. In addition, calculus and laboratory science are recommended.

Bachelor of Arts Degree Requirements

To earn a BA in computer science, majors must complete the requirements for a BS and also demonstrate proficiency in a second language. Computer science majors must complete at least 60 credits of CS courses, of which 24 must be earned in residence at the University of Oregon. In addition, majors must complete 28 credits in mathematics, 12 credits in the sciences, and 4 credits of technical or business writing. The specific requirements for the CS major fall into five categories: core courses, concentration and elective courses, mathematics, writing, and science.

Computer Science I (CS 210), Computer Science II (CS 211), Computer Science III (CS 212), Elements of Discrete Mathematics I (MATH 231), and Elements of Discrete Mathematics II (MATH 232) must be passed with grades of B– or better before students can take the upper-division core courses. Courses required for the major must be taken for a letter grade; upper-division electives in CS courses numbered 410 or higher

(12 credits) must also be taken for a letter grade. Upper-division courses must be passed with a grade of C– or better.

Code	Title	Credits
Core Courses: Lower Division		
CS 210–212	Computer Science I-III	12
MATH 231–232	Elements of Discrete Mathematics I-II	8
Core Courses: Upper Division		
CS 313	Intermediate Data Structures	4
CS 314	Computer Organization	4
CS 315	Intermediate Algorithms	4
CS 330	C/C++ and Unix	4
CS 415	Operating Systems	4
CS 422	Software Methodology I	4
CS 425	Principles of Programming Languages	4
Core Courses: Mathematics		
Select one of the following:		8
MATH 251–252	Calculus I-II	
MATH 261–262	Calculus with Theory I-II	
MATH 246–247	Calculus for the Biological Sciences I-II	
Select two of the following:		8
MATH 347	Fundamentals of Number Theory I	
or MATH 351	Elementary Numerical Analysis II	
or MATH 391	Fundamentals of Abstract Algebra I	
MATH 253	Calculus III	
or MATH 263	Calculus with Theory III	
MATH 341	Elementary Linear Algebra	
MATH 343	Statistical Models and Methods	
or MATH 425	Statistical Methods I	
or MATH 345H	Probability and Statistics for Data Science	
Core Courses: Science		
Select 12 credits from the following: [1]		12
Biology [2]		
CH 111	Introduction to Chemical Principles	
or CH 113	The Chemistry of Sustainability	
or CH 221	General Chemistry I	
or CH 224H	Advanced General Chemistry I	
BI 211,213	General Biology I,III	
or BI 211–212	General Biology I-II	
Chemistry [2]		
CH 221–223	General Chemistry	
or CH 224H-226H	Honors General Chemistry	
Earth Sciences		
ERTH 201	Dynamic Planet Earth	
ERTH 202	Earth's Surface and Environment	
ERTH 203	History of Life	
Geography		
GEOG 141	The Natural Environment	
Select two of the following:		
GEOG 321	Climatology	
GEOG 322	Geomorphology	
GEOG 323	Biogeography	
Physics [2,3]		
PHYS 201–203	General Physics	
or PHYS 251–253	Foundations of Physics I	
Psychology		
PSY 201	Mind and Brain	
Select two of the following:		
PSY 301	Scientific Thinking in Psychology	
PSY 304	Biopsychology	
PSY 305	Cognition	
PSY 348	Music and the Brain	
Core Course: Writing		
WR 320	Scientific and Technical Writing	4
or WR 321	Business Communications	
Electives: Upper Division		
Upper-division CS courses in student's chosen concentration (concentration information below)		12
Upper-division CS courses in student's chosen concentration, honors thesis, capstone project, or other upper-division courses [4,5]		8
Upper-division mathematics or theoretical computer science course [6]		4
Total Credits		**104**

1 To support interdisciplinary study, students in any concentration are encouraged to complete a minor (typically 24–32 credits) or major in a computing-related field. Students who complete a minor (other than computer information technology or mathematics) or another major (including mathematics) in a computing-related field may, with the approval of the Undergraduate Education Committee, replace the CS laboratory science requirement with the completed minor or major.

2 Students are encouraged to complete the accompanying lab courses.

3 Physics is recommended for networks concentration students.

4 If Experimental Course: [Topic] (CS 410) courses are applied, they must have different topic subtitles to satisfy this requirement.

5 A maximum of 8 credits in upper-division courses numbered less than 410, or DSCI 311. CS courses numbered 400–499 may be taken for a maximum of 4 credits when used to satisfy this requirement. Special Studies: [Topic] (CS 399), Seminar: [Topic] (CS 407), and Experimental Course: [Topic] (CS 410) courses must have different topic subtitles to satisfy this requirement. CS 399 and CS 410 courses must have a prerequisite of CS 313 and have regular weekly class meetings and homework assignments.

6 The mathematics elective is selected from upper-division mathematics courses with a prerequisite of Calculus II (MATH 252) or higher, or from theoretical computer science courses. A list of theoretical computer science courses is available in the computer science office or the department website.

Bachelor of Science Degree Requirements

To earn a BS in computer science, majors must complete at least 60 credits of CS courses, of which 24 must be earned in residence at the

University of Oregon. In addition, majors must complete 28 credits in mathematics, 12 credits in the sciences, and 4 credits of technical or business writing. The specific requirements for the CS major fall into five categories: core courses, concentration and elective courses, mathematics, writing, and science.

Computer Science I (CS 210), Computer Science II (CS 211), Computer Science III (CS 212), Elements of Discrete Mathematics I (MATH 231), and Elements of Discrete Mathematics II (MATH 232) must be passed with a grade of B– or better before students can take the upper-division core courses. Courses required for the major must be taken for a letter grade; upper-division electives in CS courses numbered 410 or higher (12 credits) must also be taken for a letter grade. Upper-division courses must be passed with a grade of C– or better.

Code	Title	Credits
Core Courses: Lower Division		
CS 210–212	Computer Science I-III	12
MATH 231–232	Elements of Discrete Mathematics I-II	8
Core Courses: Upper Division		
CS 313	Intermediate Data Structures	4
CS 314	Computer Organization	4
CS 315	Intermediate Algorithms	4
CS 330	C/C++ and Unix	4
CS 415	Operating Systems	4
CS 422	Software Methodology I	4
CS 425	Principles of Programming Languages	4
Core Courses: Mathematics		
Select one of the following:		8
MATH 251–252	Calculus I-II	
MATH 261–262	Calculus with Theory I-II	
MATH 246–247	Calculus for the Biological Sciences I-II	
Select two of the following:		8
MATH 347	Fundamentals of Number Theory I	
or MATH 351	Elementary Numerical Analysis II	
or MATH 391	Fundamentals of Abstract Algebra I	
MATH 253	Calculus III	
or MATH 263	Calculus with Theory III	
MATH 341	Elementary Linear Algebra	
MATH 343	Statistical Models and Methods	
or MATH 425	Statistical Methods I	
or MATH 345	Probability and Statistics for Data Science	
Core Courses: Science		
Select 12 credits from the following: [1]		12
Biology [2]		
CH 111	Introduction to Chemical Principles	
or CH 113	The Chemistry of Sustainability	
or CH 221	General Chemistry I	
or CH 224H	Advanced General Chemistry I	
BI 211,213	General Biology I,III	
or BI 211–212	General Biology I-II	
Chemistry [2]		
CH 221–223	General Chemistry	
or CH 224H-226H	Honors General Chemistry	
Earth Sciences		
ERTH 201	Dynamic Planet Earth	
ERTH 202	Earth's Surface and Environment	
ERTH 203	History of Life	
Geography		
GEOG 141	The Natural Environment	
Select two of the following:		
GEOG 321	Climatology	
GEOG 322	Geomorphology	
GEOG 323	Biogeography	
Physics [2,3]		
PHYS 201–203	General Physics	
or PHYS 251-253	Foundations of Physics I	
Psychology		
PSY 201	Mind and Brain	
Select two of the following:		
PSY 301	Scientific Thinking in Psychology	
PSY 304	Biopsychology	
PSY 305	Cognition	
PSY 348	Music and the Brain	
Core Course: Writing		
WR 320	Scientific and Technical Writing	4
or WR 321	Business Communications	
Electives: Upper Division		
Upper-division CS courses in student's chosen concentration (concentration information below)		12
Upper-division CS courses in student's chosen concentration, honors thesis, capstone project, or other upper-division courses [4,5]		8
Upper-division mathematics or theoretical computer science course [6]		4
Total Credits		**104**

1. To support interdisciplinary study, students in any concentration are encouraged to complete a minor (typically 24–32 credits) or major in a computing-related field. Students who complete a minor (other than computer information technology or mathematics) or another major (including mathematics) in a computing-related field may, with the approval of the Undergraduate Education Committee, replace the CS laboratory science requirement with the completed minor or major.
2. Students are encouraged to complete the accompanying lab courses.
3. Physics is recommended for networks concentration students.
4. If Experimental Course: [Topic] (CS 410) courses are applied, they must have different topic subtitles to satisfy this requirement.

[5] A maximum of 8 credits in upper-division courses numbered less than 410, or DSCI 311. CS courses numbered 400–499 may be taken for a maximum of 4 credits when used to satisfy this requirement. Special Studies: [Topic] (CS 399), Seminar: [Topic] (CS 407), and Experimental Course: [Topic] (CS 410) courses must have different topic subtitles to satisfy this requirement. CS 399 and CS 410 courses must have a prerequisite of CS 313 and have regular weekly class meetings and homework assignments.

[6] The mathematics elective is selected from upper-division mathematics courses with a prerequisite of Calculus II (MATH 252) or higher, or from theoretical computer science courses. A list of theoretical computer science courses is available in the computer science office or the department website.

Upper-Division Electives

In addition to the core computer science, mathematics, science, and writing courses, Computer Science majors must complete 20 credits of upper-division computer science and 4 credits of upper-division mathematics or theoretical computer science. Students should choose at least 12 of their upper-division CS credits with courses from courses numbered 410 or higher; the 8 remaining upper-division credits may be additional upper-division courses, an honors thesis, capstone project, or other upper-division electives. Students with a concentration should choose at least 12 of their upper-division CS credits with courses from their selected concentration.

A maximum of 8 credits in upper-division CS courses with numbers less than 410 may be applied to the upper-division electives requirement. Courses numbered 400–409 may be taken for a maximum of 4 credits when used to satisfy this requirement. (Courses numbered 399, 407, or 410 may be repeated only with different course subtitles.) Special Studies: [Topic] (CS 399) and Experimental Course: [Topic] (CS 410) courses used as upper-division electives must have a prerequisite of CS 313 and have regular weekly class meetings and homework assignments.

The mathematics elective is selected from upper-division mathematics courses with a prerequisite of MATH 252 or higher, or from theoretical computer science courses. A list of courses is available in the computer science office or at the department website.

Concentrations

Concentrations, or focus areas, highlight areas of specialization within the department and guide student elective choices. Each concentration has an approved list of CS courses, available from the Computer Science office or the department website. Concentrations may also include recommended science or mathematics courses or a recommended minor in another field.

AI and Machine Learning Concentration

The artificial intelligence and machine learning concentration prepares students to develop computational solutions to problems that require emerging problem solving techniques, often involving inference from large collections of noisy data. Course work focuses on neural and statistical approaches to inference as well as search.

Computer Networks Concentration

The Computer Networks concentration prepares students for careers as network systems administrators, network protocol developer-programmers, or network security specialists in a wide range of environments, including educational institutions, business enterprises, and government agencies, as well as for advanced graduate studies and research in the field of computer networks. Course work encompasses most aspects of network theory and practice.

High Performance Computing/Computational Science Concentration

The High Performance Computing/Computational science concentration prepares students to apply computational and mathematical techniques to the analysis and management of scientific data. Course work in this concentration combines depth in applied and formal aspects of Computer Science.

Security Concentration

The Security concentration provides a foundation in topics and concepts relating to the security of computer systems and networks. It prepares students to work as security analysts and provides a highly desirable skill set for all employers, ranging from software engineers to administrators, in both the private and government sectors. It also provides a foundation for further graduate study and research in security. Course work encompasses a strong understanding of computer systems and networks and their security, and can be tailored to a more theoretical or more applied focus.

Software Development Concentration

The Software Development concentration prepares students for careers in software engineering, software project management, software quality assurance, and other areas involving the creation of software. Course work focuses on solving problems related to the cost of development as well as the quality of the software delivered in complex software projects.

Preparation for the Major

Students who take Computer Science I (CS 210) are expected to have completed Elementary Functions (MATH 112) or the equivalent. Students who do not have the required mathematical background are strongly encouraged to take one or more introduction to programming courses such as Introduction to Programming and Problem Solving (CS 122) along with their math preparation courses. Students who are unsure about their level of preparation for CS 210 should meet with an advisor.

Sequence of Courses for Students Prepared for CS 210

First Year		Credits
CS 210–212	Computer Science I-III	12
MATH 231–232	Elements of Discrete Mathematics I-II	8
	Total Credits:	**20**

Sequence of Courses for Students Preparing for CS 210

First Year		
Fall		**Credits**
MATH 112	Elementary Functions	4
Select one of the following: [1]		4
CS 122	Introduction to Programming and Problem Solving	4
CS 111	Introduction to Web Programming	4
Winter		

MATH 231	Elements of Discrete Mathematics I	4
CS 210	Computer Science I	4
Spring		
MATH 232	Elements of Discrete Mathematics II	4
CS 211	Computer Science II	4
	Total Credits:	**32**

1 Taken either in first term or first year. Students are encouraged to take more than one course. CS 122 strongly recommended.

2 Taken in either second term or second year.

Major Progress Review and Major in Good Standing

Each major must meet with a computer science advisor and file the Major Progress Review form while taking Intermediate Data Structures (CS 313) or Computer Organization (CS 314). Mathematics and CS core courses and at least 12 credits of CS upper division elective courses numbered 410 or higher must be taken for letter grades and passed with grades of C– or better. Other courses used to satisfy the major requirements may be taken for letter grades or pass/no pass. Grades of C– or better or P must be earned in these courses. At least 12 of the CS upper-division credits applied to the degree must be taken in residence at the university. A student who receives two grades below C– in the upper-division core, or three grades below C– in any CS upper-division courses, will be removed from the major.

Mathematics and Computer Science

The Department of Computer Science and the Department of Mathematics jointly offer an undergraduate major in mathematics and computer science, leading to a bachelor of arts or a bachelor of science degree. This program is described in the Mathematics and Computer Science section of this catalog. This major prepares students for a wide range of careers in the high tech industry, for advanced graduate study, and for careers as middle school and high school teachers.

Honors Program

Students with a GPA of 3.50 or higher in computer science and a cumulative GPA of 3.50 or higher, or a GPA of 3.75 or higher in computer science and a cumulative GPA of 3.25 or higher, are encouraged to apply to the department honors program after completing Intermediate Data Structures (CS 313), Computer Organization (CS 314), Intermediate Algorithms (CS 315), and C/C++ and Unix (CS 330). The application form is available in the department office. To graduate with departmental honors, a student must write and present a thesis under the supervision of a faculty member. Honors students will take 4 credits of CS 403 Thesis and up to 4 credits of CS 401 Research to satisfy this requirement.

Computer Science Accelerated Master's Program

Computer Science undergraduate majors at the UO have the opportunity to graduate with B.S. and M.S. Computer Science degrees in a five-year program.

Students with junior or senior status in the upper-division of the Computer Science major may apply to the Accelerated Master's Program in Computer Science (AMP-CS). Upon acceptance to AMP-CS, students may take graduate courses to fulfill major requirements that will also fulfill Master's degree requirements, according to UO AMP guidelines, for students admitted to the Computer Science Master's degree program.

AMP-CS students will work closely with undergraduate and graduate advisors to ensure that they are on track to graduate with both degrees. Students who do not maintain satisfactory progress or who choose not to continue to a master's degree can still graduate with a B.S. degree in Computer Science in four years.

Computer Science Early Start Master's Program

This program is open to students who earn a BS or BA degree in computer science at the University of Oregon and who want to enter the master's degree program.

If a UO undergraduate takes one or two 400-level electives that also are offered as 500-level courses, the student can petition the department to have 4 or 8 credits deducted from the total number of elective credits required for the master's degree. The student must earn an A– or better in the 400-level course and have an overall GPA of 3.50 in upper-division CS courses to participate in this Computer Science Early Start Master's program. Note that all admission procedures, as outlined in the Master's Degree Program (p. 11) section, are also applicable. Applications are available in the department office. This program may be combined with the Computer Science Accelerated Master's Program (described above).

Internships

Practical work experience in the software industry is seen as a valuable complement to academic coursework. The department works with students to place them in internship positions in the summer and throughout the academic year. Students may also use the services of the University Career Center and other agencies to identify internship opportunities. Majors may receive academic credit (CS 404) for internships. To earn upper-division elective credit for an internship, the work experience must be at a technical level beyond Intermediate Data Structures (CS 313) and be sponsored by a CS faculty member. A contract signed by the faculty sponsor, internship supervisor, and the student must be filed with the department before the internship begins.

Research

Faculty members in the Computer Science department receive grants from government, industry, and private sources to conduct research in their areas of expertise. Undergraduate majors are encouraged to take part in the various research groups in the department. Most students begin approaching faculty members for such opportunities while taking the 300-level courses. Research can be used to fulfill upper-division electives, as part of an honors thesis, or in some cases as a paid internship.

Awards and Honor Societies

The Erwin and Gertrude Juilfs Scholarship in Computer Science, in honor of Erwin and Gertrude Juilfs, is awarded to one or more students who show exceptional promise for achievement as evidenced by grade point average, originality of research, or other creative activities.

The Geoffery Eric Wright Outstanding Junior Award, in honor of CS student Geoffery Wright, is a scholarship for students displaying high-quality academic performance, commitment to learning, and a promise of further outstanding achievement in computer science and its applications.

The J. Donald Hubbard Scholarship in Computer Science, in honor of J. Donald Hubbard, recognizes an undergraduate or graduate student who shows outstanding promise in the fields of computer-human interaction, computer graphics, or multimedia.

The Phillip Seeley Scholarship in Computer Science has been established as a permanent endowment to provide a source of income supporting a scholarship for outstanding undergraduate CS students. This scholarship is based on overall quality of academic work, commitment to learning, and potential for further academic achievement. Preference is given to resident Oregon students with financial need, as determined by the UO Office of Student Financial Aid and Scholarships.

Students with outstanding academic accomplishments may be invited to become members of Upsilon Pi Epsilon, the international honor society in computer science.

Minor Requirements

Computer Science Minor

The minor in computer science introduces the theories and techniques of computer science and develops programming skills that are applicable to the student's major. It is a strong complement to a major in any of the sciences and in related fields such as multimedia arts. Students from all majors have found their career opportunities enhanced through the CS minor.

Before enrolling in upper-division courses, students planning a minor in computer science must file an application form with the department. Each student should consult with a CS faculty advisor to plan the minor program.

Computer Science I (CS 210), Computer Science II (CS 211), Computer Science III (CS 212), Elements of Discrete Mathematics I (MATH 231), and Elements of Discrete Mathematics II (MATH 232) must be passed with grades of B- or better before students can take the upper-division core courses.

Code	Title	Credits
Lower-Division Courses		
CS 210–212	Computer Science I-III	12
MATH 231–232	Elements of Discrete Mathematics I-II	8
Upper-Division Courses		
CS 313	Intermediate Data Structures	4
Electives [1]		8
Total Credits		**32**

[1] Course numbers less than 410 with departmental permission only

Computer Information Technology Minor

The minor in computer information technology (CIT) prepares students to work with evolving technologies for work environments that require development and management of web applications, databases, computer networks, open-source platforms, and cloud computing. It provides practical experience in understanding the tools and technologies of the computing field. It goes well with majors in the professional schools such as business and journalism and is an excellent match with almost any major on campus.

Before enrolling in CIT upper-division courses, students planning a minor in computer information technology must file an application form with the department. Each student should consult with an assigned CIT faculty advisor to plan the minor program.

Lower-division courses must be completed with grades of B– or better. Upper-division courses must be taken in sequence and are offered only once a year. Upper-division courses must be completed with grades of C– or better.

Code	Title	Credits
Lower-Division Courses		
CS 110	Fluency with Information Technology	4
CS 111	Introduction to Web Programming	4
CIT 281	Web Applications Development I	4
Upper-Division Courses		
CIT 381	Database Systems	4
CIT 382	Web Applications Development II	4
CIT 383	Networking Fundamentals	4
Total Credits		**24**

Four-Year Degree Plan

The degree plan shown is only a sample of how students may complete their degrees in four years. There are alternative ways. Students should consult their advisor to determine the best path for them. Additional information may be found at the department website *(*https://cs.uoregon.edu*).*

Bachelor of Arts in Computer Science

Course	Title	Credits	Milestones
First Year			
Fall			
CS 122	Introduction to Programming and Problem Solving (recommended)	4	
MATH 112	Elementary Functions	4	
WR 121	College Composition I	4	
Core-education course in arts and letters		4	
	Credits	**16**	
Winter			
CS 210	Computer Science I	4	
MATH 251 or MATH 246 or MATH 261	Calculus I or Calculus for the Biological Sciences I or Calculus with Theory I	4	
WR 122 or WR 123	College Composition II or College Composition III	4	
Core-education course in social science			
	Credits	**12**	
Spring			
CS 211	Computer Science II	4	
MATH 252 or MATH 247 or MATH 262	Calculus II or Calculus for the Biological Sciences II or Calculus with Theory II	4	
Core-education course in arts and letters		4	

Core-education course in social science		4
	Credits	**16**
	Total Credits	**44**

Course	Title	Credits	Milestones
Second Year			
Fall			
CS 212	Computer Science III	4	
MATH 231	Elements of Discrete Mathematics I	4	
First course of additional science sequence		4	
Core-education course in social science that also satisfies a cultural literacy requirement		4	
	Credits	**16**	
Winter			
CS 314	Computer Organization	4	
MATH 232	Elements of Discrete Mathematics II	4	
Second course of additional science sequence		4	
Core-education course in arts and letters		4	
	Credits	**16**	
Spring			
CS 322 or WR 320 or WR 321	Introduction to Software Engineering (recommended) or Scientific and Technical Writing or Business Communications	4	
MATH 253 or MATH 263 or MATH 341 or MATH 343 or MATH 425 or MATH 345	Calculus III or Calculus with Theory III or Elementary Linear Algebra or Statistical Models and Methods or Statistical Methods I or Probability and Statistics for Data Science	4	
Third course of additional science sequence		4	
Core-education course in arts and letters that also satisfies a cultural literacy requirement		4	
	Credits	**16**	
	Total Credits	**48**	

Course	Title	Credits	Milestones
Third Year			
Fall			
CS 313	Intermediate Data Structures	4	
MATH 253 or MATH 263 or MATH 341 or MATH 343 or MATH 425 or MATH 345	Calculus III or Calculus with Theory III or Elementary Linear Algebra or Statistical Models and Methods or Statistical Methods I or Probability and Statistics for Data Science	4	
WR 320 or WR 321 or CS 322	Scientific and Technical Writing or Business Communications or Introduction to Software Engineering	4	
First term of second-language sequence		4	
	Credits	**16**	
Winter			
CS 315	Intermediate Algorithms	4	
CS 330	C/C++ and Unix	4	
Upper-division mathematics elective course		4	
Second term of second-language sequence		4	
	Credits	**16**	
Spring			
CS 415	Operating Systems	4	
Upper-division elective course with CS subject code		4	
PHIL 223	Data Ethics (or core-education course in social science)	4	
Third term of second-language sequence		4	
	Credits	**16**	
	Total Credits	**48**	

Course	Title	Credits	Milestones
Fourth Year			
Fall			
CS 425	Principles of Programming Languages	4	
Upper-division elective course (410-499) with CS subject code		4	
Upper-division elective course		4	
	Credits	**12**	
Winter			
CS 422	Software Methodology I	4	
Upper-division elective course (410-499) with CS subject code		4	
Upper-division elective course		4	
	Credits	**12**	
Spring			
Upper-division elective course with CS subject code		4	
Upper-division elective course (410-499) with CS subject code		4	
Upper-division elective course		4	
	Credits	**12**	
	Total Credits	**36**	

Bachelor of Science in Computer Science

Course	Title	Credits	Milestones
First Year			
Fall			
CS 122	Introduction to Programming and Problem Solving (recommended)	4	
MATH 112	Elementary Functions	4	
WR 121	College Composition I	4	
Core-education course in arts and letters		4	
	Credits	**16**	

Winter		
CS 210	Computer Science I	4
MATH 251 or MATH 246 or MATH 261	Calculus I or Calculus for the Biological Sciences I or Calculus with Theory I	4
WR 122 or WR 123	College Composition II or College Composition III	4
Core-education course in social science		4
	Credits	**16**
Spring		
CS 211	Computer Science II	4
MATH 252 or MATH 247 or MATH 262	Calculus II or Calculus for the Biological Sciences II or Calculus with Theory II	4
Core-education course in arts and letters		4
Core-education course in social science		4
	Credits	**16**
	Total Credits	**48**

Course	Title	Credits	Milestones
Second Year			
Fall			
CS 212	Computer Science III	4	
MATH 231	Elements of Discrete Mathematics I	4	
First course of additional science sequence		4	
Core-education course in social science that also satisfies a cultural literacy requirement		4	
	Credits	**16**	
Winter			
CS 314	Computer Organization	4	
MATH 232	Elements of Discrete Mathematics II	4	
Second course of additional science sequence		4	
Core-education course in arts and letters		4	
	Credits	**16**	
Spring			
CS 322 or WR 320 or WR 321	Introduction to Software Engineering (recommended) or Scientific and Technical Writing or Business Communications	4	
MATH 253 or MATH 263 or MATH 341 or MATH 343 or MATH 425 or MATH 345	Calculus III or Calculus with Theory III or Elementary Linear Algebra or Statistical Models and Methods or Statistical Methods I or Probability and Statistics for Data Science	4	
Third course of additional science sequence		4	
Core-education course in arts and letters that also satisfies a cultural literacy requirement		4	
	Credits	**16**	
	Total Credits	**48**	

Course	Title	Credits	Milestones
Third Year			
Fall			
CS 313	Intermediate Data Structures	4	
MATH 253 or MATH 263 or MATH 341 or MATH 343 or MATH 425 or MATH 345	Calculus III or Calculus with Theory III or Elementary Linear Algebra or Statistical Models and Methods or Statistical Methods I or Probability and Statistics for Data Science	4	
WR 320 or WR 321 or CS 322	Scientific and Technical Writing or Business Communications or Introduction to Software Engineering	4	
Elective course		4	
	Credits	**16**	
Winter			
CS 315	Intermediate Algorithms	4	
CS 330	C/C++ and Unix	4	
Upper-division mathematics elective course		4	
Elective course		4	
	Credits	**16**	
Spring			
CS 415	Operating Systems	4	
Upper-division elective course with CS subject code		4	
PHIL 223	Data Ethics (or core-education course in social science)	4	
Elective course		4	
	Credits	**16**	
	Total Credits	**48**	

Course	Title	Credits	Milestones
Fourth Year			
Fall			
CS 425	Principles of Programming Languages	4	
Upper-division elective course (410-499) with CS subject code		4	
Upper-division elective course		4	
	Credits	**12**	
Winter			
CS 422	Software Methodology I	4	
Upper-division elective course (410-499) with CS subject code		4	
Upper-division elective course		4	
	Credits	**12**	

Spring	
Upper-division elective course with CS subject code	4
Upper-division elective course (410-499) with CS subject code	4
Upper-division elective course	4
Credits	**12**
Total Credits	**36**

- **Master of Science**
- **Doctor of Philosophy**

Graduate Studies

The department offers programs leading to the master of science (MS) and doctor of philosophy (PhD).

Computer Science Accelerated Master's Program

Computer Science undergraduate majors at the UO will have the opportunity to graduate with B.S. and M.S. Computer Science degrees in a five-year program.

Students with junior or senior status in the upper-division of the Computer Science major may apply to the Accelerated Master's Program in Computer Science (AMP-CS). Upon acceptance to AMP-CS, students may take graduate courses to fulfill major requirements that will also fulfill Master's degree requirements, according to UO AMP guidelines, for students admitted to the Computer Science Master's degree program.

AMP-CS students will work closely with undergraduate and graduate advisors to ensure that they are on track to graduate with both degrees. Students who do not maintain satisfactory progress or who choose not to continue to a master's degree can still graduate with a B.S. degree in Computer Science in four years.

Computer Science Early Start Master's Program

This program is open to students who earn a BS or BA degree in computer science at the University of Oregon and who want to enter the master's degree program.

If a UO undergraduate takes one or two 400-level electives that also are offered as 500-level courses, the student can petition the department to have 4 or 8 credits deducted from the total number of elective credits required for the master's degree. The student must earn an A– or better in the 400-level course and have an overall GPA of 3.50 in upper-division CS courses to participate in this Computer Science Early Start Master's program. Note that all admission procedures, as outlined in the Master's Degree Program section, are also applicable. Applications are available in the department office. This program may be combined with the Computer Science Accelerated Master's Program (described above).

Master's Degree Program

Admission

Admission to the master's degree program in computer science is competitive. It is based on prior academic performance, Graduate Record Examinations (GRE) scores, and computer science background. Minimum requirements for admission with graduate master's status are as follows:

1. Documented knowledge of
   a. Principles of computer organization and operating systems
   b. Programming languages
   c. Program development and analysis
   d. Data structures and algorithm analysis
2. GRE scores on the general test. The computer science test is optional.
3. A score of at least 100 on the Internet-based option of the Test of English as a Foreign Language (TOEFL iBT) or a score of 7.0 on the International English Language Testing System (IELTS) for applicants who have no justification for a waiver. Applicants may be required to study one or more terms at the university's American English Institute or elsewhere before taking any graduate work in the department. International applicants for teaching assistantships who score at least 26 on the speaking section of the TOEFL iBT will not have to take the Speaking Proficiency English Assessment Kit (SPEAK) test upon arrival at the university.
4. Three letters of recommendation, a statement of purpose, and unofficial transcripts (via on online application). Note that official transcripts are sent to the UO Office of Admissions.

Grades from previous course work should indicate the ability to maintain at least a 3.00 grade point average in graduate-level courses.

Application materials should be submitted by February 1 for admission, via GradWeb (http://gradweb.uoregon.edu), for the following fall term.

Admission to the master's degree program requires the substantive equivalent of an undergraduate degree in computer science. A second bachelor's degree program can be used to gain the required level of computer science background. Students without this background may be admitted conditionally and required to complete remedial course work before achieving unconditional standing in the program.

Master of Science Degree Requirements

Code	Title	Credits
Breadth Requirement: 12 credits total [1]		**12**
CS 621	Algorithms and Complexity	
CS 670	Data Science	
And one of the following:		
CS 630	Distributed Systems	
CS 631	Parallel Processing	
Depth Requirement: Choose one, 12 credits total [1]		**12**
Each Depth requires three courses, at least one at 600-level		
Foundations Depth		
CS 513	Advanced Data Structures	
CS 520	Automata Theory	
CS 527	Introduction to Logic	
CS 543	User Interfaces	
CS 545	Modeling and Simulation	
CS 561	Introduction to Compilers	
CS 624	Structure of Programming Languages	
Data Science Depth		
CS 543	User Interfaces	
CS 551	Database Processing	
CS 553	Data Mining	
CS 571	Introduction to Artificial Intelligence	

CS 572	Machine Learning	
CS 573	Probabilistic Methods for Artificial Intelligence	
CS 600 level course		
Systems Depth		
CS 531	Introduction to Parallel Computing	
CS 532	Introduction to Networks	
CS 533	Computer and Network Security	
CS 541	Introduction to Computer Graphics	
CS 561	Introduction to Compilers	
CS 630	Distributed Systems	
CS 631	Parallel Processing	
CS 632	Computer Networks	
CS 633	Advanced Network Security	
Writing Requirement		**2**
CS 640	Writing in Computer Research	
Elective Options: 28 credits total [4]		**28**
Up to twelve credits in courses outside department in area closely related to professional goals may be used [5]		
Thesis Option [6]		
CS 503	Thesis	
CS 5XX and CS 6XX: Minimum of 8 graded credits, maximum of 11 P/NP credits		
Non-Thesis Option		
CS 609	Terminal Project	
CS 5XX and CS 6XX: Minimum of 18 graded credits, maximum of 10 P/NP credits		
DRP Option [8]		
CS 601	Research: [Topic]	
CS 5XX and CS 6XX: Minimum of 8 graded credits		
Completion of the directed research project (DRP) milestone in the CS PhD program and approval from the DRP committee that the project meets the standards of an MS thesis.		
Total Credits		**54**

1. A grade of B- or better is required
2. Cannot duplicate Depth course used
3. Cannot duplicate Breadth course used
4. A grade of C or better is required in graded elective credits
5. Courses must be approved by petition to the CS GEC; options include courses in linguistics, mathematics, physics, and psychology.
6. Cannot include CS 609 Final Project
7. CS 609 Final Project requirements: 8-12 credits; graded or P/NP
8. No credit of CS 503 (Thesis) or CS 609 (Final Project) may count toward the elective credit requirements

Grade Requirements

The 24 credits in the breadth courses and the depth courses must be passed with grades of B– or better. Graded elective courses must be passed with grades of C or better. A 3.00 GPA must be maintained for courses taken in the program.

Master's Thesis

The research option requires a written thesis and 9 to 12 credits in Thesis (CS 503). Thesis research is supervised by a faculty advisor; this advisor and other faculty members constitute the thesis committee. The master's thesis is expected to be scholarly and to demonstrate mastery of the practices of computer science. This option is recommended for students who plan subsequent PhD research.

Master's Project

The project option requires a minimum of 9 credits, and as many as 12, in Final Project Terminal Project (CS 609).

Under the supervision of a faculty member, the project may entail a group effort involving several master's degree students.

Awards and Honor Societies

The Erwin and Gertrude Juilfs Scholarship in Computer Science, in honor of Erwin and Gertrude Juilfs, is awarded to one or more students who show exceptional promise for achievement as evidenced by grade point average, originality of research, or other creative activities.

The J. Donald Hubbard Scholarship in Computer Science, in honor of J. Donald Hubbard, recognizes an undergraduate or graduate student who shows outstanding promise in the fields of computer-human interaction, computer graphics, or multimedia.

The Gurdeep Pall Scholarship in Computer Science, in honor of Gurdeep Pall, is awarded to a student based on the overall quality of their academic work, their commitment to learning, and their potential for further academic achievement.

Students with outstanding academic accomplishments may be invited to become members of Upsilon Pi Epsilon, the international honor society in computer science.

Doctoral Degree Program

The doctor of philosophy in computer science is, above all, a high-quality degree that is not conferred simply for the successful completion of a specified number of courses or years of study. It is a degree reserved for students who demonstrate a comprehensive understanding of computer science and an ability to do creative research. Each PhD student produces a significant piece of original research, presented in a written dissertation and defended in an oral examination.

The PhD program is structured to facilitate the process of learning how to do research. Students begin by taking required courses to build a foundation of knowledge that is essential for advanced research. Early in the program the student gains research experience by undertaking a directed research project under the close supervision of a faculty member and the scrutiny of a faculty committee. In the later stages of the program, students take fewer courses and spend most of their time exploring their dissertation area to learn how to identify and solve open problems. The final steps are to propose an independent research project, do the research, and write and defend a dissertation.

Admission

Application materials should be submitted by December 15 for the following fall term. Materials include everything required for admission to the master's program as well as a discussion of the anticipated research area.

Students who enter the UO with a master's degree may petition the Graduate Education Committee for credit toward the course requirements listed below, indicating how their prior graduate work corresponds to these courses. See the graduate coordinator for the petition.

PhD Course Requirements

Code	Title	Credits
Breadth Requirement: 12 credits total [1]		**12**
CS 621	Algorithms and Complexity	
CS 670	Data Science	
And one of the following:		
CS 630	Distributed Systems	
CS 631	Parallel Processing	
Depth Requirement: Choose one, 12 credits total [1]		**12**
Each Depth requires three courses, at least one at 600-level		
Foundations Depth		
CS 513	Advanced Data Structures	
CS 520	Automata Theory	
CS 527	Introduction to Logic	
CS 543	User Interfaces	
CS 545	Modeling and Simulation	
CS 561	Introduction to Compilers	
CS 624	Structure of Programming Languages	
Data Science Depth		
CS 543	User Interfaces	
CS 553	Data Mining	
CS 571	Introduction to Artificial Intelligence	
CS 572	Machine Learning	
CS 573	Probabilistic Methods for Artificial Intelligence	
CS 6XX TBA		
Systems Depth		
CS 531	Introduction to Parallel Computing	
CS 532	Introduction to Networks	
CS 533	Computer and Network Security	
CS 541	Introduction to Computer Graphics	
CS 561	Introduction to Compilers	
CS 630	Distributed Systems	
CS 631	Parallel Processing	
CS 632	Computer Networks	
CS 633	Advanced Network Security	
Writing Requirement		**2**
CS 640	Writing in Computer Research	
Elective Options: 24 credits total		**24**
Total Credits		**50**

1. A grade of B- or better is required
2. Cannot duplicate Depth course used
3. Cannot duplicate Breadth course used
4. A grade of C or better is required in graded elective credits

PhD Degree Requirements

PhD candidates who enter the program without a master's degree in computer science must take 48 credits in graduate course work including the core and cluster courses required for the MS program. Doctoral students must earn a minimum grade of B– and an overall GPA of 3.50 in the six courses they use to satisfy the breadth and depth requirements.

Minimum Annual Enrollment

PhD students are expected to enroll in at least 6 credits of 600-level course work each year until their advancement to candidacy. Research: [Topic] (CS 601), Dissertation (CS 603), and Reading Conference: [Topic] (CS 605) do not satisfy this requirement. After candidacy, PhD students are encouraged to continue participation in 600-level courses

Directed Research Project

Complete a directed research project, which is supervised by a faculty member and evaluated by a faculty committee. The research project comprises the following:

1. The definition and expected results of the project in the form of a Directed Research Project Contract
2. Delivery of the materials constituting the results of the project and oral presentation of the results
3. A private oral examination by the committee members

Status Change

PhD candidates are admitted conditionally. Successful completion of the directed research project leads to a change in the student's doctoral status from conditional to unconditional.

Dissertation Advisory Committee

After successfully completing the directed research project, PhD students form a Dissertation Advisory Committee chaired by their research advisor. The main role of the committee is to advise the student between completion of the research project and mounting the dissertation defense. The committee takes primary responsibility for evaluating student progress. In addition, it approves the plan for the area examination, which in turn is approved by the graduate education committee. See the graduate coordinator for further instructions.

Area Examination

The student chooses an area of research and works closely with an advisor to learn the area in depth by surveying the current research and learning research methods, significant achievements, and how to pose and solve problems. The student gradually assumes a more independent role and prepares for the area examination, which tests depth of knowledge in the research area. The examination contains the following:

1. A survey of the area in the form of a position paper and an annotated bibliography
2. A public presentation of the position paper
3. A private oral examination by committee members

Advancement to Candidacy

After the area examination, the committee decides whether the student is ready for independent research work; if so, the student is advanced to candidacy.

Dissertation and Defense

Identify a significant unsolved research problem and submit a written dissertation proposal to the dissertation committee. The dissertation committee, comprising three department members and one member

from an outside department, is approved by the graduate education committee. In addition to these four, the dissertation committee often includes a fifth examiner. This outside examiner should be a leading researcher in the candidate's field who is not at the University of Oregon. The outside member should be selected a year before the candidate's dissertation defense, and no later than six months before.

The student submits a written dissertation proposal to the committee for approval, and the proposal is then submitted to the graduate education committee. The proposal presents the research problems to be tackled, related research, methodology, anticipated results, and work plan. The committee may request an oral presentation, similar to the area exam, which allows the student to explain and answer question about the proposed research. The student then carries out the research.

The final stage is writing a dissertation and defending it in a public forum by presenting the research and answering questions about the methods and results. The dissertation committee may accept the dissertation, request small changes, or require the student to make substantial changes and schedule another defense

Division of Graduate Studies Requirements

PhD students must meet the requirements set by the Division of Graduate Studies as listed in that section of this catalog

Research Areas

It is important that a PhD student be able to work effectively with at least one dissertation advisor. Hence the student should identify, at an early stage, one or more areas of research to pursue. The student should also find a faculty member with similar interests to supervise the dissertation.

Computer Information Technology Courses

CIT 281. Web Applications Development I. 4 Credits.
Fundamentals of web application development using open-source software tools and technologies (Unix, Git), client-side frameworks, server-side programming (Node.js, PHP), model-view-controller pattern, data storage and APIs, cloud hosting.
Prereq: CS 111 with a B- or higher.

CIT 381. Database Systems. 4 Credits.
Introduction to database systems, emphasis on database design and access. Database concepts, data modeling, SQL, connecting database to web.
Prereq: B- or better in CIT 281, and CS 110 or CS 115.

CIT 382. Web Applications Development II. 4 Credits.
Server- and client-side technologies and their interaction for database-driven web applications: application frameworks, single-page applications, cloud platforms, and open-source software stacks—MEAN (MongoDB, ExpressJS, AngularJS, Node.js) versus LAMP (Linux, Apache, MySQL, PHP).
Prereq: CIT 381.

CIT 383. Networking Fundamentals. 4 Credits.
Fundamentals of data communication and networks. Network management and security.
Prereq: CIT 382.

CIT 405. Reading and Conference: [Topic]. 1-4 Credits.
Repeatable.

Computer Science Courses

CS 102. Fundamentals of Computer and Information Security. 4 Credits.
This course introduces fundamental concepts, terminologies, principles, methods, and scenarios of computer and information security.

CS 110. Fluency with Information Technology. 4 Credits.
Introduction to information technology (IT), the study of computer-based information systems. Basics of the Internet and World Wide Web. Students create websites using XHTML and CSS.

CS 111. Introduction to Web Programming. 4 Credits.
Project-based approach to learning computer programming by building interactive web pages using JavaScript and XHTML. Programming concepts including structured and object-oriented program design. CS 110 recommended preparation.
Prereq: MATH 101 or equivalent.

CS 122. Introduction to Programming and Problem Solving. 4 Credits.
Computational problem solving, algorithm design, data structures, and programming using a multi-paradigm programming language. Introduces techniques for program design, testing, and debugging.
Prereq: MATH 101 or equivalent.

CS 196. Field Studies: [Topic]. 1-2 Credits.
Repeatable.

CS 198. Workshop: [Topic]. 1-2 Credits.
Repeatable.

CS 199. Special Studies in Computer Science: [Topic]. 1-5 Credits.
Repeatable.

CS 199L. Special Studies in Computer Science: [Topic]. 1-5 Credits.
Repeatable.

CS 210. Computer Science I. 4 Credits.
Basic concepts and practices of computer science. Topics include algorithmic problem solving, levels of abstraction, object-oriented design and programming, software organization, analysis of algorithm and data structures. Sequence with CS 211, CS 212.
Prereq: MATH 112. Prior programming experience strongly encouraged.

CS 211. Computer Science II. 4 Credits.
Basic concepts and practices of computer science. Topics include algorithmic problem solving, levels of abstraction, object-oriented design and programming, software organization, analysis of algorithm and data structures. Sequence with CS 210, CS 212.
Prereq: CS 210.

CS 212. Computer Science III. 4 Credits.
Basic concepts and practices of computer science. Topics include algorithmic problem solving, levels of abstraction, object-oriented design and programming, software organization, analysis of algorithm and data structures. Sequence with CS 210, CS 211.
Prereq: CS 211.

CS 313. Intermediate Data Structures. 4 Credits.
Design and analysis of data structures as means of engineering efficient software; attention to data abstraction and encapsulation. Lists, trees, heaps, stacks, queues, dictionaries, priority queues.
Prereq: CS 210, CS 211, CS 212, MATH 231, MATH 232 with grades of B- or better.

CS 314. Computer Organization. 4 Credits.
Introduction to computer organization and instruction-set architecture -- digital logic design, binary arithmetic, design of central processing unit and memory, machine-level programming.
Prereq: CS 210, CS 211, CS 212 with grades of B- or better.

CS 315. Intermediate Algorithms. 4 Credits.
Algorithm design, worst-case and average-behavior analysis, correctness, computational complexity.
Prereq: CS 313.

CS 322. Introduction to Software Engineering. 4 Credits.
A project-intensive introduction to software engineering intended to build skills, knowledge, and habits of mind that prepare students for 400-level computer science courses, internships, and other software.
Prereq: CS 210, CS 211, CS 212 with grades of B- or better.

CS 330. C/C++ and Unix. 4 Credits.
Practical software design and programming activities in a C/C++ and Unix environment, with emphasis on the details of C/C++ and good programming style and practices.
Prereq: CS 314.

CS 333. Applied Cryptography. 4 Credits.
This course provides a systematic study of cryptography and its application. It covers cryptographic algorithms, including symmetric-key cryptography, public-key cryptography, cryptanalysis, cryptographic hash functions, and their usage toward message authentication codes, digital signatures, key management and distribution, and user authentication protocols.
Prereq: CS 212.

CS 372M. Machine Learning for Data Science. 4 Credits.
Introduction to Machine Learning, with an emphasis on topics relevant for data science. Multilisted with DSCI 372M.
Prereq: CS 212, DSCI 345M, MATH 342.

CS 399. Special Studies: [Topic]. 1-5 Credits.
Repeatable when the topic changes.

CS 400M. Temporary Multilisted Course. 1-5 Credits.
Repeatable

CS 401. Research: [Topic]. 1-16 Credits.
Repeatable.
Prereq: CS 313.

CS 402. Supervised College Teaching. 1-2 Credits.
Repeatable.
Prereq: CS 313.

CS 403. Thesis. 1-12 Credits.
Repeatable.
Prereq: CS 313.

CS 404. Internship; [Topic]. 1-4 Credits.
Repeatable.
Prereq: CS 313.

CS 405. Reading and Conference: [Topic]. 1-12 Credits.
Repeatable up to five times.
Prereq: CS 313.

CS 406. Practicum: [Topic]. 1-2 Credits.
Supervised consulting. Students provide learning assistance in computer science courses. Repeatable for a maximum of 4 credits.
Prereq: CS 313.

CS 407. Seminar: [Topic]. 1-5 Credits.
Repeatable when the topic changes. Opportunity to study in greater depth specific topics arising out of other courses.
Prereq: CS 313.

CS 408. Workshop: [Topic]. 1-21 Credits.
Repeatable.
Prereq: CS 313.

CS 409. Terminal Project. 1-12 Credits.
Repeatable.
Prereq: CS 313.

CS 410. Experimental Course: [Topic]. 1-5 Credits.
Repeatable when the topic changes.

CS 413. Advanced Data Structures. 4 Credits.
Complex structures, storage management, sorting and searching, hashing, storage of texts, and information compression.
Prereq: CS 315.

CS 415. Operating Systems. 4 Credits.
Principles of operating system design. Process and memory management, concurrency, scheduling, input-output and file systems, security.
Prereq: CS 330.

CS 420. Automata Theory. 4 Credits.
Provides a mathematical basis for computability and complexity. Models of computation, formal languages, Turing machines, solvability. Nondeterminism and complexity classes.
Prereq: CS 315.

CS 422. Software Methodology I. 4 Credits.
Technical and nontechnical aspects of software development, including specification, planning, design, development, management and maintenance of software projects. Student teams complete projects.
Prereq: CS 313.

CS 423. Software Methodology II. 4 Credits.
Application of concepts and methodologies covered in CS 422/CS 522. Student teams complete a large system design and programming project. Final system specification, test plan, user documentation, and system walk throughs.
Prereq: CS 422 with a grade of B- or better.

CS 425. Principles of Programming Languages. 4 Credits.
Syntax and semantics. Scope rules, environments, stores, denoted and expressed values, procedures, and parameters. Definitional interpreters. Types, overloading, parametric polymorphism, and inheritance. Varieties of abstraction.
Prereq: CS 315.

CS 429. Computer Architecture. 4 Credits.
RISC (reduced instruction-set computer) and CISC (complex instruction-set computer) design, storage hierarchies, high-performance processor design, pipelining, vector processing, networks, performance analysis.
Prereq: CS 313, CS 314, CS 330.

CS 431. Introduction to Parallel Computing. 4 Credits.
Parallel architecture, theory, algorithms, and programming with emphasis on parallel programming, focusing on models, languages, libraries, and runtime systems.
Prereq: CS 330.

CS 432. Introduction to Networks. 4 Credits.
Principles of computer network design. Link technologies, packet switching, routing, inter-networking, reliability. Internet protocols. Programming assignments focus on protocol design.
Prereq: CS 330. CS 415 recommended.

CS 433. Computer and Network Security. 4 Credits.
Security for various aspects of computers and networks. Elementary cryptography, program security, trusted operating systems, network security, privacy, and legal and ethical issues.
Prereq: CS 415.

CS 434. Computer and Network Security II. 4 Credits.
This course covers security threats and solutions for distributed systems and networks, particularly the Internet, the Internet of Things, and distributed systems based on them.
Prereq: CS 432, CS 433.

CS 436. Secure Software Development. 4 Credits.
This course establishes a foundation for applying security principles to the lifecycle of software development in order to minimize software vulnerabilities and counter cyber threats.
Prereq: CS 330.

CS 441. Introduction to Computer Graphics. 4 Credits.
Introduction to the hardware, geometrical transforms, interaction techniques, and shape representation schemes that are important in interactive computer graphics. Programming assignments using contemporary graphics hardware and software systems.
Prereq: CS 330.

CS 443. User Interfaces. 4 Credits.
Introduction to user interface software engineering. Emphasis on theory of interface design, understanding the behavior of the user, and implementing programs on advanced systems.
Prereq: CS 313.

CS 445. Modeling and Simulation. 4 Credits.
Theoretical foundations and practical problems for the modeling and computer simulation of discrete and continuous systems. Simulation languages, empirical validation, applications in computer science.
Prereq: CS 315, CS 330.

CS 451. Database Processing. 4 Credits.
Fundamental concepts of DBMS. Data modeling, relational models and normal forms. File organization and index structures. SQL, embedded SQL, and concurrency control.
Prereq: CS 313, CS 314.

CS 453. Data Mining. 4 Credits.
Databases, machine learning, artificial intelligence, statistics, and data visualization. Examines data warehouses, data preprocessing, association and classification rule mining, and cluster analysis.
Prereq: CS 451/CS 551.

CS 461. Introduction to Compilers. 4 Credits.
Lexical analysis, parsing, attribution, code generation.
Prereq: CS 314, CS 425. CS 420 strongly recommended.

CS 471. Introduction to Artificial Intelligence. 4 Credits.
Basic themes, issues, and techniques of artificial intelligence, including agent architecture, knowledge representation and reasoning, problem solving and planning, game playing, and learning.
Prereq: CS 315.

CS 472. Machine Learning. 4 Credits.
A broad introduction to machine learning and its established algorithms. Topics include concept learning, decision trees, neural network.
Prereq: CS 315.

CS 473. Probabilistic Methods for Artificial Intelligence. 4 Credits.
Fundamental techniques for representing problems as probability distributions, performing inference, and learning from data. Topics include Bayesian and Markov networks, variable elimination, loopy belief propagation, and parameter.
Prereq: CS 315.

CS 500M. Temporary Multilisted Course. 1-5 Credits.
Repeatable.

CS 503. Thesis. 1-16 Credits.
Repeatable.

CS 507. Seminar: [Topic]. 1-5 Credits.
Repeatable. Opportunity to study in greater depth specific topics arising out of other courses.

CS 508. Workshop: [Topic]. 1-21 Credits.
Repeatable.

CS 510. Experimental Course: [Topic]. 1-5 Credits.
Repeatable.

CS 513. Advanced Data Structures. 4 Credits.
Complex structures, storage management, sorting and searching, hashing, storage of texts, and information compression.

CS 520. Automata Theory. 4 Credits.
Provides a mathematical basis for computability and complexity. Models of computation, formal languages, Turing machines, solvability. Nondeterminism and complexity classes.

CS 522. Software Methodology I. 4 Credits.
Technical and nontechnical aspects of software development, including specification, planning, design, development, management and maintenance of software projects. Student teams complete projects.

CS 523. Software Methodology II. 4 Credits.
Student teams complete a large system design and programming project. Final system specifications, test plan, user documentation, and system walk-through.
Prereq: CS 522.

CS 529. Computer Architecture. 4 Credits.
RISC (reduced instruction-set computer) and CISC (complex instruction-set computer) design, storage hierarchies, high-performance processor design, pipelining, vector processing, networks, performance analysis.

CS 531. Introduction to Parallel Computing. 4 Credits.
Parallel architecture, theory, algorithms, and programming with emphasis on parallel programming, focusing on models, languages, libraries, and runtime systems.

CS 532. Introduction to Networks. 4 Credits.
Principles of computer network design. Link technologies, packet switching, routing, inter-networking, reliability. Internet protocols. Programming assignments focus on protocol design.

CS 533. Computer and Network Security. 4 Credits.
Security for various aspects of computers and networks. Elementary cryptography,program security, trusted operating systems, network security, privacy, and legal and ethical issues.

CS 534. Computer and Network Security II. 1-4 Credits.
This course covers security threats and solutions for distributed systems and networks, particularly the Internet, the Internet of Things, and distributed systems based on them.
Prereq: CS 532, CS 533.

CS 536. Secure Software Development. 4 Credits.
This course establishes a foundation for applying security principles to the lifecycle of software development in order to minimize software vulnerabilities and counter cyber threats.

CS 541. Introduction to Computer Graphics. 4 Credits.
Introduction to the hardware, geometrical transforms, interaction techniques, and shape representation schemes that are important in interactive computer graphics. Programming assignments using contemporary graphics hardware and software systems.

CS 543. User Interfaces. 4 Credits.
Introduction to user interface software engineering. Emphasis on theory of interface design, understanding the behavior of the user, and implementing programs on advanced systems.

CS 545. Modeling and Simulation. 4 Credits.
Theoretical foundations and practical problems for the modeling and computer simulation of discrete and continuous systems. Simulation languages, empirical validation, applications in computer science.

CS 551. Database Processing. 4 Credits.
Fundamental concepts of DBMS. Data modeling, relational models and normal forms. File organization and index structures. SQL, embedded SQL, and concurrency control.

CS 553. Data Mining. 4 Credits.
Databases, machine learning, artificial intelligence, statistics, and data visualization. Examines data warehouses, data preprocessing, association and classification rule mining, and cluster analysis.
Prereq: CS 551.

CS 561. Introduction to Compilers. 4 Credits.
Lexical analysis, parsing, attribution, code generation.
Prereq: CS 314 or equivalent. CS 520 strongly recommended.

CS 571. Introduction to Artificial Intelligence. 4 Credits.
Basic themes, issues, and techniques of artificial intelligence, including agent architecture, knowledge representation and reasoning, problem solving and planning, game playing, and learning.

CS 572. Machine Learning. 4 Credits.
A broad introduction to machine learning and its established algorithms. Topics include concept learning, decision trees, neural network.

CS 573. Probabilistic Methods for Artificial Intelligence. 4 Credits.
Fundamental techniques for representing problems as probability distributions, performing inference, and learning from data. Topics include Bayesian and Markov networks, variable elimination, loopy belief propagation, and parameter.

CS 601. Research: [Topic]. 1-16 Credits.
Repeatable.

CS 602. Supervised College Teaching. 1-5 Credits.
Repeatable.

CS 603. Dissertation. 1-16 Credits.
Repeatable.

CS 604. Internship: [Topic]. 1-4 Credits.
Repeatable.

CS 605. Reading and Conference: [Topic]. 1-16 Credits.
Repeatable.

CS 606. Field Studies: [Topic]. 1-16 Credits.
Repeatable.

CS 607. Seminar: [Topic]. 1-5 Credits.
Repeatable. Research topics are presented.

CS 608. Workshop: [Topic]. 1-16 Credits.
Repeatable.

CS 609. Terminal Project. 1-16 Credits.
Repeatable. Final project for master's degree without thesis.

CS 610. Experimental Course: [Topic]. 1-5 Credits.
Repeatable.

CS 621. Algorithms and Complexity. 4 Credits.
Design and analysis of algorithms, strategies for efficient algorithms, introduction to complexity theory including NP-completeness.
Prereq: CS 520 recommended.

CS 624. Structure of Programming Languages. 4 Credits.
Introduction to axiomatic, operational, and denotational semantics. Environments, stores, and continuations. Type theory, subtypes, polymorphism, and inheritance. Functional and logic programming.

CS 630. Distributed Systems. 4 Credits.
Principles of distributed computer systems: interprocess communication, distributed file systems, distributed timing and synchronization, distributed programming, transactions, process scheduling, distributed shared memory.
Prereq: CS 529.

CS 631. Parallel Processing. 4 Credits.
Advanced topics in parallel processing including massively parallel computer architecture, supercomputers, parallelizing compiler technology, performance evaluation, parallel programming languages, parallel applications.
Prereq: CS 529.

CS 632. Computer Networks. 4 Credits.
Advanced issues in computer networks, focusing on research to extend the services offered by the Internet.
Prereq: CS 532.

CS 633. Advanced Network Security. 4 Credits.
Classic and state-of-the-art research topics in network security; threats and attacks, defense algorithms and mechanisms, measurement and evaluation of both security problems and solutions. Offered alternate years.
Prereq: CS 533.

CS 640. Writing in Computer Research. 2 Credits.
Students learn to provide and accept constructive criticism of writing samples in a workshop format.

CS 670. Data Science. 4 Credits.
Data science is the development of methods to study large and complex data sets. Methods that scale to very large data sets are of particular interest. This course introduces state-of-art data science methods focused on processing very large data sets of real-world data.
Prereq: CS 551.