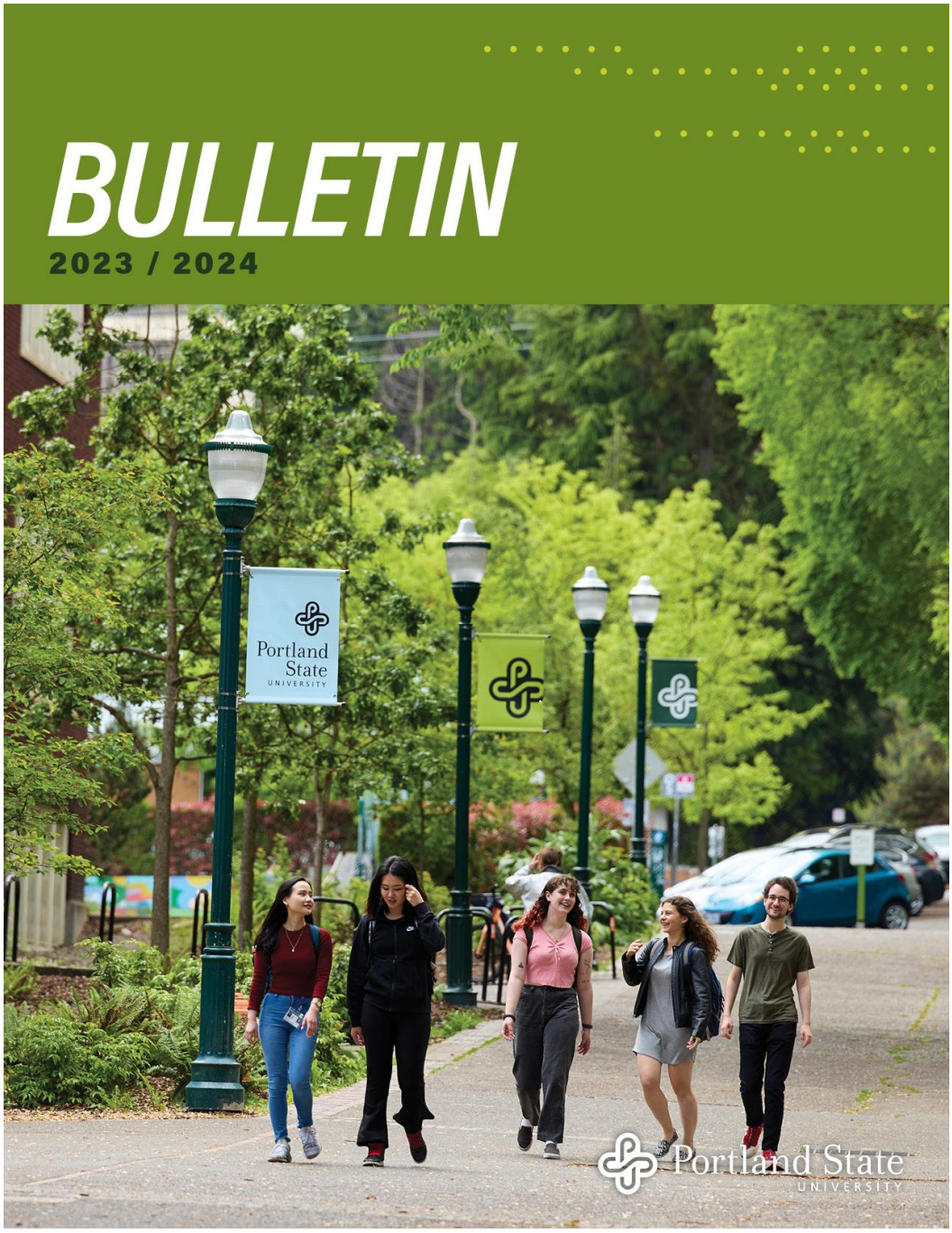
PORTLAND STATE UNIVERSITY

2023-2024 BULLETIN



MASEEH COLLEGE OF ENGINEERING AND COMPUTER SCIENCE

Joseph Bull, Dean
Chris Monsere, Associate Dean for Academic Affairs
Antonie Jetter, Associate Dean for Research
James Hook, Associate Dean for International Partnerships
Tong Zhang, Assistant Dean for Inclusive Innovation
TBD, Assistant Dean for Finance & Administration
Suite 500, Engineering Building
www.pdx.edu/engineering/

- B.S.—Civil Engineering, Computer Engineering, Computer Science, Electrical Engineering, Environmental Engineering and Mechanical Engineering
- Minor in Computer Science, Electrical Engineering, Environmental Engineering
- M.S.—Civil and Environmental Engineering, Computer Science, Electrical and Computer Engineering, Engineering and Technology Management, Mechanical Engineering, and Materials Science and Engineering
- M.Eng.—Civil and Environmental Engineering
- Ph.D.—Civil and Environmental Engineering, Computer Science, Electrical and Computer Engineering, Mechanical Engineering, Technology Management
- Ph.D.—Participating college in Systems Science Doctoral Program
- Ph.D.—Participating college in Environmental Sciences and Resources Doctoral Program
- Graduate Certificates

Engineering and computer science professions will continue to be front and center in addressing grand challenges that affect society and the planet, from reinventing the built environment for human health to low-impact energy sources, and resilient physical and cyber systems to effective transportation networks. For these reasons, national projections indicate that the need for engineers and computer scientists will continue to increase significantly during the years ahead, with commensurate increases in what are already high wages and opportunities to make substantial positive differences in the world.

All undergraduate programs require a core of engineering or computer science, mathematics, science, and liberal arts courses. Graduate programs provide extended educational opportunities in various engineering and computer science specialties, from cyber-security to healthy buildings, environmental monitoring to resilient infrastructure, technology management to nano-fabrication, water resources to artificial intelligence, neurosystems and robotics to power systems, and more.

Undergraduate programs

Our undergraduate programs are accredited as follows:

- Civil Engineering program is accredited by the Engineering Accreditation Commission of ABET, https://www.abet.org, under the General Criteria and the Civil and Similarly Named Engineering Program Criteria.
- Environmental Engineering program is accredited by the Engineering Accreditation Commission of ABET, https://www.abet.org, under the General Criteria and the Environmental Engineering and Similarly Named Engineering Program Criteria.
- Electrical Engineering program is accredited by the Engineering Accreditation Commission of ABET, https://www.abet.org, under the General Criteria and the Electrical, Computer, Communications, Telecommunication(s) and Similarly Named Engineering Program Criteria.
- Computer Engineering program is accredited by the Engineering Accreditation Commission of ABET, https://www.abet.org, under the General Criteria and the Electrical, Computer, Communications, Telecommunication(s) and Similarly Named Engineering Program Criteria.
- Mechanical Engineering program is accredited by the Engineering Accreditation Commission of ABET, https://www.abet.org, under the General Criteria and the Mechanical and Similarly Named Engineering Criteria.
- Computer Science program is accredited by the Computing Accreditation Commission of ABET, https://www.abet.org, under the General Criteria and the Computer Science and Similarly Named Computing Program Criteria

DEGREE MAPS AND LEARNING OUTCOMES

Please refer to each department to view the degree maps and expected learning outcomes for Engineering and Computer Science's undergraduate degrees.

ADMISSION REQUIREMENTS

Policy on admission to undergraduate programs

Students may declare engineering or computer science as their major at any time after enrolling at Portland State University. However, students must be admitted formally to a specific degree program in civil engineering, computer engineering, computer science, electrical engineering, environmental engineering or mechanical engineering before they will (1) be allowed to enroll in restricted upper-division courses offered by the program and (2) be graduated from that program. Students apply for formal department admission one to two terms before completing all eligibility requirements. Specific department application deadlines, criteria for admission and applications are available at the Maseeh College website: www.pdx.edu/engineering/upper-division-admissions.

Students transferring from other institutions who want to be admitted formally to a specific engineering degree program (civil engineering, computer engineering, computer science, electrical engineering, environmental engineering, mechanical engineering) must:

- Meet all eligibility requirements.
- Apply for admission to PSU.
- Apply for program admission to the Maseeh College of Engineering and Computer Science

Transfer students should consult the information and transfer guides posted on the Maseeh College website: www.pdx.edu/engineering/transfer-admissions

Transfer courses that are not evaluated by the Office of the Registrar or specified in other MCECS agreements as discrete numbered/direct equivalent courses will be evaluated by the department chair or their designee. In addition to the transcript, the student requesting the specific course equivalency may be asked to provide catalog descriptions and/or documents certifying course content. To ensure that the student is well prepared for the current curriculum, course equivalency will be evaluated against the content of the current course. Appeals of transfer course equivalence may be made to the MCECS Associate Dean.

Please see department websites for more specific admissions information.

Graduate programs

The Maseeh College of Engineering and Computer Science offers graduate programs leading to the degrees of Master of Science, Master of Engineering, Master of Software Engineering, and Doctor of Philosophy.

Master's programs are available in civil and environmental engineering, computer science, software engineering, electrical and computer engineering, mechanical engineering, engineering & technology management, materials science and engineering, and systems engineering.

Ph.D. programs are available in civil and environmental engineering, computer science, electrical and computer engineering, mechanical engineering, and technology management.

Graduate Certificates are also available in select departments.

Master of Software Engineering

Suite 120
Fourth Avenue Building

- M.S.E.—Master of Software Engineering
- Graduate Certificate in Software Engineering

Applications to the Master of Software Engineering and the Graduate Certificate in Software Engineering have been suspended pending a major curriculum revision.

Systems Engineering

- M.Eng.— Systems Engineering
- Graduate Certificate

Systems Engineering is the practice of creating the means of performing useful functions through the combination of two or more interacting elements. Systems engineering focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then continuing with design synthesis and system validation while considering the complete problem. Systems engineering integrates all the disciplines and specialty groups into a team effort, forming a structured development process that proceeds from concept to production to operation. Many of us already practice systems engineering, but call it something else: design or development of product, process, service. This course of study will enable the engineer to function in an interdisciplinary team and apply their area of engineering specialty toward the development of a product, process, or service.

ADMISSION REQUIREMENTS

Master of Engineering and Graduate Certificate

In addition to meeting general University admission requirements (p. 44), applicants to the program need a minimum of three years of professional experience, baccalaureate degree, and at least 2.50 GPA. Admission is based on program approval by the Director of Systems Engineering and the PSU Admissions office.

8. Submit the written dissertation in compliance with University guidelines and deadlines.

Internship credits (CE 604) require a project and final report; these credits must be arranged in advance between the CEE faculty advisor and the student. CEE courses for which the student receives a grade of "C+" or lower will not be counted toward fulfilling the requirements. Grades of C+, C, or C- may sometimes be counted toward the degree with the approval of the student's advisor and the Graduate Program Chair.

All courses taken in the Department of Civil and Environmental Engineering by degree candidates must be taken for a letter grade, unless a course is only offered with a pass/no pass option. Courses outside the Department of Civil and Environmental Engineering may be taken pass/no pass only with the consent of the student's advisor. Non-degree seeking students may take Civil and Environmental Engineering courses pass/no pass with the consent of the instructor.

Departmental policies and other helpful information for graduate students can be found in the Department's Graduate Handbook, located on the CEE website. All other degree requirements for the PhD program are established by PSU's Graduate School. Please refer to the Graduate School for the university's doctoral degree requirements.

HYDROLOGY GRADUATE CERTIFICATE

The Graduate Certificate of Hydrology (p. 265) is designed to give students advanced training in hydrology, and leads to professional certification with the American Institute of Hydrology (AIH). Additional information about the certificate can be found on the Environmental Science and Management Department website.

SUSTAINABILITY GRADUATE CERTIFICATE

The Graduate Certificate in Sustainability (p. 404) offers an integrated series of post-baccalaureate courses that allow students to deeply explore and understand the three spheres of sustainability: social, economic, and environmental. The courses cover theory as well as practice, providing experience analyzing real-world approaches and solutions. Courses can be taken by students admitted solely to the certificate program or concurrently enrolled in masters and doctoral programs at PSU. The certificate is administered by the Economics department. More information about the certificate and application procedures can be found on the Economic Department's website.

TRANSPORTATION GRADUATE CERTIFICATE

The Graduate Certificate in Transportation (p. 426) is a 21 credit hour program designed to build the technical and analytical knowledge of those who are in or wish to enter the transportation field. This program could be completed in a single year on a full-time basis or over two years on a part-time basis. The certificate includes courses from the Toulan School of Urban Studies and Planning and the Department of Civil and Environmental Engineering. Credits taken as part of this certificate program may be used to satisfy partial M.S. degree requirements in either program. Admission to this program will require an undergraduate degree at an accredited university and a GPA that meets university admission requirements. More information about the certificate and application procedures can be found on the School of Urban Studies and Planning website.

Computer Science

120 Fourth Avenue Building
503-725-4036
www.pdx.edu/computer-science/

- B.S.—Computer Science
- Minor in Computer Science
- M.S.—Computer Science
- Ph.D.—Computer Science
- Graduate Certificate in Computer Security

The Department of Computer Science offers a full range of courses and degree programs that are designed to provide students with the educational background to achieve a career in the computing industry. We offer a community to learn, discover, innovate, and share a curriculum based on the application and theoretical foundations of Computer Science. Our faculty members specialize in a variety of research areas such as artificial intelligence & machine learning; computer science education; computer security & privacy; computer vision & computer graphics; data science; natural language processing; programming languages & formal methods; software engineering; systems & networking; and theory.

Undergraduate program

The undergraduate computer science program is designed to provide students with the educational background required for a professional career in the computing industry and for further study at the graduate level. The program includes a core of required courses and an elective program of courses over a wide range of topics. Seniors work in teams to carry out community-based projects during the two-term capstone course in software engineering.

The undergraduate Computer Science program is accredited by the Computing Accreditation Commission of ABET, https://www.abet.org, under the General Criteria and the Computer Science and Similarly Named Computing Program Criteria.

DEGREE MAPS AND LEARNING OUTCOMES

To view degree maps for the undergraduate program, visit www.pdx.edu/academics/programs/undergraduate/computer-science. Expected learning outcomes are listed at www.pdx.edu/computer-science/program-objectives-outcomes.

PROGRAM OBJECTIVES

The objectives of the undergraduate program in computer science are to produce graduates with:

- The knowledge and skills necessary for career success.
- A commitment to uphold the highest standards of professionalism, integrity, and ethical behavior.
- The motivation and preparation to adapt to rapidly changing technology, and to engage in life-long learning.

ADMISSION REQUIREMENTS

Students who are intending to graduate with an undergraduate degree in Computer Science must be admitted to Portland State University and complete an opt-in for the Computer Science major after satisfying the lower-division requirements described below. Students with questions should contact the Computer Science Department. No more than 8 upper-division CS course credits (including any approved upper-division transfer credits) taken prior to admission to the program will be counted toward the student's departmental requirement of upper-division CS courses. Students also must be in admitted status during the term they intend to graduate.

Computer Science Admission Requirements

Applies to students pursuing a B.S. in Computer Science wishing to enroll in upper-division CS courses for the 2023-24 academic year.
Computer Science Department website:
www.pdx.edu/computer-science

Terms of Admission & Deadlines

- Completing an opt-in for admission to the Computer Science major requires a copy of the student's DARS report, showing that they have passed all of the courses that are required for admission.
- Although there are no formal deadlines, students are strongly encouraged to complete the opt-in as soon as possible in the term after they have satisfied the admission requirements, so as to avoid delays in being able to register for upper-division CS courses for the following term.
- The opt-in form is available from the Computer Science Department website at www.pdx.edu/computer-science/undergraduate-admission.
- A detailed four-year course plan/sample schedule is provided by the "Computer Science Blue Sheet", which is available at www.pdx.edu/computer-science/bachelor.

Requirements for Admission to the Computer Science Major

1. Completion of each of the following core CS courses with a C or better.

- CS 161 Introduction to Programming and Problem Solving (4) (This requirement will be waived for students who have completed CS 162 prior to Fall 2022)
- CS 162 Intro to Computer Science (4)
- CS 163 Data Structures (4)
- CS 205 System Programming and Architecture (4)
- CS 250 Discrete Structures I (4)
- CS 251 Discrete Structures II (4)

2. Completion of each of the following non-CS courses with a grade of C- or better.

- MTH 251 Calculus I
- MTH 252 Calculus II or MTH 261 Linear Algebra
- Three Approved Laboratory Science courses

Prior to admission, PSU students are expected to complete the Freshman and Sophomore Inquiry series. Similarly, transfer students are expected to complete the Maseeh College lower division general education requirements. Completing the general education requirements prior to admission will allow students to focus on CS major courses and likely result in a shorter time to graduation.

Appeals

- Students denied admission to the Computer Science program may submit a written appeal which will then be reviewed by the department's Appeals Committee. All decisions are final.

Priority Registration

- Most applicants will have completed 90 credits by the time they complete the opt-in. This gives them registration priority over students that have completed less than 90 credits. Students who complete the opt-in with less than 90 credits may find classes are full by the time they are able to register. The Computer Science Department is unable to provide overrides to allow earlier registration.

Pass/No Pass

- All required classes must be taken for a grade (not P/NP) unless they are only offered as P/NP.
- There is no GPA penalty for a Pass or No Pass (but the student may be required to retake the class for a letter grade).

Additional Information (exceptions, preferences, etc.)

- No preference is given to PSU students versus students who completed required courses elsewhere.

Continuation Criteria

- Admitted CS undergraduate students who are not making acceptable progress towards their degree requirements will be dropped from the program and required to reapply for admission. Acceptable progress is defined as completion of at least eight credits of coursework with acceptable grades (C or better for required CS courses, C- or better for required non-CS courses), satisfying departmental requirements, over the preceding academic year. Readmission will be determined by the CS Undergraduate Committee.

Prerequisite Policy

- In order to enroll in most upper-division or graduate CS courses, students must be admitted to the Computer Science program or have the instructor's permission. Before enrolling in any CS course, students should read the course descriptions and ensure that they have completed all prerequisites with a C or better for undergraduate courses, or a B or better for graduate courses. Students who have not met this requirement or who do not meet applicable admission requirements may be administratively dropped from the course.
- A limited set of upper division CS courses, designated "dual use courses", may be taken prior to admission, subject to completion of the associated prerequisites. Students should not take a total of more than 8 credits of upper division CS courses, including any dual use courses, before they have been admitted to the CS major. To avoid a gap in available courses that could delay progress to graduation, students should not generally take dual use courses before they have completed the admission requirements. Students are strongly encouraged to consult the CS Department website, or to meet with an advisor, for further details.

Laptop Requirement

- Students registering for upper-division (300- or above) CS courses, must have access to a laptop with wireless Internet access meeting a set of minimum requirements. These minimum requirements can be found on the computer science website at www.pdx.edu/computer-science/laptop. Chromebooks, iPads, and similar devices do not meet the requirements to run many applications that may be used in various CS courses.

Department Communication

- Academic year 2023-24 admission requirements are available the preceding academic year's spring term (Spring 2023).

COMPUTER SCIENCE B.S.

Majors in computer science must complete the following University and departmental degree requirements.

1. All computer science courses used to satisfy the departmental major must be graded C or better. Courses taken outside the department as part of departmental requirements must be graded C- or better. If a course is offered only on a Pass/No Pass (P/NP) grading scale, it must be graded as a Pass.
2. All courses specifically required by the department must be taken for a letter grade unless a required course is only offered with a Pass/No Pass option.
3. After admission to the computer science program, students are required to complete a minimum of 44 upper-division computer science credits in residence at PSU.
4. Freshmen entering with 29 or fewer prior university/college credits must complete all University Studies requirements, including freshman and sophomore inquiry sequences and upper-division cluster courses.
5. Transfer students must have a minimum of 39 credits of University Studies courses and/or arts and letters/social science courses prior to graduation; 12 of these credits are upper-division cluster courses that must be taken at PSU. Transfer students should consult with the CS departmental adviser for more information.

The following is a sample curriculum. Students choosing to make modifications to this schedule are urged to consult with an adviser.

REQUIREMENTS

Freshman year

CS 161	Introduction to Programming and Problem-Solving	4
CS 162	Introduction to Computer Science	4
Mth 251	Calculus I	4
Mth 252	Calculus II	4
	Approved Laboratory Science	15
	Freshman Inquiry	15
		Subtotal: 46

Sophomore year

CS 163	Data Structures	4
CS 205	System Programming and Architecture	4
CS 250	Discrete Structures I	4
CS 251	Discrete Structures II	4
Mth 253	Calculus III	4
	or	
Mth 261	Introduction to Linear Algebra	4
	Sophomore Inquiry	8
	Free electives	16
		Subtotal: 44

Junior year

CS 302	Programming Methodologies and Software Implementation	4
CS 314	Elements of Software Engineering	4
CS 333	Introduction to Operating Systems	4
CS 350	Algorithms and Complexity	4
CS 486	Introduction to Database Management Systems	4
	Approved upper-division programming intensive CS elective	4
	Approved Computer Science security elective	4
ECE 341	Introduction to Computer Hardware	4
Wr 227Z	Technical Writing	4
	Upper-division cluster	12
		Subtotal: 48

Senior year

CS 305	Social, Ethical, and Legal Implications of Computing	2
CS 358	Principles of Programming Languages	4
CS 469	Software Engineering Capstone I	3
CS 470	Software Engineering Capstone II	3
	Approved upper-division computer science electives	16
	Approved Math electives	7
	Free electives	7
		Subtotal: 42

Note: The University requires all students to have a minimum of 62 upper-division credits to graduate. Since fewer than 62 upper-division credits are required in computer science, mathematics, and general education for the computer science major, the extra credits of upper-division work must be taken from either the approved math or the free electives.

UPPER DIVISION CREDITS

Approved Computer Science security elective

Students must complete one course related to the principles and practices for secure computing. The current list of approved courses includes:

CS 491	Introduction to Computer Security	4
CS 495	Web and Cloud Security	4
CS 496	Network Security	4

Approved upper-division Computer Science electives

The total may include any regular 300- and 400-level computer science course, and any of the courses:

ECE 485	Microprocessor System Design	4
ECE 486	Computer Architecture	4

except that no more than a total of 4 credits may be taken from:

CS 401	Research	1-6
CS 405	Reading and Conference	1-6
CS 406	Special Projects	1-6
CS 407	Seminar	1-6
CS 409	Practicum	1-9

At least 4 credits of approved "Programming Intensive" courses must be taken. These courses can be identified by the "P" suffix in the course number (e.g., CS 410P, CS 465P, CS 494P, etc.). Additionally, CS 404, University Studies courses, and courses specifically described as not being applicable to the CS degree may not be used.

Approved Laboratory Science

Students must select 3 courses from the following, including their associated laboratories:

Biology:

Bi 211	Principles of Biology: Biology of Cells	4
Bi 212	Principles of Biology: The biology of organisms	4
Bi 213	Principles of Biology: Evolution and ecology of living organisms	4

	With	
Bi 214	Principles of Biology Lab I	1
Bi 215	Principles of Biology Lab II	1
Bi 216	Principles of Biology Lab III	1
Chemistry:		
Ch 221	General Chemistry I	4
Ch 222	General Chemistry II	4
Ch 223	General Chemistry III	4
	With	
Ch 227	General Chemistry Laboratory	1
Ch 228	General Chemistry Laboratory	1
Ch 229	General Chemistry Laboratory	1
Geology:		
G 201	Dynamic Earth: Interior	3
G 202	Dynamic Earth: Surface	3
	With	
G 204	Geology Laboratory	1
G 205	Geology Laboratory	1
Physics:		
Ph 201	General Physics	4
Ph 202	General Physics	4
Ph 203	General Physics	4
Ph 211	General Physics (with Calculus) I	4
Ph 212	General Physics (with Calculus) II	4
Ph 213	General Physics (with Calculus) III	4
Ph 221	General Physics (with Calculus) I	3
Ph 222	General Physics (with Calculus) II	3
Ph 223	General Physics (with Calculus) III	3
Ph 231	General Physics I with Life Science and Medical Applications	4
Ph 232	General Physics II with Life Science and Medical Applications	4
Ph 233	General Physics III with Life Science and Medical Applications	4
	With	
Ph 214	Lab for Ph 201 or Ph 211 or Ph 221 or Ph 231	1
Ph 215	Lab for Ph 202 or Ph 212 or Ph 222 or Ph 232	1
Ph 216	Lab for Ph 203 or Ph 213 or Ph 223 or Ph 233	1

Credit is only given for General Physics Ph 201-Ph 203, Ph 211-Ph 213, Ph 221-Ph 223, Ph 231-Ph 233 one time. You may not receive duplicate credit.

Approved Mathematics electives

Students must complete 7 or more credits of approved mathematics electives. The current list of approved courses includes:

Mth 261	Introduction to Linear Algebra	4
Mth 311	Introduction to Mathematical Analysis I	4
Mth 343	Applied Linear Algebra	4
Mth 344	Introduction to Group Theory and Applications	4
Mth 346	Number Theory	4
Mth 356	Discrete Mathematics	4
Mth 457	The Mathematical Theory of Games I	3
Mth 458	The Mathematical Theory of Games II	3
Mth 461	Graph Theory I	3
Mth 462	Graph Theory II	3
Stat 366	Introduction to Experimental Design	4
Stat 451	Applied Statistics for Engineers and Scientists I	4
Stat 452	Applied Statistics for Engineers and Scientists II	3
Stat 464	Applied Regression Analysis	3
Stat 467	Applied Probability I	3
Stat 468	Applied Probability II	3

Other upper-division mathematics or statistics courses may be used to satisfy the requirement with prior written approval from the Computer Science Undergraduate Adviser.

COMPUTER SCIENCE MINOR

A minor in computer science is available within the Maseeh College of Engineering and Computer Science in the area of computer science.

REQUIREMENTS

To earn a minor in computer science, a student must complete 24 credits as follows:

Courses

CS 161	Introduction to Programming and Problem-Solving	4
CS 162	Introduction to Computer Science	4
CS 163	Data Structures	4
CS 205	System Programming and Architecture	4
CS 250	Discrete Structures I	4
	Approved CS Minor course	4

Only grades of C or better count toward departmental requirements. At least 12 of the required 24 credits must be taken at Portland State University.

Approved CS Minor courses include any lower-division or upper-division Computer Science course. Some upper-division courses are reserved for Computer Science majors and may require permission from the instructor in order to register.
Subtotal: 24

COMPUTER SCIENCE - HONORS TRACK

The honors degree in computer science requires the writing of an honors thesis. Details about the program can be found on the computer science website at www.pdx.edu/computer-science/honors-track.

BIOMEDICAL INFORMATICS PROGRAM

Portland State University and Oregon Health & Science University offer an accelerated, collaborative degree program in biomedical informatics. Designed for high achieving freshmen, this program combines courses from both schools to award a B.S. in computer science and Master of Biomedical Informatics at the end of five years. Details about the program can be found on the computer science website at www.pdx.edu/computer-science/biomedical-informatics.

Graduate Programs

ADMISSIONS REQUIREMENTS

To be considered for admission to the graduate program in computer science, the student must have a four-year baccalaureate degree from an accredited institution. Prospective graduate students are not required to have a bachelor's degree in Computer Science, but they must make up the background needed for graduate study before applying for admission. The Grad Prep program described at https://www.pdx.edu/computer-science/grad-prep may be helpful in developing knowledge of the core curriculum of an undergraduate computer science degree.

A cumulative undergraduate GPA of at least 3.00 is required. See the CS graduate admissions webpage for the admission and application requirements: https://www.pdx.edu/computer-science/graduate-admission.

Students may apply to the Ph.D. program with or without having already obtained an M.S. in computer science.

COMPUTER SCIENCE M.S.

The Master's program in computer science is designed to prepare students for advanced careers in the computer industry, to create a research environment in computer science, and to prepare students for graduate work at the Ph.D. level.

See University Master's degree requirements. The Master's program in computer science consists of two options. The first option involves the completion of an approved program of 45 credits. The second option requires the completion of an approved program of 45 credits, which includes 6 to 9 credits of thesis. In both options, the coursework includes core courses in theory and programming practice, plus a 9-credit concentration in one of the tracks described below. For the thesis option, successful completion of a final oral examination covering the thesis is required. For more information, visit: https://www.pdx.edu/computer-science/master.

CORE COURSE REQUIREMENT

One theory course from:

CS 581	Theory of Computation	3
CS 584	Algorithm Design and Analysis	3
CS 578	Programming Language Semantics	3

Subtotal: 3

One programming practice course from:

CS 558	Programming Languages	3

Or any 500-level course designated by the department as a "Programming Intensive" course, as indicated by the "P" suffix in the corresponding 400-level course number.
Subtotal: 3

ELECTIVES

Students must take enough electives to complete 45 total credits for the Master's degree. Electives can be any 500-level CS course and may include up to 6 credits of CS 505 (Reading and Conference) and CS 506 (Special Projects). CS 501 Research, CS 502 Independent Study, CS 504 Internship, and CS 509 Practicum credits cannot be applied. A limited number of credits taken outside Computer Science can count towards the elective requirements, with advisor approval. A minimum of 30 credits must be taken in Computer Science at Portland State University. Given this, students may use a combined total of 15 pre-admission, transfer, and non-CS credits toward their Master's degree with advisor approval. One additional credit beyond the 15 credit limit can be used provided none of the courses are a 1-credit course.

- Pre-admission credits (taken before the term of formal admission) can include both transfer and PSU credits. Pre-admission credits taken at PSU are requested via a DARS exception submitted to the Graduate School. This request should be made soon after admission to the graduate program.

- Transfer credits refer to credits taken from another institution other than PSU. To request approval of transfer credits, complete and submit the GO-21M form (Proposed Transfer Credit) to the CS Graduate Advisor. Students should submit the GO-21 form during the first term of enrollment in the program, so there is sufficient time to complete any additional coursework that may be necessary. Any transfer credits must be approved before graduation paperwork can be processed. OHSU joint campus credits are considered transfer credits and are transferred via a different process. For more information, visit: www.pdx.edu/gradschool/joint-campus-registration.
- Non-CS credits taken outside of Computer Science, such as ECE or Math, can count towards elective requirements once approved. Students should obtain advisor approval in advance to avoid the risk of taking a course that will not be approved. To request approval, submit a plan of study with the courses listed to the Graduate Advisor. Non-CS courses must be graduate level. All ETM courses are eligible to transfer but students are limited to using only one for the CS degree requirements.

Subtotal: 30

TRACK REQUIREMENT

Take three courses from one of the following tracks:

Databases

Covers concepts, languages, implementation and application of database management systems. Other topics that have been offered in the track include formal foundations of databases, databases for cloud and cluster environments, and data stream systems.

CS 586	Introduction to Database Management Systems	3

And two courses from the following*:

CS 530	Internet, Web, & Cloud Systems	3
CS 587	Database Management Systems Implementation	3
CS 588	Cloud and Cluster Data Management	3
CS 589	Blockchain Development & Security	3

*Or any approved CS 510 course in Databases.

Languages and Programming

Focuses on the design, implementation, and use of programming languages. It includes exposure to a variety of programming paradigms, experience using programming languages to express the essential abstractions of a problem domain, courses on programming language implementation, and the study of formal methods for specifying and reasoning about programs and programming languages.

CS 558	Programming Languages	3

Two courses from the following*:

CS 515	Parallel Programming	3
CS 520	Object-Oriented Programming & Design	3
CS 553	Design Patterns	3
CS 557	Functional Programming	3
CS 578	Programming Language Semantics	3

*Or any approved CS 510 course in Languages and Programming.

Security

Focuses on protecting computing systems and user data from unauthorized access and use. Topics include cryptography, network and host-based access control, vulnerability analysis, penetration testing, and reverse engineering.

CS 591	Introduction to Computer Security	3

Two courses from the following*:

CS 530	Internet, Web, & Cloud Systems	3
CS 576	Computer Security Research Seminar	3
CS 585	Cryptography	3
CS 592	Malware Reverse Engineering	3
CS 593	Digital Forensics	3
CS 595	Web and Cloud Security	3
CS 596	Network Security	3

*Or any approved CS 510 course in Security.

Software Engineering

Studies the principles, processes, techniques, and tools for building software systems. Topics include software requirement, design, development, validation, and maintenance.

CS 554	Software Engineering	3

Two courses from the following*:

CS 530	Internet, Web, & Cloud Systems	3
CS 552	Building Software Systems with Components	3
CS 553	Design Patterns	3
CS 555	Software Specification and Verification	3
CS 556	Software Implementation and Testing	3

CS 561	Open Source Software Development Laboratory	3

*Or any approved CS 510 course in Software Engineering.

Systems and Networking

Studies the design and implementation of operating systems, wired and wireless computer networks including high performance computer systems, data centers, cloud computing architectures, distributed systems, fault tolerance, concurrency, systems programming, and theoretical topics related to these areas.

CS 533	Concepts of Operating Systems	3
CS 594	Internetworking Protocols	3

One course from the following*:

CS 515	Parallel Programming	3
CS 530	Internet, Web, & Cloud Systems	3
CS 531	Introduction to Performance Measurement, Modeling and Analysis	3
CS 535	Accelerated Computing	3
CS 538	Computer Architecture	3
CS 572	Operating System Internals	3
CS 590	Introduction to Multimedia Computing and Networking	3
CS 598	Introduction to Wireless Network Protocols	3
ECE 586	Computer Architecture	4

*Or any approved CS 510 course in Systems and Networking.

Artificial Intelligence and Machine Learning

Covers modern algorithms underlying intelligent and learning systems. Examples of topics covered in this track include knowledge representation, planning, reasoning, combinatorial and adversarial search methods, natural language processing, computer vision, statistical machine learning, and evolutionary and reinforcement learning.

CS 541	Artificial Intelligence	3
CS 545	Machine Learning	3

One course from the following*:

CS 542	Advanced Artificial Intelligence: Combinatorial Games	3
CS 543	Advanced Artificial Intelligence: Combinatorial Search	3
CS 546	Advanced Topics in Machine Learning	3
CS 570	Machine Learning Seminar	1
Stat 671	Statistical Learning I	3
Stat 672	Statistical Learning II	3
Stat 673	Statistical Learning III	3

*Or any approved CS 510 course in Artificial Intelligence or Machine Learning.
Subtotal: 9

Total Credit Hours: 45

Cumulative Graduate GPA

Students must have a graduate GPA of 3.0 or above in all graduate level coursework taken at PSU to graduate from a master's degree, doctoral degree, or graduate certificate program at PSU.

Degree Program GPA

Students must have a GPA of 3.0 or above in all courses being used to meet the degree requirements. All graded courses must be passed with a grade of B- or better. Core requirements must be passed with a grade of B or better. All courses taken Pass/No Pass must be passed. Students must have the minimum number of credits needed for their degree before they can graduate.

Application for Graduation

Students must apply for graduation no later than the first Friday of the term in which they wish to graduate. Information about applying for graduation can be found at: https://www.pdx.edu/gradschool/applying-for-graduation.

COMPUTER SCIENCE PH.D.

The doctoral degree program in Computer Science is designed to prepare students for advanced research or university teaching in the field.

See University doctoral degree requirements. The student must complete an approved program of 90 graduate credits, including 18 credits of core courses and 27 credits of dissertation research. To be admitted to Ph.D. candidacy, a student must pass the Research Proficiency Examination (RPE) and must present an acceptable dissertation proposal. The dissertation comprises original research work, which is expected to be of a quality meriting publication in a refereed journal or conference. For more information, visit: www.pdx.edu/computer-science/doctor-philosophy-phd.

CYBERSECURITY GRADUATE CERTIFICATE

The cybersecurity certificate program requires admission as a graduate student, similar to admission to the Master’s program, in the Computer Science department. The program requires 21 total credits of graduate classes. There are two core classes for a total of 6 credits. In addition, five elective classes must be taken for the needed additional 15 credits. In summary, seven total graduate classes must be taken; two are core and five are electives.

Required Core Courses		
CS 591	Introduction to Computer Security	3
CS 595	Web and Cloud Security	3
Subtotal: 6		

Five of the following courses*:		
CS 554	Software Engineering	3
CS 555	Software Specification and Verification	3
CS 556	Software Implementation and Testing	3
CS 576	Computer Security Research Seminar	3
CS 585	Cryptography	3
CS 592	Malware Reverse Engineering	3
CS 593	Digital Forensics	3
CS 594	Internetworking Protocols	3
CS 596	Network Security	3

*Or any CS 510 course in Security.
Subtotal: 15

Total Credit Hours: 21

Electrical and Computer Engineering

1900 SW Fourth Ave., Suite 160
503-725-3806
www.pdx.edu/electrical-computer-engineering/

- B.S.—Computer Engineering
- B.S.—Electrical Engineering
- Minor in Electrical Engineering
- M.S.—Electrical and Computer Engineering
- Ph.D.—Electrical and Computer Engineering

Mission, Vision & Values

Mission

We prepare students for successful engineering careers and lifelong learning, and we conduct research that creates new technologies and engineering knowledge.

Vision

Our vision is to be a source of premier electrical and computer engineering talent and high-impact research. This means our graduates are successful, our research is recognized worldwide, and we are the intellectual center for our discipline in the Portland region.

Values

We value

- The success of our graduates
- Contributions to research and knowledge creation
- High intellectual and ethical standards
- High quality education for traditional and nontraditional students
- A diverse student population
- Our contribution to the Oregon economy
- Lifelong learning
- Technical and professional relationships with the engineering community

Undergraduate programs

The Department of Electrical and Computer Engineering offers programs in electrical and computer engineering. Cooperative educational arrangements with Portland-area industries, government agencies, and engineering consulting offices are available to qualified students. Qualified freshmen are encouraged to participate in the University Honors Program. Qualified upper-division students should consider the Electrical and Computer Engineering departmental honors track as described below.

The Electrical Engineering and Computer Engineering programs are accredited by the Engineering Accreditation Commission of ABET, https://www.abet.org, under the General Criteria and the Electrical, Computer, Communications, Telecommunication(s) and Similarly Named Engineering Program Criteria.

DEGREE MAPS AND LEARNING OUTCOMES

To view the degree maps and expected learning outcomes for Electrical and Computer Engineering's undergraduate degrees, go to www.pdx.edu/academic-programs/a-z.

PROGRAM EDUCATIONAL OBJECTIVES

The electrical and computer engineering programs prepare our graduates for the following program educational objectives:

1. Graduates are expected to be employed as electrical or computer engineers or in related fields that benefit from an electrical and computer engineering education.
2. Graduates are expected to advance in their profession and engage in the professional community.
3. Graduates are expected to continue to learn and adapt in a world of constantly changing environment and technology.

ADMISSION REQUIREMENTS

CS - Computer Science

CS 105 - Computing Fundamentals I (4)

Intended as an overview of computers and computer technology for non-CS majors, this course is often described as a computer literacy course. The primary focus is on the personal computer and personal productivity software. Hardware components of computers such as processors, memory, and input/output devices are discussed and compared. Software is the primary focus of the course. The main topics are system software (Windows, OS X, etc) and applications (such as browsers, word processors, spreadsheets, presentation graphics and database managers). The course concludes with discussions concerning legal and ethical issues surrounding computer technology, management information systems, and systems analysis. Expected preparation: high school algebra.

CS 106 - Computing Fundamentals II (4)

Introduction to programming, appropriate for non-CS majors. Introduction to the logical thought processes and problem-solving strategies used when programming. Concepts presented include problem definition and requirements gathering, generating a description of a step-by-step solution (the algorithm), writing a program, testing, and documentation. The programming language Visual Basic is used; several programming projects are completed during the term. Expected preparation: high school algebra, knowledge of Windows and the ability to use Windows Explorer.

CS 107 - Computing Fundamentals III (4)

Introduction to Web programming and associated web tool usage for non-CS majors. Centering around the more sophisticated aspects of browsers. Web pages that represent the input to browsers are defined. In-depth study of HTML, VBScript and JavaScript. Brief exploration into CGI Scripts and other server-side tools. Course differentiates between Web page design (a graphics designer's role) and Web page programming, taking the results of their work and committing it to workable code. Recommended prerequisites: high school algebra and CS106 or some programming experience.

CS 161 - Introduction to Programming and Problem-Solving (4)

Introduction to fundamental concepts of computer science. Problem solving, algorithm and program design, data types, loops, control structures, subprograms, and arrays. Learn to write programs in a high level programming language. Surveys current social and ethical aspects of computer science. Recommended prerequisite: Mth 111Z.

CS 161L - Introduction to Programming and Problem-Solving Laboratory (0)

Lab for CS 161 Introduction to Programming and Problem-Solving.

CS 162 - Introduction to Computer Science (4)

The goals of this class are to teach the syntax of C++ to students who already know how to program. Students are expected to be proficient at using conditionals, I/O, loops, and functions with arguments. Topics include: conditionals, I/O, files, functions, classes, pointers, dynamic memory, linear linked lists, and multi-dimensional arrays in C++, as well as program correctness, verification, and testing. Three hours lecture and one 3-hour laboratory. The laboratory emphasizes practical programming skills.

Prerequisite: CS 161.

CS 162L - Lab for CS 162 (0)

Lab for CS 162.

Corequisite: CS 162.

CS 163 - Data Structures (4)

Data abstraction with formal specification. Elementary algorithm analysis. Basic concepts of data and its representation inside a computer. Linear, linked, and orthogonal lists; tree structures. Data structures are implemented as data abstractions using pointer based implementations. Sorting and search strategies. Data management. Three hours lecture and one 3-hour laboratory. The laboratory emphasizes practical programming skills.

Prerequisite: CS 162 with a grade of C or better. Co-requisites: concurrent enrollment in CS163L. Corequisite: CS 163L.

CS 163L - Lab for CS 163 (0)

Lab for CS 163.

Corequisite: CS 163.

CS 199 - Special Studies (0-12)

(Credit to be arranged.)

CS 205 - System Programming and Architecture (4)

Introduction to computer systems from a software perspective. Topics include: 1) Basic machine organization, 2) System programming using C and assembly language, 3) System tools such as compilers and debuggers, 4) Data representation (bits bytes, characters, integers, floating point numbers), 5) Implementation of control flow, procedure calls, and complex data types at machine level, 6) Improving program performance, and 7) Introduction to memory hierarchy.

Prerequisite: CS 162.

CS 250 - Discrete Structures I (4)

Introduces discrete structures and techniques for computing. Sets. Graphs and trees. Functions: properties, recursive definitions. Relations: properties, equivalence, partial order. Proof techniques, inductive proof. Counting techniques and discrete probability.

Prerequisite: CS 161 and Mth 112Z or passing at the necessary level on the mathematics placement test within the last year.

CS 251 - Discrete Structures II (4)

Continuation of CS 250. Logic: propositional calculus, first-order predicate calculus. Formal reasoning: natural deduction, resolution. Applications to program correctness and automatic reasoning. Introduction to algebraic structures in computing.

Prerequisite: CS 250.

CS 299 - Special Studies (0-4)

(Credit to be arranged.)

CS 299L - Lab for CS 299 (0)

Lab for CS 299.

CS 302 - Programming Methodologies and Software Implementation (4)

Introduces principles and techniques for producing high-quality software solutions to computational problems using modern programming languages. Important topics include: analysis of informal specifications and documentation; unit testing; abstract data types; object-oriented and functional programming design techniques; and use of software libraries. Laboratory exercises will include application of contemporary software tools, including integrated development environments, debuggers, version control, and build frameworks.

Prerequisite: CS 163, CS 205, CS 250. Corequisite: CS 302L.

CS 302L - Lab for CS 302 (0)

Lab for CS 302L.

Corequisite: CS 302.

CS 305 - Social, Ethical, and Legal Implications of Computing (2)

History of computing, social context of computing, professional and ethical responsibilities, risks and liabilities of safety-critical systems, intellectual property, privacy and civil liberties, social implications of the Internet, computer crime, economic issues in computing.

Prerequisite: a course in computer science at the 300 or higher level. Sophomore inquiry or a course in public speaking and a course in writing a research paper.

CS 311 - Computational Structures (4)

Introduces the foundations of computing. Regular languages and finite automata. Context free languages and pushdown automata. Turing machines and equivalent models of computation. Computability. Introduction to complexity. An appropriate programming language is used for programming experiments.

Prerequisite: CS 251.

CS 313 - Artificial Intelligence and Game Design (4)

Study of the basic principles of computer game design, the most popular techniques and technologies for game implementation, focusing on the many ways in which advances in artificial intelligence influences game design.

Prerequisite: Prior computer programming experience equivalent to CS 163.

CS 314 - Elements of Software Engineering (4)

Practical techniques of program development for medium-scale software produced by individuals. Software development from problem specification through design, implementation, testing, and maintenance. The fundamental design techniques of step-wise refinement and data abstraction. A software project will be carried through the development cycle.

Prerequisite: CS 302.

CS 333 - Introduction to Operating Systems (4)

Introduction to the principles of operating systems and concurrent programming. Operating system services, file systems, resource management, synchronization. The concept of a process; process cooperation and interference. Introduction to networks, and protection and security. Examples drawn from one or more modern operating systems. Programming projects, including concurrent programming.

Prerequisite: CS 302.

CS 333L - Introduction to Operating Systems Lab (0)

Lab for Introduction to Operating Systems.

Corequisite: CS 333.

CS 340 - Discrete Structures for Engineers (4)

A one-term introduction to discrete structures with applications to computing problems. Topics include sets, relations, functions, counting, graphs, trees, recursion, propositional and predicate logic, proof techniques, Boolean algebra. The course may not be used as part of the degree requirements for the BS degree in Computer Science.

Prerequisite: CS 163, Math 252.

CS 345 - Cyberculture: The Internet and Popular Culture (4)

Study of the effect of computers and the internet on popular culture. Graduates of the course will become more intelligent and successful users of the Internet, understand how the internet works, be aware of the wide variety of applications that exist on the internet, and will understand the primary principles that underlie the success the Internet has had in changing popular culture. Typical topics will include history and technologies of the web, social

networks, the long tail in business and culture, the power of groups, user generated content, complex systems, virtual worlds and the power of search.

Prerequisite: Sophomore Inquiry: Popular Culture (UNST 254).

CS 345 - Cyberculture: The Internet and Popular Culture (4)

Study of the effect of computers and the internet on popular culture. Graduates of the course will become more intelligent and successful users of the Internet, understand how the internet works, be aware of the wide variety of applications that exist on the internet, and will understand the primary principles that underlie the success the Internet has had in changing popular culture. Typical topics will include history and technologies of the web, social networks, the long tail in business and culture, the power of groups, user generated content, complex systems, virtual worlds and the power of search.

CS 346U - Exploring Complexity in Science and Technology (4)

Introduction to Complex Systems, an interdisciplinary field that studies how collections of simple entities organize themselves to produce complex behavior, use information, and adapt and learn. Focuses on common principles underlying complexity in science and technology, and includes ideas from physics, biology, the social sciences, and computer science. The course may not be used as one of the upper-division CS Electives for the BS degree in Computer Science. This course is the same as SySc 346; course may be taken only once for credit.

Cross-Listed as: SySc 346U.

CS 347U - The Internet Age (4)

Examination of the Internet and its evolution over the last 30 years to become an essential part of today's society. Also examines the impact the Internet has had on society as well as potential threats to its continued success. The course may not be used as one of the upper-division CS Electives for the BS degree in Computer Science.

CS 348U - Digital Media, Technology and Society (4)

Covers, from a computing perspective, the transition of society to one that is primarily digital. Provides an understanding of digital media, its technical limitations, copyright and digital rights management, and digital media communications. The course may not be used as an upper-division CS Elective for the BS degree in Computer Science.

CS 350 - Algorithms and Complexity (4)

Techniques for the design and analysis of algorithms. Case studies of existing algorithms (sorting, searching, graph algorithms, dynamic programming, matrix multiplication, fast Fourier transform.) NP-Completeness.

Prerequisite: CS 163, CS 250, Mth 251.

CS 358 - Principles of Programming Languages (4)

Syntax and semantics. Compilers and interpreters. Programs as data. Regular expressions and context free grammars. Programming paradigms, including procedural, functional, and object-oriented programming. Type systems, including dynamic and static typing disciplines. Binding, scope, data abstraction, and modularity. Denotational, operational, and axiomatic semantics. Introduction to program correctness.

Prerequisite: CS 302, CS 314, and CS 350.

CS 399 - Special Studies (0-6)

(Credit to be arranged.) Consent of instructor.

CS 399L - Lab for CS 399 (0)

Lab for CS 399 special studies.

CS 399P - Special Studies (1-6)
(Credit to be arranged.)

CS 401 - Research (1-6)
(Credit to be arranged.) Consent of instructor.

CS 402 - Independent Study (1-12)
(Credit to be arranged.)

CS 403 - Honors Thesis (1-4)
(Credit to be arranged.) Consent of instructor.

CS 404 - Cooperative Education/Internship (1-12)
(Credit to be arranged.) Consent of instructor.

CS 405 - Reading and Conference (1-6)
(Credit to be arranged.) Consent of instructor.

CS 406 - Special Projects (1-6)
(Credit to be arranged.) Consent of instructor.

CS 407 - Seminar (1-6)
(Credit to be arranged.) Consent of instructor.

CS 409 - Practicum (1-9)
(Credit to be arranged.) Consent of instructor.

CS 410 - Selected Topics (1-6)
(Credit to be arranged.) Consent of instructor.

CS 410L - Selected Topics Lab (0)
Lab for CS 410.

CS 410P - Selected Topics (1-6)
Programming intensive version of CS 410. (Credit to be arranged.) Consent of instructor.

CS 415 - Parallel Programming (4)

An introduction to parallel programming concepts and techniques. Topics include: parallel programming models and languages, share-memory programming, message-passing programming, performance models and analysis techniques, domain-specific parallel algorithms.

Also offered for graduate-level credit as CS 515 and may be taken only once for credit. . Prerequisite: CS 302 and CS 333.

CS 415L - Lab for CS 415P (0)

Lab for CS 415P Parallel Programming.

Corequisite: CS 415P.

CS 415P - Parallel Programming (4)

An introduction to parallel programming concepts and techniques. Topics include: parallel programming models and languages, share-memory programming, message-passing programming, performance models and analysis techniques, domain-specific parallel algorithms.

Prerequisite: CS 302 and CS 333. Corequisite: CS 415L.

CS 418 - Cultural Competence in Computing (4)

Students will learn about different identities (e.g., race, ethnicity, gender, class, sexuality, and ability), understand how algorithmic bias in technology affects different identities, and learn how to create accessible and equitable products and more inclusive communities at school, in industry, and beyond.

Also offered for graduate level credit as CS 518 and may be taken only once for credit. Prerequisite: Admission to the program.

CS 420 - Object-Oriented Programming and Design (4)

The fundamental concepts of object-oriented programming, including object-oriented modeling and design. The focus of the course will be to help students create programs that model their application domain, that exhibit that model to other programmers who read the code, and that are as a consequence maintainable and robust to change. Issues addressed may include data abstraction and modeling, the use and misuse of inheritance, higher-order data structures and their operations, reusability, refactoring, concurrency control, and usability. Includes programming assignments in an OO language.

Also offered for graduate-level credit as CS 520 and may be taken only once for credit. . Prerequisite: CS 358.

CS 420P - Object-Oriented Programming (4)

The fundamental concepts of object-oriented programming languages, including data abstraction and typing, class inheritance and generic types, prototypes and delegation, concurrency control and distribution, object-oriented databases, and implementation. To illustrate these issues, programming assignments in languages such as Smalltalk, Eiffel and C++ will be given.

Prerequisite: CS 358.

CS 421 - Programming Language Implementation: Syntax and Static Semantics (4)

Techniques and tools for construction of compiler and interpreter front-ends, including: representation of programs using abstract syntax trees; lexical analysis, and lexer generators; parsing (recursive descent, top-down, and bottom-up), and parser generators; type checking and static analysis. Design and implementation of a front-end for a small programming language.

Prerequisite: CS 205, CS 302, CS 314, and CS 358.

CS 421L - Lab for CS 421 (0)

Lab for CS 421.

CS 421P - Programming Language Implementation: Syntax and Static Semantics (4)

Techniques and tools for construction of compiler and interpreter front-ends, including: representation of programs using abstract syntax trees; lexical analysis, and lexer generators; parsing (recursive descent, top-down, and bottom-up), and parser generators; type checking and static analysis. Design and implementation of a front-end for a small programming language.

Prerequisite: CS 205, CS 302, CS 311, CS 314, and CS 358.

CS 422 - Programming Language Implementation: Code Generation and Dynamic Semantics (4)

Techniques and tools for construction of compiler and interpreter back-ends, including: interpreter design; code generation strategies for standard programming constructs; intermediate representations; optimization techniques; run-time organization, including functions, objects, and closures; run-time systems. Design and implementation of an interpreter and a compiler back-end for a small programming language.

Prerequisite: CS 205, CS 302, CS 311, CS 314, and CS 358.

CS 422L - Lab for CS 422 (0)

Lab for CS 422.

CS 422P - Programming Language Implementation: Code Generation and Dynamic Semantics (4)

Techniques and tools for construction of compiler and interpreter back-ends, including: interpreter design; code generation strategies for standard programming constructs; intermediate representations; optimization techniques; run-time organization, including functions, objects, and closures; run-time systems. Design and implementation of an interpreter and a compiler back-end for a small programming language.

Prerequisite: CS 205, CS 302, CS 314, and CS 358.

CS 430P - Internet, Web, & Cloud Systems (4)

Covers modern networked computing systems and the abstractions they provide. Specifically, students will learn about and apply their knowledge of topics such as Internet protocols, virtual machines and containers, web servers and frameworks, and databases as well as their deployment in modern cloud environments.

Also offered for graduate-level credit as CS 530 and may be taken only once for credit. Prerequisite: Upper-division standing and admission into the CS program.

CS 431 - Introduction to Performance Measurement, Modeling and Analysis (4)

A survey of the fundamentals of computer application and system performance. Hands on programming exercises will allow us to apply the techniques to increasingly complex problems. We will use a variety of state of the art tools for measurement, modeling, simulation, and analysis throughout the course.

Also offered for graduate-level credit as CS 531 and may be taken only once for credit. Prerequisite: CS 205 and CS 302 and CS 333.

CS 435 - Accelerated Computing (4)

Heterogeneous approaches that use special-purpose processors to accelerate the execution of a variety of applications. GPUs, Intel Xeon Phi, APUs, FPGUs. The sustainability implications of these platforms. Lectures, homeworks, labs, and group programming projects using NVIDIA GPUs and Intel Xeon Phi.

Also offered for graduate-level credit as CS 535 and may be taken only once for credit. Prerequisite: CS 333.

CS CS 435L - Lab for CS 435 (0)

Lab for CS 435.

Corequisite: CS 435.

CS 438 - Computer Architecture (4)

Processors, memory hierarchy, and bus systems. Multi-level caches and cache coherence in MP systems. Arithmetic algorithms. RISC vs. CISC instructions, pipelining, and software pipelining. Superscalar, super pipelined, and

VLIW architectures. Connection networks. Performance evaluation, simulation, and analytic models. Performance enhancement through branch prediction and out-of-order execution.

Also offered for graduate-level credit as CS 538 and may be taken only once for credit. . Prerequisite: CS 333.

CS 440 - Deep Learning: Computational Structures and Programming (4)

Deep learning is a powerful tool for machine learning systems. This class provides an introduction to this topic and will focus on classic as well as emerging deep learning techniques. Topics studied include multi-layer perceptrons, convolution neural networks, long-short term memory, attention mechanisms, autoencoders, generative adversarial networks, and natural language models. The class will focus on learning concepts and applying them via several programming assignments.

Also offered for graduate-level credit as CS 540 and may be taken only once for credit. Prerequisite: CS 350.

CS 441 - Artificial Intelligence (4)

Introduction to the basic concepts and techniques of artificial intelligence. Knowledge representation, problem solving, machine learning, natural language understanding, and AI search techniques.

Also offered for graduate-level credit as CS 541 and may be taken only once for credit. . Prerequisite: CS 302.

CS 442 - Advanced Artificial Intelligence: Combinatorial Games (4)

Covers the theory and practice of finding optimal and satisfying solutions to one-player and two-player combinatorial games, including such popular games as Sokoban, Othello, checkers, chess, backgammon, bridge, and CCGs. Simple applications in decision theory and economics may also be discussed. Emphasis on implementation of state-of-the-art solution techniques.

Also offered for graduate-level credit as CS 542 and may be taken only once for credit. . Prerequisite: CS 302 or experience with algorithms and data structures.

CS 442P - Advanced Artificial Intelligence: Combinatorial Games (4)

Covers the theory and practice of finding optimal and satisfying solutions to one-player and two-player combinatorial games, including such popular games as Sokoban, Othello, checkers, chess, backgammon, bridge, and CCGs. Simple applications in decision theory and economics may also be discussed. Emphasis on implementation of state-of-the-art solution techniques.

Prerequisite: CS 302 or experience with algorithms and data structures.

CS 443 - Advanced Artificial Intelligence: Combinatorial Search (4)

Explores methods for the solution of constraint satisfaction and related problems using search techniques, in the context of real-world problems such as resource-bounded scheduling, enterprise planning, classical planning, and one- and two-player games. Emphasis on coding projects, and on reading and reporting on selected literature.

Also offered for graduate-level credit as CS 543 and may be taken only once for credit. . Prerequisite: CS 302 or experience with algorithms and data structures.

CS 445 - Machine Learning (4)

Provides a broad introduction to techniques for building computer systems that learn from experience; conceptual grounding and practical experience with several learning systems; and grounding for advanced study in statistical learning methods, and for work with adaptive technologies used in speech and image processing, robotic planning and control, diagnostic systems, complex system modeling, and iterative optimization. Students gain practical experience implementing and evaluating systems applied to pattern recognition, prediction, and optimization problems.

Also offered for graduate-level credit as CS 545 and may be taken only once for credit. . Prerequisite: Mth 261 OR Mth 343; CS 302.

CS 446 - Advanced Topics in Machine Learning (4)

Covers a number of more advanced topics in machine learning from a more mathematically oriented view. Provides preparation for successfully using machine-learning techniques for various applications. Also provides preparation for graduate-level research in machine learning and adaptive systems.

Also offered for graduate-level credit as CS 546 and may be taken only once for credit. . Prerequisite: CS 445/545.

CS 447 - Computer Graphics (4)

This course will provide an introduction to graphics systems and applications. Basic structure of interactive graphics systems, characteristics of various hardware devices. Control of display devices, implementation of simple packages, device independence, and standard packages. Distributed architectures for graphics, hidden line and hidden surfaces algorithms, representations of curves and surfaces.

Also offered for graduate-level credit as CS 547 and may be taken only once for credit. . Prerequisite: CS 302, Mth 261.

CS 447P - Computer Graphics (4)

This course will provide an introduction to graphics systems and applications. Basic structure of interactive graphics systems, characteristics of various hardware devices. Control of display devices, implementation of simple packages, device independence, and standard packages. Distributed architectures for graphics, hidden line and hidden surfaces algorithms, representations of curves and surfaces.

Prerequisite: CS 302, Mth 261.

CS 451 - Numerical Computation (4)

Introduction to numerical methods. Includes topics from elementary discussion of errors, polynomials, interpolation, quadrature, linear systems of equations, and solution of nonlinear equations.

Also offered for graduate-credit as CS 551 and may be taken only once for credit. . Prerequisite: Mth 261.

CS 452 - Building Software Systems with Components (4)

Designed to familiarize students with the concepts behind and opportunities afforded by modern component architectures, such as Microsoft COM, Java Beans, and CORBA. Students are exposed to component development techniques and methods for developing complex software architectures using components. Students become familiar with component development, scripting and composing components, and the strengths and weaknesses of using components in designing large complex software systems.

Also offered for graduate-level credit as CS 552 and may be taken only once for credit. . Prerequisite: CS 314, CS 333, CS 350; knowledge of C++ or Java programming.

CS 454 - Software Engineering (4)

Current methodologies for the development of large, industrial strength software systems. Topics include requirements, specification, design, implementation, testing, project management and cost estimation, formal methods, and software process improvement.

Also offered for graduate-level credit as CS 554 and may be taken only once for credit. . Prerequisite: CS 302, CS 314, and CS 358.

CS 457 - Functional Programming (4)

Introduction to functional notation, recursion, higher-order functions, reasoning about functions, and models for the evaluation of applicative expressions. Use of functional languages.

Also offered for graduate-level credit as CS 557 and may be taken only once for credit. . Prerequisite: CS 302.

CS 461 - Open Source Software Development Laboratory (4)

Explores Open Source software engineering and its methodologies in a laboratory classroom setting. Focuses on the development and delivery of Open Source software projects by teams of 1-3 students. Students prepare and present material, working using email and the web.

Also offered for graduate-level credit as CS 561 and may be taken only once for credit. . Prerequisite: CS 314.

CS 461P - Open Source Software Development Laboratory (4)

Explores Open Source software engineering and its methodologies in a laboratory classroom setting. Focuses on the development and delivery of Open Source software projects by teams of 1-3 students. Students prepare and present material, working using email and the web.

Prerequisite: CS 314.

CS 462 - Advanced Open Source Software Engineering (4)

Surveys the growing academic literature describing tools, techniques, community management, project management and collaboration strategies used in open source software development. Emphasis is placed upon tool-driven development, upon open development processes and tools, and upon comparison with processes and practices in proprietary software.

Also offered for graduate-level credit as CS 562 and may be taken only once for credit. . Prerequisite: CS 314.

CS 463 - Intro to Web Development (4)

Students will learn the fundamentals of web development, the structure and functionality of the web, and how to create responsive and accessible web applications using HTML, CSS, and JavaScript.

Also offered for graduate-level credit as CS 563 and may be taken only once for credit. Prerequisite: Admission to the program.

CS 464P - Front End Web Technologies (4)

Students will learn the languages, libraries, and frameworks needed to build user interfaces. This class will start with a review of HTML, CSS, and JavaScript, before focusing on React and the React ecosystem. Students will also work with CSS animations, gain experience with data visualization libraries, and learn about design principles and design systems. The final project is a dashboard application, which will leverage many of the topics covered in class and give students the opportunity to create a data-driven React application.

Also offered for graduate-level credit as CS 564 and may be taken only once for credit. Prerequisite: CS 463.

CS 465P - Full Stack Web Development (4)

This class provides an overview of how the web works and covers the spectrum of full stack web development, including using front-end and back-end frameworks to build accessible and responsive applications.

Also offered for graduate-level credit as CS 565 and may be taken only once for credit. . Prerequisite: CS 302, CS 463.

CS 466 - Voice Assistants (4)

Provides an introduction to voice technologies and how to design and build voice-enabled applications, by learning the concepts, techniques, and frameworks needed to build fully functional chatbots and virtual assistants. Students will explore the conversational design process and how to build effective voice user interfaces (VUIs) and conversational user interfaces (CUIs), and create voice-enabled applications and virtual assistants using popular APIs and platforms. Course assumes a working knowledge of JavaScript and Node.js or Python.

Also offered for graduate-level credit as CS 566 and may be taken only once for credit. Prerequisite: Admission to program.

CS 467 - The Wireless Web (4)

Covers the basics of the Wireless Application Protocol (WAP) as used in modern mobile phones and other handheld devices. Provides an overview of the WAP architecture, as well as an in-depth exploration of the WAP Application Layer (WAE), including WML, WMLScript, and the WAP push framework.

Also offered for graduate-level credit as CS 567 and may be taken only once for credit. Prerequisite: CS 465/565.

CS 469 - Software Engineering Capstone I (3)

Emphasizes teamwork on a substantial project that will be developed for a real customer. The course integrates the knowledge and skills from the rest of the CS curriculum. This course creates an obligation for participation for two consecutive quarters. This is the first course in a sequence of two CS 469, CS 470 and must be taken in sequence. Offered as P/NP only.

Prerequisite: CS 302, CS 314, CS 333, CS 350, CS 358, and at least one Programming intensive course.

CS 470 - Software Engineering Capstone II (3)

Emphasizes teamwork on a substantial project that will be developed for a real customer. The course integrates the knowledge and skills from the rest of the CS curriculum. This course creates an obligation for participation for two

consecutive quarters. This is the second course in a sequence of two CS 469, CS 470 and must be taken in sequence. Offered as P/NP only.

Prerequisite: CS 469.

CS 480 - Randomized Algorithms and Probabilistic Analysis (4)

Probabilistic tools used in the design and analysis of modern algorithms and data structures. Topics include: review discrete random, occupancy problems, tail bounds, Markov chains, the probabilistic method, martingales, Monte Carlo methods. The course explores a variety of CS applications.

Also offered for graduate-level credit as CS 580 and may be taken only once for credit. . Prerequisite: CS 350, Stats 451.

CS 485 - Cryptography (4)

The goal of cryptography is the encoding of information via a cryptographic system. Cryptanalysis studies the breaking of cryptosystems. This course focuses on cryptography but with respect to cryptanalysis. An overview of classical systems with an in-depth examination of modern cryptosystems. This includes block algorithms such as DES; public-key cryptosystems, such as RSA; and one-way functions. Additional topics include cryptographic protocols, signature schemes, pseudo-random number generation, Shannon's information theory, and stream ciphers.

Also offered for graduate-level credit as CS 585 and may be taken only once for credit. . Prerequisite: CS 350.

CS 486 - Introduction to Database Management Systems (4)

Introduction to fundamental concepts of database management systems using primarily the relational model. Schema design and refinement. Query languages. Database application development environments. Overview of physical data organization, query optimization and processing, physical design, security, and transactions used in recovery and concurrency control. Expected preparation: CS 251.

Also offered for graduate-level credit as CS 586 and may be taken only once for credit. . Prerequisite: CS 161 and CS 250.

CS 487 - Database Management Systems Implementation (4)

Internal design of a relational database management system. Concurrency control; lock managers; crash recovery; query and operator evaluation; query optimization; storage management; index structures; system catalogs.

Also offered for graduate-level credit as CS 587 and may be taken only once for credit. . Prerequisite: CS 486 and CS 333.

CS 487P - Database Management Systems Implementation (4)

Internal design of a relational database management system. Concurrency control; lock managers; crash recovery; query and operator evaluation; query optimization; storage management; index structures; system catalogs.

Prerequisite: CS 486 and CS 333.

CS 488P - Cloud and Cluster Data Management (4)

Covers advanced data management solutions emerging for cloud and cluster computing environments, focusing on horizontal and vertical scalable approaches. Also covers principles behind data management in these environments, plus specific data management systems that are currently in use or being developed. Topics range from novel data processing paradigms to commercial data management platforms and open-source NoSQL databases. Students will gain broad knowledge about these systems and practical experience with them.

Also offered for graduate-level credit as CS 588 and may be taken only once for credit. Prerequisite: CS 486 or consent of the instructor.

CS 489 - Blockchain Development & Security (4)

Overview of blockchain systems, how they are built, and how they can be exploited. Students will get hands-on experience working with public blockchains as well as build and deploy permissioned blockchains. They will then examine security vulnerabilities in blockchain systems and how they may be automatically exploited.

Also offered for graduate-level credit as CS 589 and may be taken only once for credit. Prerequisite: Upper-division standing.

CS 490 - Introduction to Multimedia Computing and Networking (4)

Introductory course in multimedia computing and networking intended for senior undergraduate or graduate level students. The objective of this course is to introduce many of the fundamental concepts involved with handling multimedia data and applications. The course will cover (i) basic representation and compression of multimedia data types including H.261, JPEG, and MPEG, (ii) techniques to support multimedia quality-of-service in computing and networked systems, and (iii) networked streaming media techniques such as buffering and adaptation.

Also offered for graduate-level credit as CS 590 and may be taken only once for credit. Prerequisite: CS 333 or instructor's permission. .

CS 491 - Introduction to Computer Security (4)

Provides a broad overview of computer security. Introduces foundational principles and shows how they are applied to secure real computing systems. Covers how cryptography, access control, and authentication support confidentiality, integrity, and availability when applied to networks, hosts, information, software, applications, and users.

Also offered for graduate-level credit as CS 591 and may be taken only once for credit. . Prerequisite: CS 205.

CS 492 - Malware Reverse Engineering (4)

Studies the techniques malicious code developers employ to exploit vulnerable computer systems. The course explores the form and function of a range of malware while exploring how the increased mixing of code and data is now exposing us to an array of security vulnerabilities and exploits. Given these threats, the course will then examine modern defenses against malware and how they can be used to protect users.

Also offered for graduate-level credit as CS 592 and may be taken only once for credit. . Prerequisite: Junior-standing and admission into the CS program.

CS 493 - Digital Forensics (4)

Detailed, hands-on approach to the investigation of criminal incidents in which computers or computer technology play a significant or interesting role. Familiarization with the core computer science theory and practical skills necessary to perform rudimentary computer forensic investigations, understanding the role of technology in investigating computer-based crime, and preparation to deal with investigative bodies. Recommended: CS 333 or CS 533. No prior background in criminal justice or law is assumed.

Also offered for graduate-level credit as CS 593 and may be taken only once for credit. .

CS 494 - Internetworking Protocols (4)

Advanced study of the protocols and algorithms used in the Internet (IETF) family of networking protocols. For example, ARP, IP, UDP, TCP, multicasting, routing protocols like RIP and OSPF, and application protocols like DNS, NFS, SNMP, FTP and HTTP. Issues such as addressing, name service, protocol design, and scalability will be explored.

Also offered for graduate-level credit as CS 594 and may be taken only once for credit. . Prerequisite: CS 333.

CS 494P - Internetworking Protocols (4)

Advanced study of the protocols and algorithms used in the Internet (IETF) family of networking protocols. For example, ARP, IP, UDP, TCP, multicasting, routing protocols like RIP and OSPF, and application protocols like DNS, NFS, SNMP, FTP and HTTP. Issues such as addressing, name service, protocol design, and scalability will be explored.

Prerequisite: CS 333.

CS 495 - Web and Cloud Security (4)

Covers web and cloud systems and how they can be subverted. The class will focus on the highest risk vulnerabilities, give students practical experience in how they work, and study how they can be prevented. The class will consist mostly of laboratory exercises focused on developing student skills in performing penetration testing.

Also offered for graduate-level credit as CS 595 and may be taken only once for credit. Prerequisite: CS 205.

CS 496 - Network Security (4)

Focus on network security including a review of various forms of network attacks; a review of basic techniques in applied cryptography, and secure protocols will be covered including network-layer security and various application-layer secure protocols. Also covers network-side security management including both passive measures, as well as active intrusion detection and response. Covers protocols for protection of privacy and anonymity.

Also offered for graduate-level credit as CS 596 and may be taken only once for credit. Prerequisite: CS 205.

CS 497 - Sensor Networks (4)

Foundations of sensor networks, with a focus on activity-based learning through a sequence of hands-on programming exercises with embedded devices with a high-level programming language. Basic building blocks in designing and deploying a sensor network application. Positioning and time synchronization of networked sensor devices, wireless communication characteristics of low-powered radios, energy conservation and harvesting, macro-programming a network of sensor devices and security. Recommended prerequisites: Familiarity with computer systems concepts that could be satisfied by CS 205. Familiarity with programming in C, C++ or Java. Familiarity with basic concepts in probability and linear algebra that could be satisfied by Mth 301 or equivalent.

Also offered for graduate-level credit as CS 597 and may be taken only once for credit. .

CS 498 - Introduction to Wireless Network Protocols (4)

Classification of wireless networking systems; study of multiple access protocols in single hop and multi-hop networks; performance analysis of protocols; overview of emerging radio technologies for high-throughput next generation systems; study of wireless communication protocol standards for cellular systems; case studies of deployed systems.

Also offered for graduate-level credit as CS 598 and may be taken only once for credit. . Prerequisite: CS 250 or ECE 271.

CS 501 - Research (1-9)

(Credit to be arranged.) Consent of instructor.

CS 502 - Independent Study (1-9)

(Credit to be arranged.)

CS 503 - Thesis (1-9)

(Credit to be arranged.) Consent of instructor.

CS 504 - Cooperative Education/Internship (1-9)

(Credit to be arranged.) Consent of instructor.

CS 505 - Reading and Conference (1-12)

(Credit to be arranged.) Consent of instructor.

CS 506 - Special Projects (1-9)

(Credit to be arranged.) Consent of instructor.

CS 507 - Seminar (1-6)

(Credit to be arranged.) Consent of instructor.

CS 509 - Practicum (1-9)

(Credit to be arranged.) Consent of instructor.

CS 510 - Selected Topics (1-6)

(Credit to be arranged.) Consent of instructor.

CS 515 - Parallel Programming (3)

An introduction to parallel programming concepts and techniques. Topics include: parallel programming models and languages, share-memory programming, message-passing programming, performance models and analysis techniques, domain-specific parallel algorithms.

Also offered for undergraduate-level credit as CS 415 and may be taken only once for credit. . Corequisite: CS 515L.

CS 515L - Lab for CS 515P (0)

Lab for CS 515P Parallel Programming.

Corequisite: CS 515.

CS 518 - Cultural Competence in Computing (3)

Students will learn about different identities (e.g., race, ethnicity, gender, class, sexuality, and ability), understand how algorithmic bias in technology affects different identities, and learn how to create accessible and equitable products and more inclusive communities at school, in industry, and beyond.

Also offered for undergraduate level credit as CS 418 and may be taken only once for credit. Prerequisite: Admission to the program.

CS 520 - Object-Oriented Programming & Design (3)

The fundamental concepts of object-oriented programming, including object-oriented modeling and design. The focus of the course will be to help students create programs that model their application domain, that exhibit that model to other programmers who read the code, and that are as a consequence maintainable and robust to change. Issues addressed may include data abstraction and modeling, the use and misuse of inheritance, higher-order data structures and their operations, reusability, refactoring, concurrency control, and usability. Includes programming assignments in an OO language.

Also offered for undergraduate-level credit as CS 420 and may be taken only once for credit. . Prerequisite: CS 553.

CS 530 - Internet, Web, & Cloud Systems (3)

Covers modern networked computing systems and the abstractions they provide. Specifically, students will learn about and apply their knowledge of topics such as Internet protocols, virtual machines and containers, web servers and frameworks, and databases as well as their deployment in modern cloud environments.

Also offered for graduate-level credit as CS 430P and may be taken only once for credit. Prerequisite: Graduate-standing and admission into CS program.

CS 531 - Introduction to Performance Measurement, Modeling and Analysis (3)

A survey of the fundamentals of computer application and system performance. Hands on programming exercises will allow us to apply the techniques to increasingly complex problems. We will use a variety of state of the art tools for measurement, modeling, simulation, and analysis throughout the course.

Also offered for undergraduate-level credit as CS 431 and may be taken only once for credit. Prerequisite: Graduate standing; CS 333 or an equivalent introductory course in Operating Systems.

CS 532 - Operating System Foundations (3)

Foundational concepts of operating system design including processes, threads, scheduling, concurrent programming, synchronization mechanisms, memory management, virtual address translation, file systems and security. A primary goal of the course is to help graduate students acquire the foundational knowledge necessary to succeed in CS 533.

CS 533 - Concepts of Operating Systems (3)

Survey of concepts and techniques used in modern operating systems. Sample concepts covered are concurrency, IPCs, scheduling, resource allocation, memory management, file systems, and security. Techniques for implementing operating systems taught through a programming project.

Prerequisite: CS 333 or CS 532.

CS 535 - Accelerated Computing (3)

Heterogeneous approaches that use special-purpose processors to accelerate the execution of a variety of applications. GPUs, Intel Xeon Phi, APUs, FPGUs. The sustainability implications of these platforms. Lectures, homeworks, labs, and group programming projects using NVIDIA GPUs and Intel Xeon Phi.

Also offered for undergraduate-level credit as CS 435 and may be taken only once for credit.

CS 538 - Computer Architecture (3)

Processors, memory hierarchy, and bus systems. Multi-level caches and cache coherence in MP systems. Arithmetic algorithms. RISC vs. CISC instructions, pipelining, and software pipelining. Superscalar, super pipelined, and VLIW architectures. Connection networks. Performance evaluation, simulation, and analytic models. Performance enhancement through branch prediction and out-of-order execution.

Also offered for undergraduate-level credit as CS 438 and may be taken only once for credit. .

CS 540 - Deep Learning: Computational Structures and Programming (3)

Deep learning is a powerful tool for machine learning systems. This class provides an introduction to this topic and will focus on classic as well as emerging deep learning techniques. Topics studied include multi-layer perceptrons, convolution neural networks, long-short term memory, attention mechanisms, autoencoders, generative adversarial networks, and natural language models. The class will focus on learning concepts and applying them via several programming assignments.

Also offered for undergraduate-level credit as CS 440 and may be taken only once for credit.

CS 541 - Artificial Intelligence (3)

Introduction to the basic concepts and techniques of artificial intelligence. Knowledge representation, problem solving, machine learning, natural language understanding, and AI search techniques.

Also offered for undergraduate-level credit as CS 441 and may be taken only once for credit. .

CS 542 - Advanced Artificial Intelligence: Combinatorial Games (3)

Covers the theory and practice of finding optimal and satisfying solutions to one-player and two-player combinatorial games, including such popular games as Sokoban, Othello, checkers, chess, backgammon, bridge, and CCGs. Simple applications in decision theory and economics may also be discussed. Emphasis on implementation of state-of-the-art solution techniques.

Also offered for undergraduate-level credit as CS 442 and may be taken only once for credit. . Prerequisite: Experience with algorithms and data structures.

CS 543 - Advanced Artificial Intelligence: Combinatorial Search (3)

Explores methods for the solution of constraint satisfaction and related problems using search techniques, in the context of real-world problems such as resource-bounded scheduling, enterprise planning, classical planning, and one- and two-player games. Emphasis on coding projects, and on reading and reporting on selected literature.

Also offered for undergraduate-level as CS 443 and may be taken only once for credit. Prerequisite: Experience with algorithms and data structures.

CS 545 - Machine Learning (3)

Provides a broad introduction to techniques for building computer systems that learn from experience; conceptual grounding and practical experience with several learning systems; and grounding for advanced study in statistical learning methods, and for work with adaptive technologies used in speech and image processing, robotic planning and control, diagnostic systems, complex system modeling, and iterative optimization. Students gain practical experience implementing and evaluating systems applied to pattern recognition, prediction, and optimization problems.

Also offered as undergraduate-level credit as CS 445 and may be taken only once for credit.

CS 546 - Advanced Topics in Machine Learning (3)

Covers a number of more advanced topics in machine learning from a more mathematically oriented view. Provides preparation for successfully using machine-learning techniques for various applications. Also provides preparation for graduate-level research in machine learning and adaptive systems.

Also offered for undergraduate-level credit as CS 446 and may be taken only once for credit. . Prerequisite: CS 445/545.

CS 547 - Computer Graphics (3)

This course will provide an introduction to graphics systems and applications. Basic structure of interactive graphics systems, characteristics of various hardware devices. Control of display devices, implementation of simple packages, device independence, and standard packages. Distributed architectures for graphics, hidden line and hidden surfaces algorithms, representations of curves and surfaces.

Also offered for undergraduate-level credit as CS 447 and may be taken only once for credit. .

CS 549 - Computational Geometry (3)

Perspective and projective geometry. Analytic projective geometry, projective lines and projective planes. Projective transformations of lines and planes. Homogeneous coordinates. Applications to two-dimensional computer graphics. Conic sections in design.

Prerequisite: CS 163 and 451.

CS 550 - Parallel Algorithms (3)

Definition and nature of parallel computation. Parallel computation from the point of view of hardware/architecture, program/scheduling, and algorithms. Why and how parallel computation is different from serial computation. Examples to highlight the differences. Parallel algorithms in general: illustration of the most important features and techniques. Illustration of the limitations. A survey of major results, general form of results, limitations on speed-up.

Prerequisite: CS 350.

CS 551 - Numerical Computation (3)

Introduction to numerical methods. Includes topics from elementary discussion of errors, polynomials, interpolation, quadrature, linear systems of equations, and solution of nonlinear equations.

Also offered for undergraduate-credit as CS 451 and may be taken only once for credit. .

CS 552 - Building Software Systems with Components (3)

Designed to familiarize students with the concepts behind and opportunities afforded by modern component architectures, such as Microsoft COM, Java Beans, and CORBA. Students are exposed to component development techniques and methods for developing complex software architectures using components. Students become familiar with component development, scripting and composing components, and the strengths and weaknesses of using components in designing large complex software systems.

Also offered for undergraduate-level credit as CS 452 and may be taken only once for credit. . Prerequisite: CS 300, CS 333, CS 350; knowledge of C++ or Java programming.

CS 553 - Design Patterns (3)

Software design patterns are reusable solutions to recurring software problems. They capture successful experiences and convey expert insight and knowledge to less experienced developers. Course provides an in-depth view of

patterns using Java as the presentation language. Course is suitable to software architects and developers who are already well-versed in this language. In addition, it offers continuous opportunities for learning the most advanced features of the Java language and understanding some principles behind the design of its fundamental libraries.

Also offered as CS 653 and may be taken only once for credit. . Prerequisite: programming in Java and CS 520.

CS 554 - Software Engineering (3)

Current methodologies for the development of large, industrial strength software systems. Topics include requirements, specification, design, implementation, testing, project management and cost estimation, formal methods, and software process improvement.

Also offered for undergraduate-level credit as CS 454 and may be taken only once for credit. .

CS 555 - Software Specification and Verification (3)

Theoretical and practical aspects of the software development process or software lifecycle. Covers the first part of the cycle: formulating the external requirements, specifying what the software is to do, and the abstract design. Emphasis will be on the formal aspects of specification and verification.

Also offered as CS 655 and may be taken only once for credit.

CS 556 - Software Implementation and Testing (3)

Theoretical and practical aspects of the software development process or software lifecycle. Covers the second part of the cycle: detailed design, implementation in a programming language, testing, and maintenance. Emphasis will be on the technical aspects of software testing.

Also offered as CS 656 and may be taken only once for credit.

CS 557 - Functional Programming (3)

Introduction to functional notation, recursion, higher-order functions, reasoning about functions, and models for the evaluation of applicative expressions. Use of functional languages.

Also offered for undergraduate-level credit as CS 457 and may be taken only once for credit. . Prerequisite: Graduate-standing and admission into the CS program.

CS 558 - Programming Languages (3)

In-depth study of current and historical issues in the design, implementation, and application of programming languages. Topics range from basic to advanced. Areas include syntax, semantics, scoping, typing, abstraction, exceptions, and concurrency. Computational paradigms such as functional, logic, and/or object oriented are analyzed. Several "recent" programming languages used. Expected preparation: CS 320.

Also offered as CS 658 and may be taken only once for credit. .

CS 559 - Software Measurement and Models (3)

Survey, evaluation, and application of software measurement techniques and models. Particular emphasis on product metrics such as Software Science, Cyclomatic Complexity, and Function Points.

CS 560 - Human-Computer Interaction (3)

Introduction to the basic theory of human-computer interaction. Principles of human cognition and interface design, interface evaluation techniques. Several prototyping tools will be presented. A project is required.

CS 561 - Open Source Software Development Laboratory (3)

Explores Open Source software engineering and its methodologies in a laboratory classroom setting. Focuses on the development and delivery of Open Source software projects by teams of 1-3 students. Students prepare and present material, working using email and the web.

Also offered for undergraduate-level credit as CS 461 and may be taken only once for credit. .

CS 562 - Advanced Open Source Software Engineering (3)

Surveys the growing academic literature describing tools, techniques, community management, project management and collaboration strategies used in open source software development. Emphasis is placed upon tool-driven development, upon open development processes and tools, and upon comparison with processes and practices in proprietary software.

Also offered for undergraduate-level credit as CS 462 and may be taken only once for credit. .

CS 563 - Intro to Web Development (3)

Students will learn the fundamentals of web development, the structure and functionality of the web, and how to create responsive and accessible web applications using HTML, CSS, and JavaScript.

Also offered for undergraduate-level credit as CS 463 and may be taken only once for credit.

CS 564 - Front End Web Technologies (3)

Students will learn the languages, libraries, and frameworks needed to build user interfaces. This class will start with a review of HTML, CSS, and JavaScript, before focusing on React and the React ecosystem. Students will also work with CSS animations, gain experience with data visualization libraries, and learn about design principles and design systems. The final project is a dashboard application, which will leverage many of the topics covered in class and give students the opportunity to create a data-driven React application.

Also offered for undergraduate-level credit as CS 464P and may be taken only once for credit. Prerequisite: CS 563.

CS 565 - Full Stack Web Development (3)

This class provides an overview of how the web works and covers the spectrum of full stack web development, including using front-end and back-end frameworks to build accessible and responsive applications.

Also offered for undergraduate-level credit as CS 465P and may be taken only once for credit. . Prerequisite: CS 563.

CS 566 - Voice Assistants (3)

Provides an introduction to voice technologies and how to design and build voice-enabled applications, by learning the concepts, techniques, and frameworks needed to build fully functional chatbots and virtual assistants. Students will explore the conversational design process and how to build effective voice user interfaces (VUIs) and conversational user interfaces (CUIs), and create voice-enabled applications and virtual assistants using popular APIs and platforms. Course assumes a working knowledge of JavaScript and Node.js or Python.

Also offered for undergraduate-level credit as CS 466 and may be taken only once for credit.

CS 567 - The Wireless Web (3)

Covers the basics of the Wireless Application Protocol (WAP) as used in modern mobile phones and other handheld devices. Provides an overview of the WAP architecture, as well as an in-depth exploration of the WAP Application Layer (WAE), including WML, WMLScript, and the WAP push framework.

Also offered for undergraduate-level credit as CS 467 and may be taken only once for credit. . Prerequisite: CS 465/565.

CS 568 - Functional Logic Programming (3)

Introduction to functional logic programming. Foundations and basic principles of this paradigm will be explained in some depth and complemented by encoding practical problems in a functional logic language using a leading compiler/interpreter. Focus on non-determinism and computations with incomplete information. Implementation techniques will be briefly discussed.

Also offered as CS 668 and may be taken only once for credit. . Prerequisite: CS 558 Programming Languages.

CS 569 - Scholarship Skills for Computer Science and Engineering (3)

The purpose of this course is to make participants better scholars in Computer Science. In particular it attempts to help students become better researchers, better writers, better presenters, and better reviewers. It concentrates on reading, writing and composition skills: on the production and consumption of the "media" used by computer

scientists to communicate professionally. At the completion of the course, students should be familiar with the tasks and activities of modern scholars in computer science.

Also offered as CS 669 and may be taken only once for credit. . Prerequisite: admission into a Ph.D. program within MCECS.

CS 570 - Machine Learning Seminar (1)

Graduate seminar on machine learning. Students will read and discuss recent papers in the machine learning literature. This one-credit course will be offered each term, and students may take it multiple times.

Prerequisite: CS 445 or CS 545 or permission of instructor.

CS 572 - Operating System Internals (3)

Internals of a specific operating system including structure of the kernel, block buffering cache, file system structure and system calls, process structure and scheduling, memory management, device driver interface, and inter process communication.

Also offered as CS 672 and may be taken only once for credit. .

CS 575 - Computer Systems Analysis (3)

An advanced course on computer systems. Topics include operating systems, performance evaluation, device analysis, construction and proof of monitors, file systems, objects and processes, reliability, and protection.

Prerequisite: CS 333, Stat 460.

CS 576 - Computer Security Research Seminar (3)

Seminar on emerging topics in computer security.

Also offered as CS 676 and may be taken only once for credit. .

CS 577 - Modern Language Processors (3)

An advanced course on compiler construction for modern programming languages, such as object-oriented or functional languages. Topics include type-checking, executable intermediate representations, interpretation and virtual machines, code generation for modern architectures, memory management and garbage collection, and optimization.

Also offered as CS 677 and may be taken only once for credit. . Prerequisite: CS 421.

CS 578 - Programming Language Semantics (3)

Introduction to the formal mathematical study of program meaning (semantics), using one or more approaches such as operational semantics, denotational semantics, or programming logics. Emphasis on rigorous mathematical development and formal proof techniques. Language features to be studied may include types and type safety, purity and imperative effects, functional and modular abstraction, polymorphism, higher-order functions, and object-oriented features. Expected preparation: CS 558 and/or CS 557.

Also offered as CS 678 and may be taken only once for credit. .

CS 579 - Formal Verification of Hardware/Software Systems (3)

Introduction to the formal verification of functional correctness of hardware and software systems. Topics to be covered include: formal logics for system verification (first-order logic, higher order logic, temporal logic), formal specifications, theorem proving systems, circuit verification, microprocessor verification, and system software verification.

Prerequisite: CS 333.

CS 580 - Randomized Algorithms and Probabilistic Analysis (3)

Probabilistic tools used in the design and analysis of modern algorithms and data structures. Topics include: review discrete random, occupancy problems, tail bounds, Markov chains, the probabilistic method, martingales, Monte Carlo methods. The course explores a variety of CS applications.

Also offered for undergraduate-level credit as CS 480 and may be taken only once for credit. . Prerequisite: CS 350, Stats 451.

CS 581 - Theory of Computation (3)

Computability theory: study of models of computation (Turing, Church, Kleene), recursive function theory, properties of recursive, and recursively innumerable sets.

Prerequisite: CS 311.

CS 582 - Theory of Computation: Advanced Topics (3)

Complexity theory: study of resource bounded computation, the complexity classes (P, NP, PSPACE, and PH), NP-completeness, relativized computation, randomized classes.

Prerequisite: CS 311, 350.

CS 583 - Automata and Formal Languages (3)

An advanced study of the theory of automata, formal languages and computational complexity. Main subjects are finite state concepts, formal grammars, computability, Turing machines, and computational complexity.

Prerequisite: CS 582/682.

CS 584 - Algorithm Design and Analysis (3)

An advanced in-depth study in the design and analysis of algorithms. Topics include models of computation, sorting, data structures, graph algorithms, matrix multiplication, fast Fourier transform, polynomial arithmetic, pattern matching, and NP-complete problems.

Also offered as CS 684 and may be taken only once for credit. .

CS 585 - Cryptography (3)

The goal of cryptography is the encoding of information via a cryptographic system. Cryptanalysis studies the breaking of cryptosystems. This course focuses on cryptography but with respect to cryptanalysis. An overview of classical systems with an in-depth examination of modern cryptosystems. This includes block algorithms such as DES; public-key cryptosystems, such as RSA; and one-way functions. Additional topics include cryptographic protocols, signature schemes, pseudo-random number generation, Shannon's information theory, and stream ciphers.

Also offered for undergraduate-level credit as CS 485 and may be taken only once for credit. .

CS 586 - Introduction to Database Management Systems (3)

Introduction to fundamental concepts of database management systems using primarily the relational model. Schema design and refinement. Query languages. Database application development environments. Overview of physical data organization, query optimization and processing, physical design, security, and transactions used in recovery and concurrency control.

Also offered for undergraduate-level credit as CS 486 and may be taken only once for credit. .

CS 587 - Database Management Systems Implementation (3)

Internal design of a relational database management system. Concurrency control; lock managers; crash recovery; query and operator evaluation; query optimization; storage management; index structures; system catalogs.

Also offered for undergraduate-level credit as CS 487 and may be taken only once for credit. . Prerequisite: CS 486/586 and CS 333.

CS 588 - Cloud and Cluster Data Management (3)

Covers advanced data management solutions emerging for cloud and cluster computing environments, focusing on horizontal and vertical scalable approaches. It covers principles behind data management in these environments, plus specific data management systems that are currently in use or being developed. The topics range from novel data processing paradigms to commercial data management platforms and open-source NoSQL databases. Students will gain broad knowledge about these systems and practical experience with them.

Also offered for undergraduate-level credit as CS 488P and may be taken only once for credit. Prerequisite: CS 586 or consent of the instructor.

CS 589 - Blockchain Development & Security (3)

Overview of blockchain systems, how they are built, and how they can be exploited. Students will get hands-on experience working with public blockchains as well as build and deploy permissioned blockchains. They will then examine security vulnerabilities in blockchain systems and how they may be automatically exploited.

Also offered for undergraduate-level credit as CS 489 and may be taken only once for credit.

CS 590 - Introduction to Multimedia Computing and Networking (3)

Introductory course in multimedia computing and networking intended for senior undergraduate or graduate level students. The objective of this course is to introduce many of the fundamental concepts involved with handling multimedia data and applications. The course will cover (i) basic representation and compression of multimedia data types including H.261, JPEG, and MPEG, (ii) techniques to support multimedia quality-of-service in computing and networked systems, and (iii) networked streaming media techniques such as buffering and adaptation.

Also offered for undergraduate-level credit as CS 490 and may be taken only once for credit.

CS 591 - Introduction to Computer Security (3)

Provides a broad overview of computer security. Introduces foundational principles and shows how they are applied to secure real computing systems. Covers how cryptography, access control, and authentication support confidentiality, integrity, and availability when applied to networks, hosts, information, software, applications, and users.

Also offered for undergraduate-level credit as CS 491 and may be taken only once for credit. .

CS 592 - Malware Reverse Engineering (3)

Studies the techniques malicious code developers employ to exploit vulnerable computer systems. The course explores the form and function of a range of malware while exploring how the increased mixing of code and data is now exposing us to an array of security vulnerabilities and exploits. Given these threats, the course will then examine modern defenses against malware and how they can be used to protect users.

Also offered for undergraduate-level credit as CS 492 and may be taken only once for credit. . Prerequisite: Admission into the CS program.

CS 593 - Digital Forensics (3)

Detailed, hands-on approach to the investigation of criminal incidents in which computers or computer technology play a significant or interesting role. Familiarization with the core computer science theory and practical skills necessary to perform rudimentary computer forensic investigations, understanding the role of technology in investigating computer-based crime, and preparation to deal with investigative bodies. Recommended: CS 333 or CS 533. No prior background in criminal justice or law is assumed.

Also offered for undergraduate-level credit as CS 493 and may be taken only once for credit. .

CS 594 - Internetworking Protocols (3)

Advanced study of the protocols and algorithms used in the Internet (IETF) family of networking protocols. For example, ARP, IP, UDP, TCP, multicasting, routing protocols like RIP and OSPF, and application protocols like DNS, NFS, SNMP, FTP and HTTP. Issues such as addressing, name service, protocol design, and scalability will be explored.

Also offered for undergraduate-level credit as CS 494 and may be taken only once for credit. .

CS 595 - Web and Cloud Security (3)

Covers web and cloud systems and how they can be subverted. The class will focus on the highest risk vulnerabilities, give students practical experience in how they work, and study how they can be prevented. The class will consist mostly of laboratory exercises focused on developing student skills in performing penetration testing.

Also offered for undergraduate-level credit as CS 495 and may be taken only once for credit.

CS 596 - Network Security (3)

The class will focus on network security. In order to understand the network security problem, the course will include a review of various forms of network attacks. We will then review basic techniques in applied cryptography, and then secure protocols will be covered including network-layer security and various application-layer secure protocols. We then turn to network-side security management including both passive measures like firewall defense schemes including packet filers and bastion hosts, as well as active intrusion detection and response. Finally, we will cover protocols for protecting privacy and anonymity.

Also offered for undergraduate-level credit as CS 496 and may be taken only once for credit.

CS 597 - Sensor Networks (3)

Foundations of sensor networks, with a focus on activity-based learning through a sequence of hands-on programming exercises with embedded devices with a high-level programming language. Basic building blocks in designing and deploying a sensor network application. Positioning and time synchronization of networked sensor devices, wireless communication characteristics of low-powered radios, energy conservation and harvesting, macro-programming a network of sensor devices and security. Recommended prerequisites: Familiarity with computer systems concepts that could be satisfied by CS 205. Familiarity with programming in C, C++ or Java. Familiarity with basic concepts in probability and linear algebra that could be satisfied by Mth 301 or equivalent.

Also offered for undergraduate-level credit as CS 497 and may be taken only once for credit. .

CS 598 - Introduction to Wireless Network Protocols (3)

Classification of wireless networking systems; study of multiple access protocols in single hop and multi-hop networks; performance analysis of protocols; overview of emerging radio technologies for high-throughput next generation systems; study of wireless communication protocol standards for cellular systems; case studies of deployed systems.

Also offered for undergraduate-level credit as CS 498 and may be taken only once for credit. . Prerequisite: CS 250 or ECE 271.

CS 601 - Research (1-12)

(Credit to be arranged.) Consent of instructor.

CS 602 - Independent Study (1-12)

(Credit to be arranged.)

CS 603 - Dissertation (1-12)

(Credit to be arranged.) Consent of instructor.

CS 604 - Cooperative Education/Internship (1-8)

(Credit to be arranged.) Consent of instructor.

CS 605 - Reading and Conference (1-8)

(Credit to be arranged.) Consent of instructor.

CS 606 - Special Projects (1-12)

(Credit to be arranged.) Consent of instructor.

CS 607 - Seminar (1-4)

(Credit to be arranged.)

CS 610 - Selected Topics (1-8)

(Credit to be arranged.) Consent of instructor.

CS 653 - Design Patterns (3)

Software design patterns are reusable solutions to recurring software problems. They capture successful experiences and convey expert insight and knowledge to less experienced developers. Course provides an in-depth view of patterns using Java as the presentation language. Course is suitable to software architects and developers who are already well-versed in this language. In addition, it offers continuous opportunities for learning the most advanced features of the Java language and understanding some principles behind the design of its fundamental libraries.

Also offered as CS 553 and may be taken only once for credit. . Prerequisite: programming in Java and CS 520.

CS 655 - Software Specification and Verification (3)

Theoretical and practical aspects of the software development process or software lifecycle. Covers the first part of the cycle: formulating the external requirements, specifying what the software is to do, and the abstract design. Emphasis will be on the formal aspects of specification and verification.

Also offered as CS 555 and may be taken only once for credit.

CS 656 - Software Implementation and Testing (3)

Theoretical and practical aspects of the software development process or software lifecycle. Covers the second part of the cycle: detailed design, implementation in a programming language, testing, and maintenance. Emphasis will be on the technical aspects of software testing.

Also offered as CS 556 and may be taken only once for credit.

CS 658 - Programming Languages (3)

In-depth study of current and historical issues in the design, implementation, and application of programming languages. Topics range from basic to advanced. Areas include syntax, semantics, scoping, typing, abstraction, exceptions, and concurrency. Computational paradigms such as functional, logic, and/or object oriented are analyzed. Several "recent" programming languages used.

Also offered as CS 558 and may be taken only once for credit. .

CS 659 - Software Measurement and Models (3)

Survey, evaluation, and application of software measurement techniques and models. Particular emphasis on product metrics such as Software Science, Cyclomatic Complexity, and Function Points.

CS 668 - Functional Logic Programming (3)

Introduction to functional logic programming. Foundations and basic principles of this paradigm will be explained in some depth and complemented by encoding practical problems in a functional logic language using a leading compiler/ interpreter. Focus on non-determinism and computations with incomplete information. Implementation techniques will be briefly discussed.

Also offered as CS 568 and may be taken only once for credit. . Prerequisite: CS 558 Programming Languages.

CS 669 - Scholarship Skills for Computer Science and Engineering (3)

The purpose of this course is to make participants better scholars in Computer Science. In particular it attempts to help students become better researchers, better writers, better presenters, and better reviewers. It concentrates on reading, writing and composition skills: on the production and consumption of the "media" used by computer

scientists to communicate professionally. At the completion of the course, students should be familiar with the tasks and activities of modern scholars in computer science.

Also offered as CS 569 and may be taken only once for credit. . Prerequisite: admission into a Ph.D. program within MCECS.

CS 672 - Operating System Internals (3)

Internals of a specific operating system including structure of the kernel, block buffering cache, file system structure and system calls, process structure and scheduling, memory management, device driver interface, and inter process communication.

Also offered as CS 572 and may be taken only once for credit. .

CS 676 - Computer Security Research Seminar (3)

Seminar on emerging topics in computer security.

Also offered as CS 576 and may be taken only once for credit. .

CS 677 - Modern Language Processors (3)

An advanced course on compiler construction for modern programming languages, such as object-oriented or functional languages. Topics include type-checking, executable intermediate representations, interpretation and virtual machines, code generation for modern architectures, memory management and garbage collection, and optimization.

Also offered as CS 577 and may be taken only once for credit. . Prerequisite: CS 421.

CS 678 - Programming Language Semantics (3)

Introduction to the formal mathematical study of program meaning (semantics), using one or more approaches such as operational semantics, denotational semantics, or programming logics. Emphasis on rigorous mathematical development and formal proof techniques. Language features to be studied may include types and type safety, purity and imperative effects, functional and modular abstraction, polymorphism, higher-order functions, and object-oriented features. Expected preparation: CS 558 and/or CS 557.

Also offered as CS 578 and may be taken only once for credit. .

CS 684 - Algorithm Design and Analysis (3)

An advanced in-depth study in the design and analysis of algorithms. Topics include models of computation, sorting, data structures, graph algorithms, matrix multiplication, fast Fourier transform, polynomial arithmetic, pattern matching, and NP-complete problems.

Also offered as CS 584 and may be taken only once for credit. .

CS 696 - Network Management and Security (3)

Covers both network management and network security. Network management will include the design of LAN-based networks, including spanning tree protocols, bridge learning protocols, virtual LANs, and Ethernet switches, and the security of switches and routers. Network management protocols will be covered in-depth including switch and router management information bases, as well as associated SNMP protocols, and network monitoring tools. The second half of the class will focus on network security. In order to understand the network security problem, the security section will begin with a review of various forms of network attacks. We then turn to network-side security management including both passive measures like firewall defense schemes including packet filers, and bastion hosts. Newer secure protocols will then be covered including network-layer security and various application-layer secure protocols.

Prerequisite: CS 594.

CS 699 - Special Studies (1-6)

Credit to be arranged.

D - Dance

D 104 - Dance Appreciation (4)

Develop an awareness and appreciation of dance in its artistic, social and cultural contexts through a variety of experiences, viewing and participating in dance. Will cover the basic roles in dance along with concepts and principals such as space, time and effort as well as expression, form, style and period.

D 193 - Dance Laboratory: Modern I, II, III (2)

Beginning modern dance technique, emphasis on body alignment, strength, flexibility and development of basic technical skills. Maximum: 12 credits.

D 195 - Dance Laboratory: Topics I, II, III (2)

Beginning dance technique in topics to be named, for example musical theatre, tap, hip hop, etc. Maximum: 12 credits.

D 196 - Dance Laboratory: Ballet I, II, III (2)

Beginning ballet technique, emphasis on body alignment, development of basic technical skills, and understanding basic ballet vocabulary. Maximum: 12 credits.

D 197 - Dance Laboratory: Jazz I, II, III (2)

Beginning laboratory in jazz dance technique emphasizing body alignment, contraction, and isolation technique of Latin, West Indian, African and American rhythms. Maximum: 12 credits.

D 304 - Dance Appreciation (4)

Develop an awareness and appreciation of dance in its artistic, social and cultural contexts through a variety of experiences, viewing and participating in dance. Covers the basic roles involved in dance along with concepts and principals of dance such as space, time and effort as well as expression, form, style and period.

Prerequisite: Upper-division standing.

D 350 - Dance Improvisation (4)

An exploration of spontaneous movement as individual and group creativity and expression, as a potential performance form and as the beginnings of choreography. "The body thinks." Designed to develop awareness, focus, sensitivity and personal movement vocabularies. Expected preparation: upper division standing.

D 351 - Dance Composition (4)

Exploration of basic elements of dance and choreographic strategies through readings, observations and preparation of solo dance studies. Expected preparation: upper division standing.

D 352 - Dance Choreography (4)

Exploring compositional devices and craft unique to group choreography. Choreographing and producing a dance in a performance setting. Expected preparation: D 350, D 351.