Application Requirements

Application requirements consist of institutional and program requirements for applicable semesters of entry during the current academic year. Additional information and changes to admission requirements for semesters other than the current academic year can be found on The Graduate College's website (http://www.gradcollege.txstate.edu/).

Unless otherwise noted on The Graduate College program page, AI tools can only be used to correct spelling and grammar errors in application materials.

Institutional Requirements

Institutional requirements are the minimum standards for admission to any graduate program at Texas State. These include:

- Completed online application
- Nonrefundable application fee
  - Degree Programs (Doctoral and Master's)
    - $55 fee, or
    - $90 for applications with international credentials
  - Post-Baccalaureate Programs (Certificate, Certification, Non-Degree, and Visiting)
    - $20 fee, or
    - $60 for applications with international credentials
- Official transcripts from each institution where course credit was granted. Final transcripts showing degree completion are required before the student may register for their second term of enrollment.
- GPA requirements (a higher GPA may be listed in the Program Requirements)
  - Doctoral programs require a 3.00 overall GPA or a 3.00 GPA in your last 60 hours (https://www.gradcollege.txst.edu/admissions/policy.html#gpa) of undergraduate course work (plus any completed graduate courses).
  - Master's and Specialist programs require a 2.75 overall GPA or a 2.75 GPA in your last 60 hours (https://www.gradcollege.txst.edu/admissions/policy.html#gpa) of undergraduate course work (plus any completed graduate courses).
  - Post-Baccalaureate programs require a 2.50 overall GPA or a 2.50 GPA in your last 60 hours (https://www.gradcollege.txst.edu/admissions/policy.html#gpa) of undergraduate course work (plus any completed graduate courses).
- Baccalaureate degree from a regionally accredited university. (Non-U.S. degrees must be equivalent to a four-year U.S. Bachelor's degree. In most cases, three-year degrees are not considered. Visit our International FAQs (https://www.gradcollege.txst.edu/international/faqs.html) for more information.)

Approved English Proficiency Exam Scores

Applicants are required to submit an approved English proficiency exam score that meets the minimum requirements below unless they have earned a bachelor's degree or higher from a regionally accredited U.S. institution or the equivalent from a country on our exempt countries list (http://www.gradcollege.txstate.edu/international/language.html#waiver). Some programs may restrict acceptable tests or require higher scores than the institutional scores; this will be noted in the Program Requirements.

- official TOEFL iBT scores required with a 78 overall if taken on or before January 21, 2026
- official TOEFL iBT scores required with a 4 overall if taken after January 21, 2026
- official PTE scores required with a 52 overall
- official IELTS (academic) scores required with a 6.5 overall and minimum individual module scores of 6.0
- official Duolingo scores required with a 110 overall
- official TOEFL Essentials scores required with an 8.5 overall
- official Texas State Intensive English Program score of 90% or higher in the highest-level course (level 5)

The institution does **not** offer admission if the scores above are not met.

Program Requirements

Texas State Computer Science majors who have at least a 3.0 GPA by the end of the spring semester of their junior year and who will have completed all their prescribed courses, including CS 4371 and the Computer Science Project course, by the spring of their junior year will be eligible to apply for the program.

- Minimum overall 3.0 GPA or 3.0 GPA in the last 60 hours of undergraduate course work (plus any completed graduate courses)
- GRE not required
- Resume
- Statement of purpose (maximum two pages) that discusses the student's reasons for pursuing professional master's degree in Computer Science at Texas State University
- A letter of endorsement from the Graduate Advisor
- An interview may be required by program faculty

Students will follow departmental requirements for admission into the BS in Computer Science program. CS majors who have at least a 3.0 GPA by the spring semester of their junior year and who have completed all of their prescribed courses by the spring of their junior year will be eligible to apply for the five-year program.

Applicants to the five-year program will submit materials as other MS in CS applicants by the spring deadline of their junior year. The admission criteria for the five-year program will be the same as for other (post-baccalaureate) MS in CS applicants.

Minimum required: 138 semester credit hours

General Requirements

The program can be completed at the San Marcos Main Campus (M).

1. The general education core curriculum courses are listed in the degree plan below along with the statewide component code number. See the General Education Core Curriculum section of this catalog for the Texas State requirements and options in the core curriculum, including Honors courses.
2. In addition to satisfying the University graduation requirements, students must earn a grade of C or higher in all computer science and mathematics courses used to satisfy the requirements of the undergraduate computer science major.

3. For transfer students, 26-32 semester credit hours in computer science (or their equivalents) may be transferred from a Texas public institution of higher education for the Computer Science Field of Study and be applied to the Bachelor of Science degree with a major in Computer Science at Texas State University. More information about the Field of Study is available in the Academic Policies section of this catalog. If transferring additional computer science courses please contact the Department of Computer Science for assistance. The transferable Texas Common Course Number (TCCN) is listed below the Texas State University course number in the following course list.

Code	Title	Hours
CS 1319	Fundamentals of Computer Science (TCCN: COSC 1336 or 1436 (CS 1319 + 1 hour CS ELNA))	3
CS 1428	Foundations of Computer Science I (TCCN: COSC 1337 or 1437)	4
CS 2308	Foundations of Computer Science II (TCCN: COSC 2336 or 2436 (CS 2308 + 1 hour CS ELNA))	3
MATH 2471	Calculus I (TCCN: MATH 2313 or 2413)	4
MATH 2472	Calculus II (TCCN: MATH 2314 or 2414)	4
PHYS 2325 & PHYS 2125	Mechanics and Mechanics Laboratory (TCCN: PHYS 2325 and 2125)	4
PHYS 2326 & PHYS 2126	Electricity and Magnetism and Electricity and Magnetism Laboratory (TCCN: PHYS 2326 and 2126)	4
Total Hours		**26**

4. Students pursuing this B.S. degree program are required to complete 3 hours of technical or scientific writing. A grade of C or higher is required in these hours to satisfy the graduation requirements of the computer science major. Students may select from ENG 3303 or ENG 3313.

5. No more than 3 credit hours may be applied to the student's major elective from any combination of the following courses:

Code	Title	Hours
CS 3190	Cooperative Education	1
CS 3290	Advanced Cooperative Education	2
CS 4100	Computer Science Internship	1
CS 4298	Undergraduate Research I	2
CS 4299	Undergraduate Research II	2
CS 4395	Independent Study in Computer Science	3
HON 4390B	Honors Capstone	3
RES 4399	Mentored Research and Creative Expression	3

6. The required courses for this major include 14 of the 20 hours of coursework required for a Mathematics or Applied Mathematics minor. Therefore, this degree plan includes two additional courses needed to complete one of these minors.

7. Nine hours of writing intensive (WI) courses are required for graduation.

8. Students must complete a minimum of 36 advanced hours (3000 or 4000 level courses).

9.If two years of the same language are taken in high school, then no additional language hours will be required for the degree. In the absence of language taken in high school, then two semesters of the same modern language (1410 and 1420) must be taken at the college level, and the requirement will be added to the student's degree audit.

10. Students must complete 46 hours of Computer Science courses, including one CS project course from: CS 4318, CS 4326, CS 4380, or CS 4398.

11. The number of free electives a student will complete varies, depending on the number of hours needed to satisfy the 120 and/or the 36 advanced or 9 hours writing intensive requirements. Students should consult with the academic advisor before enrolling in any free elective courses to ensure that electives are needed.

12. Students entering Texas State with fewer than 16 credit hours completed after high school graduation will be required to take US 1100. All others will be exempt from taking this course. Students may be required to earn an additional elective to reach the 120 minimum total credit hour requirement for the awarding of a degree.

Course Requirements

				Year 1
First Semester	**Hours**		**Second Semester**	**Hours**
CS 1428	4		CS 2308	3
MATH 2471 (Mathematics Component Code 020)	4		MATH 2472 (Component Area Option Code 092)	4
US 1100	1		PHIL 1305 or 1320 (Language, Philosophy, and Culture Component Code 040)	3
ENG 1310, 1320, or 1321 (Communication Component Code 010)	3		MATH 2358	3
COMM 1310 (Component Area Option Code 090/091 [TCCN SPCH 1311])	3		POSI 2310 (Government/ Political Science Component Code 070 [TCCN GOVT 2306])	3
	15			**16**

				Year 2
First Semester	**Hours**		**Second Semester**	**Hours**
CS 2325	3		CS 2315	3
CS 3358	3		CS 3354	3
MATH 3398	3		Life and Physical Sciences Component Code 030	4
Life and Physical Sciences Component Code 030	4		Social and Behavioral Sciences Component Code 080	3
			Elective	3
	13			**16**

				Year 3
First Semester	**Hours**		**Second Semester**	**Hours**
CS 3360	3		CS 4371	3
CS 3398	3		Computer Science Project Course	3
Life and Physical Sciences	4		CS Advanced Elective [1]	3

ENG 3303 (Communication Component Code 010)	3	MATH 3305	3
American History Component Code 060	3	American History Component Code 060	3
	16		**15**

Year 4

First Semester Hours		Second Semester Hours	
CS 5329	3	CS 5306, 5310, or 5332	3
CS Elective [2]	3	CS Elective [2]	3
Math/Applied Math minor	3	Math/Applied Math Minor	3
ART 2313, DAN 2313, MU 2313, or TH 2313 (Creative Arts Component Code 050 [HUMA 1315])	3	Elective	2
Elective	3	POSI 2320 (Government/ Political Science Component Code 070 [TCCN GOVT 2305]))	3
	15		**14**

Fifth Year

First Semester Hours		Second Semester Hours	
CS 5346 or 5391	3	CS 5318, 5338, or 5351	3
Masters CS Elective[2]	6	Masters CS Elective[2]	6
	9		**9**

Total Hours: 138

[1] Any of the master's level courses can count as a 4000-level CS advanced elective (as long as a corresponding 4000-level course has not already been taken). Certain 4000-level courses should be avoided here if the student wants or needs to take a corresponding 5000-level course:

- CS 4326/CS 5326
- CS 4347/CS 5313
- CS 4332/CS 5332 (not stacked, but cannot take both)
- CS 4355/CS 5329
- CS 4379G(Bora)/CS 5311
- CS 4379N/CS 5369N
- CS 4380/CS 5351
- CS 4371/CS 5378
- CS 4372/CS 5375
- CS 4388/CS 5388
- CS 4398/CS 5394
- CS4379K/CS5324

In some cases, like CS 4355/CS 5329, if the student decides to enroll in the 3+2 program, the student should take CS 5329, which is a core course for the master's degree, instead of CS 4355 which is an elective course for the bachelor's degree.

[2] See list below for Masters CS Elective options

Code	Title	Hours
Masters CS Electives:		
CS 5306	Advanced Operating Systems	3
CS 5310	Network and Communication Systems	3
CS 5316	Data Mining	3
CS 5318	Principles of Programming Languages	3
CS 5326		3
CS 5331	Crafting Compilers	3
CS 5332	Database Theory and Design	3
CS 5334	Advanced Internet Information Processing	3
CS 5338	Formal Languages	3
CS 5341	Advanced Network Programming	3
CS 5343	Wireless Communications and Networks	3
CS 5346	Advanced Artificial Intelligence	3
CS 5351	Parallel Processing	3
CS 5352	Distributed Computing	3
CS 5369J		3
CS 5369L		3
CS 5369Q	Recommender Systems	3
CS 5369Y	Green Computing	3
CS 5375	Multimedia Computing	3
CS 5378	Advanced Computer Security	3
CS 5388	Advanced Computer Graphics	3
CS 5389		3
CS 5391	Survey of Software Engineering	3
CS 5392	Formal Methods in Software Engineering	3
CS 5393	Software Quality	3
CS 5394	Advanced Software Engineering Project	3
CS 5395	Independent Study in Advanced Computer Science	3

3+2 program students can also take most CS 7000 level courses just like regular CS master's degree students in their 5th year.

13. Comprehensive exam for MS degree

- The comprehensive exams of the computer science master programs consist of multiple components. Specifically, all Computer Science graduate students must complete/pass:

1. Programming exam: Not required.
2. Communication exam: Not required.
3. GRE: optional.
4. Core course exam.
5. Attendance requirement of computer science seminars

Details of these exams can be found from the Graduate Catalog.

Courses Offered

Computer Science (CS)

CS 1308. Computer Literacy and the Internet.
This course provides an introduction to fundamental computing concepts for students with little or no prior background in computer science. Topics include introductory concepts in computing systems and hardware, data representation, networking and internet fundamentals, cybersecurity, artificial intelligence and data science. Students develop practical skills using common productivity software, including word processing, spreadsheets, and presentations, and examine basic algorithmic problem solving techniques. The course emphasizes understanding how computing technologies function and how they are used across academic, professional, and everyday contexts. Designed for non majors, the course offers foundational knowledge for informed engagement with digital systems.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter
TCCN: COSC 1301

CS 1309. AI for Everyone.
This course provides an introduction to the principles and techniques used in artificial intelligence (AI). It covers both foundational and advanced principles underlying AI and explains the differences between AI, machine learning, and deep learning. It explores critical topics such as neural networks, natural language processing, reinforcement learning, and robotics. Students will gain experience with generative AI tools and prompt engineering. Ethical questions and potential sources of bias in AI systems are studied as topics of analysis to support informed evaluation. This course will not satisfy CS major or minor requirements.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Credit/No Credit

CS 1319. Fundamentals of Computer Science.
This course introduces fundamental concepts of computer science through a breadth#first examination of major computing layers, including information, hardware, programming, operating systems, applications, and communication systems. The course examines historical developments, societal influences, and technical constraints that have shaped modern computing. Emphasis is placed on conceptual understanding supported by practical examples aligned with widely recognized computing curricula. Students complete basic computing exercises to develop familiarity with how computer systems function and interact. This course is designed for non#majors and does not apply toward computer science major or minor requirements.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter
TCCN: COSC 1336

CS 1342. Programming for Scientists and Engineers.
This course examines computational problem solving and program development in the context of engineering and the physical sciences. Students analyze scientific and engineering problems and construct algorithmic solutions using structured programming techniques in a high-level language. Topics include fundamental concepts of computer organization, data representation, expressions, control structures, functions, arrays and collections, and modular program design. Emphasis is placed on systematic approaches to testing and debugging, effective program organization, and the relationship between algorithm design, data handling, and the correctness and reliability of computational solutions. The course also considers computational tools and practices commonly used in scientific computing and data analysis.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 1428. Foundations of Computer Science I.
This course examines foundational concepts in computer science through problem solving and algorithm development. It focuses on how computational problems are analyzed, designed, and implemented using structured programming techniques in C++. Students analyze problems and construct solutions using primitive and complex data types, expressions, control structures, functions, arrays, and related constructs. Clear program organization, consistent and readable coding style, and systematic approaches to testing and debugging are central themes. Particular attention is given to the relationship between algorithm design, program structure, and the correctness and reliability of software. Prerequisite: [MATH 1315 or MATH 1317 or MATH 1319 or MATH 1329 or MATH 2417 or MATH 2471 with a grade of "C" or better] or [ACT Mathematics score of 24 or better] or [New ACT Mathematics score of 25 or better] or [SAT Mathematics score of 520 or better] or [SAT Math Section score of 550 or better] or [Accuplacer College Mathematics score of 86 or better] or [Compass College Algebra score of 46 or better] or [Next-Generation Advanced Algebra and Functions Test of 263 or better].
4 Credit Hours. 3 Lecture Contact Hours. 2 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering|Lab Required
Grade Mode: Standard Letter
TCCN: COSC 1437

CS 2308. Foundations of Computer Science II.
This course examines intermediate concepts in computer science through the design, implementation, and analysis of larger programs in C ++. Topics include abstract data types (ADTs) such as lists, stacks, and queues, and their implementation using arrays and linked lists. Students learn how pointers, dynamic memory allocation, and linked structures support flexible data representation and manipulation. The course emphasizes object-oriented programming, including classes, objects, encapsulation, and the separation of interface and implementation. Students also study fundamental searching and sorting algorithms and analyze their performance using time complexity. Program development, compilation, and testing are performed in a Linux command-line environment. Prerequisite: CS 1428 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter
TCCN: COSC 2336

CS 2315. Computer Ethics.
This course introduces students to ethical, legal, and professional issues that arise in a world shaped by pervasive computing and rapidly evolving AI. Grounded in classical and contemporary ethical frameworks, the course develops students' ability to analyze real-world cases involving privacy, security, intellectual property, automation, accessibility, globalization, and the social impact of computing on race, class, and gender. Through intensive writing, collaborative presentations, and a substantial term paper, students practice clear communication for diverse audiences, apply professional codes of ethics, and critically evaluate sources. Emphasis is placed on civic and professional responsibility, helping future computing professionals anticipate downstream consequences of technology and justify responsible courses of action. Prerequisites: CS 1428 and [COMM 1310 or COMM 2338] and [ENG 1310 or ENG 1320 or ENG 1321 or ENG 3303] and [PHIL 1305 or PHIL 1320] with grades of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering|Writing Intensive
Grade Mode: Standard Letter

CS 2318. Assembly Language.
This course provides a foundational exploration of digital data representation, computer organization, and assembly language programming. Students will examine how machine-level instructions are structured and executed, how processors manage and manipulate data, and how high-level programming constructs are translated into low-level executable code. Through a combination of written assignments and hands-on programming exercises, students gain practical experience working directly with hardware abstractions. By the end of the course, students will be equipped to explain core architectural components, analyze and describe instruction set formats, and design and implement fundamental algorithms using assembly language. Prerequisite: CS 2308 and MATH 2358 with grades of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter
TCCN: COSC 2325

CS 2325. Computer Organization.
This course provides an introduction to computer systems from a programmer's perspective. Students become familiar with computer organization by learning assembly language and studying computer architecture. Topics include data representation, assembly language fundamentals and programming, the compilation process, processor components, logic circuits, and memory hierarchy design. The fundamental understanding of computer organization provided by this course will enable students to write more efficient programs and prepare them for advanced computer architecture courses. Prerequisite: CS 2308 and MATH 2358 both with grades of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 3190. Cooperative Education.
This course provides students participating in approved computer science internships the opportunity to integrate academic knowledge with professional practice in an occupational setting. Students apply concepts from prior coursework to technical tasks encountered in the workplace and analyze how academic preparation supports professional responsibilities. The course emphasizes reflection on problem solving processes, technical communication, and skill development through supervised work experience. Students complete structured reports or presentations that examine the relationship between classroom learning and workplace activities. A maximum of three credit hours of cooperative education may be applied toward the computer science major as elective credit. Prerequisite: Minimum 2.25 Overall GPA and instructor approval.
1 Credit Hour. 0 Lecture Contact Hours. 40 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Dif Tui- Science & Engineering
Grade Mode: Credit/No Credit

CS 3290. Advanced Cooperative Education.
This course provides students participating in approved computer science internships the opportunity to integrate academic knowledge with professional practice in an occupational setting. Students apply concepts from prior coursework to technical tasks encountered in the workplace and analyze how academic preparation supports professional responsibilities. The course emphasizes reflection on problem solving processes, technical communication, and skill development through supervised work experience. Students complete structured reports or presentations that examine the relationship between classroom learning and workplace activities. A maximum of three credit hours of cooperative education may be applied toward the computer science major as elective credit. Prerequisite: A minimum 2.25 Overall GPA and instructor approval.
2 Credit Hours. 0 Lecture Contact Hours. 40 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Dif Tui- Science & Engineering
Grade Mode: Credit/No Credit

CS 3320. Internet Software Development.
This course presents internet web development using a full-stack methodology covering both static and dynamic web pages with client-side and server-side components, and includes the networking concepts necessary for communication between layers. It begins with client-side static web pages and progresses through styling, consuming web APIs from the client side, implementing web APIs on the server side, and persisting data in an external data store. It culminates with a final capstone project to demonstrate mastery and interworking of all layers. Prerequisite: CS 2308 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 3339. Computer Architecture.
This course equips students with a foundational understanding of how modern computers are designed and built from the ground up. Students will explore how hardware components, from arithmetic logic units to memory systems, work together to execute instructions efficiently. Through hands-on engagement with topics like instruction set architectures, pipelining, multiprocessing, and virtual memory, students develop the analytical skills needed to evaluate performance trade-offs and write lower-level programs. By the end of the course, students will understand not just what a computer does, but precisely how and why it does it. Prerequisite: CS 2308 and [CS 2318 or CS 2325 or EE 3320] with grades of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 3354. Object-Oriented Design and Programming.
This course covers object-oriented design principles and programming for students with prior programming experience. Students will explore inheritance and polymorphism, object-oriented design processes, UML diagrams, design patterns, exception handling, multithreading, testing, performance profiling, and the use of AI in extending and refactoring software systems. Students will design and implement Java programs using design patterns and apply event-driven programming principles by developing graphical user interfaces with a Java graphics library. Thread concurrency is also explored through Java programs that manage shared data via read and write operations. Prerequisite: CS 3358 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 3358. Data Structures and Algorithms.
This course examines classic data structures and algorithms, including abstract data types such as sets, lists, stacks and queues, as well as trees, heaps, hash tables, and graphs, with emphasis on algorithm design and quantitative analysis of time and space complexity. Students explore generic programming, sorting and searching algorithms and recursion, gaining exposure to a wider range of problem-solving strategies. By the end of the course, students are expected to demonstrate proficiency in selecting appropriate data structures, applying algorithms to problem solving, and analyzing algorithmic performance using Big-O complexity analysis and other established complexity measures. Prerequisite: CS 2308 and MATH 2358 with grades of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 3360. Computing Systems Fundamentals.
This course covers fundamental concepts underlying the design and implementation of computing systems. Topics include performance evaluation, queueing systems and scheduling, communication and networks, parallel programming and synchronization. Students will explore key concepts, examine critical issues, and study classic solutions across these topics. After completing the course, students should be able to describe a variety of problems that reoccur in computing systems, analyze the causes and impacts of those problems, and apply appropriate tools and algorithms to solve them. Prerequisite: [CS 2318 or CS 2325 or EE 3320] and CS 3358 with grades of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 3378. Theory of Automata.
This course introduces the fundamental concepts and principles of automata theory, formal languages and computability. Course topics include finite state machines, pushdown automata, Turing machines, regular grammar, regular expression, context free grammar, Church-Turing thesis, Halting problem, and reduction proofs. The course is theoretical and mathematical in nature, however, interactive simulation tools such as JFLAP will be used to help students visualize, design, and simulate abstract machines. Mini programming projects will be assigned to help students connect theory with practice. Prerequisite: CS 3358 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 3398. Software Engineering.
This course introduces the study of software design, architecture, implementation, testing and validation techniques through team projects. Through team projects, students will learn to self-organize, plan deliverables, assign work, report status and present demonstrations that allow all stakeholders to assess the quality, features, architecture and schedule of the projects. Students will complete projects that approximate industry experience using skills such as Agile sprints, stand-ups, end-of-sprint presentations and reviews, use of AI, and version control, as well as architecture design and documentation. Prerequisite: CS 3354 and CS 3358 and [CS 2315 or EE 2300] with grades of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering|Writing Intensive
Grade Mode: Standard Letter

CS 4100. Computer Science Internship.
This course provides CS undergraduate students on-the-job training supervised by computer professionals in industry internship programs. During the internship, students apply what they have learned in the classroom toward the internship work. They are required to submit a mid-term and a final report of their internship work, describing what they have learned as a result of the internship and what curriculum additions or improvements they would suggest as a result of the internship experience. Students need the approval of the department in advance to take this internship course. Prerequisite: Minimum 2.25 Overall GPA and instructor approval.
1 Credit Hour. 0 Lecture Contact Hours. 20 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Credit/No Credit

CS 4298. Undergraduate Research I.
This course allows interested undergraduate students to pursue an individual research project under the supervision of a computer science faculty member. The course content depends on the chosen research project but typically includes an in-depth study of the selected topic, a review of related work, as well as the implementation of an idea. It is the first course of a two-semester course sequence and cannot be given degree credit until the satisfactory completion of CS 4299. Prerequisites: Minimum 3.00 Major GPA and instructor approval.
2 Credit Hours. 1 Lecture Contact Hour. 2 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Dif Tui- Science & Engineering
Grade Mode: Credit/No Credit

CS 4299. Undergraduate Research II.
This course allows interested undergraduate students to pursue an individual research project under the supervision of a computer science faculty member. The course content depends on the chosen research project but typically includes the evaluation of the idea implemented in CS 4298 as well as the presentation and a write-up of the project and its results. It is the second course of a two-semester course sequence and can only be taken after the satisfactory completion of CS 4298. Prerequisites: Instructor Approval.
2 Credit Hours. 1 Lecture Contact Hour. 2 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4310. Computer Networks.
This course introduces the fundamental concepts underlying the design and operation of computer networks, with emphasis on the Internet's TCP/IP architecture. Students explore layered network models, packet switching, IP addressing, routing, transport protocols, congestion control, wireless networking, and foundational security concepts. The course emphasizes protocol behavior, performance analysis, and hands#on investigation using tools such as Wireshark. By the end of the course, students will have a strong foundation for designing, evaluating, and troubleshooting networked systems, preparing them for advanced study in networking and cybersecurity. Prerequisite: CS 3360 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4315. Introduction to Data Mining and Information Retrieval.
This course covers fundamental principles and techniques in two closely related fields of data mining and information retrieval. Data mining topics include classification, cluster analysis, and frequent pattern mining. Information retrieval topics include Boolean retrieval, vector space model, and web search. Through programming assignments and projects, students will enhance their understanding of theoretical concepts and gain hands-on experience on practical techniques. They will also explore popular tool packages such as Weka, Orange, Apache Lucene, and Elasticsearch. Prerequisite: CS 3358 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4318. Compiler Construction.
This course examines the design and implementation of compilers, focusing on the translation of high-level programs into executable code. Topics include lexical analysis, parsing, semantic analysis, intermediate representations, optimization, and code generation, emphasizing correctness and performance across architectures. Students construct components of a compiler and integrate them into a working system, applying formal models, data structures, and algorithms to solve translation problems. By the end of the course, students produce executable code from source programs, analyze design trade-offs in compilation strategies, and evaluate the impact of optimizations on program behavior and performance. Prerequisite: CS 3358 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4328. Operating Systems.
This course examines the principles and design of modern operating systems. Topics include process and thread management, CPU scheduling, synchronization, interprocess communication, deadlocks, memory management, virtual memory, file systems, I/O, virtualization, and cloud computing. Students explore system calls, kernel structure, and security mechanisms, gaining insight into how operating systems manage and allocate hardware resources efficiently. Emphasis is placed on the interaction between hardware and software and on performance trade-offs inherent in real-world system design. Programming assignments provide hands-on experience with system-level concepts using a Unix/Linux environment, reinforcing theoretical knowledge through practical implementation. Prerequisite: [CS 2325 or CS 3339] and CS 3360 with grades of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4332. Introduction to Database Systems.
This course introduces students to the fundamental concepts and practices of database management systems. Topics include data modeling methodologies, relational database design, normalization techniques, and Entity-Relationship (ER) diagramming. Students will develop proficiency in Structured Query Language (SQL) and gain hands-on experience with industry-standard Database Management System (DBMS) platforms. Additional topics include file structures, query optimization, and database performance tuning. By the end of the course, students will be equipped to design, implement, and maintain relational databases for real-world applications. Prerequisite: CS 3358 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4337. Introduction to Computer Vision.
This course provides a comprehensive foundation in image processing and computer vision, spanning both classical analytical techniques and modern data-driven deep learning paradigms. Students explore core topics ranging from edge detection, morphological operations, and dimensionality reduction to convolutional neural networks, object detection, and generative vision. The curriculum employs a rigorous, hands-on programmatic methodology that requires students to build and train algorithms capable of processing complex visual data. Ultimately, students will develop the capacity to design, implement, and critically evaluate computer vision systems for real-world applications while navigating the ethical implications of artificial intelligence. Prerequisite: CS 3358 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4346. Introduction to Artificial Intelligence.
This course introduces the fundamental principles of Artificial Intelligence (AI) and Generative AI, highlighting the relationship between traditional AI techniques and modern generative methods. Students study problem-solving, system design, and ethical considerations in computing while building a foundation in core AI methodologies, including search strategies, knowledge representation, basic machine learning, and generative models. Students also explore applications such as question-answering systems, game playing, and planning, and learn how to design and implement intelligent systems. By the end of the course, students will be able to apply AI and Generative AI techniques to develop intelligent systems for real-world applications. Prerequisite: CS 3358 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4347. Introduction to Machine Learning.
This course introduces students to machine learning concepts and techniques for solving real-world prediction and pattern recognition problems. Topics include supervised learning methods such as regression, classification, decision trees, random forests, support vector machines, and neural networks, as well as unsupervised approaches including clustering and dimensionality reduction. Students will apply these methods using Python and common machine learning libraries to work with realistic datasets, emphasizing data preprocessing, model selection, and performance evaluation. At the end of the course, students will be able to explain core machine learning tasks, apply appropriate algorithms to new problems, and evaluate the effectiveness and limitations of different machine learning models in practical contexts. Prerequisite: CS 3358 and MATH 3305 with grades of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4350. Unix Systems Programming.
This course examines systems programming on Unix/Linux platforms with an emphasis on low-level software development using C and modern system interfaces. Topics include Unix file system and environment, memory management, processes and signals, threads and synchronization, interprocess communication, device interaction, network programming, and secure systems programming, along with contemporary topics including multithreading scalability, containerization, and performance profiling. By the end of the course, students will be able to design and implement efficient and secure system software, analyze system behavior and performance, and apply Unix/Linux programming concepts to real-world computing environments. Prerequisite: CS 3358 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4355. Algorithms and Analysis.
This course examines foundational and advanced algorithms in computer science with an emphasis on algorithm design, analysis, and problem formulation. Topics include techniques such as divide-and-conquer, greedy methods, dynamic programming, and backtracking, along with advanced tree, graph, sorting, searching, and string processing algorithms. Students develop skills in evaluating algorithm correctness, efficiency, and space usage using asymptotic analysis. Through analytical reasoning and implementation-based exercises, students learn to select and apply appropriate algorithmic strategies to solve non-trivial computational problems. Prerequisite: CS 3358 with grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4371. Computer System Security.
This course examines the theoretical and applied principles of computer system security, including policy frameworks, cryptographic mechanisms, and offensive and defensive methodologies. Topics include security models, industry-standard practices, system vulnerabilities, and strategies for their mitigation. Students analyze secure network architectures, evaluate protective protocols, and investigate attack techniques that exploit system weaknesses. Through hands-on application and critical analysis, students develop the skills necessary to assess risk, respond to threats, and maintain the integrity of computing environments. Prerequisite: CS 3358 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4372. Introduction to Digital Multimedia.
This course introduces the fundamental concepts and techniques of digital multimedia, focusing on how text, audio, images, and video are represented, processed, and compressed. Students examine multimedia systems and applications, multimedia transmission, and the role of standards in enabling interoperability and efficient media exchange. Topics include graphics and image representation, image and video processing, visual perception and color models, lossless and lossy compression (including frequency-domain methods), multimedia compression standards, and emerging technologies such as virtual and augmented reality. Prerequisite: CS 3358 with a grade of "C" of better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4373. Introduction to Network Science.
This course explores the fundamental concepts and algorithms of network science, including essential topics such as graph representations, centrality measures, link analysis, and community detection. Through a combination of theoretical study and computational projects, students will apply discrete mathematics and programming skills to model, visualize, and analyze real-world network datasets. By the end of the course, students should possess the analytical toolkit necessary to interpret large-scale network-structured data in the fields of data science and social computing. Prerequisite: CS 3358 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4379F. Distributed Data Processing.
This course introduces the principles and practice of distributed data processing for large-scale data-intensive applications. It covers core concepts such as distributed database architectures, distributed file systems, parallel processing models (e.g., MapReduce), and data management tools within the Hadoop ecosystem, including Hive, Pig, and HBase, as well as modern frameworks such as Apache Spark. The course emphasizes both foundational concepts and evolving technologies that support scalable data analytics across clusters. Instruction combines lectures with hands-on programming assignments and projects using Hadoop and related tools to reinforce practical skills. By the end of the course, students will be able to design, implement, and evaluate distributed data processing pipelines and reason about performance, scalability, and system trade-offs. Prerequisite: CS3354 and CS 3358 with grades of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Topics
Grade Mode: Standard Letter

CS 4379G. Data Analysis and Visualization.
This course introduces fundamental and practical techniques for data analysis and visualization in Python. Students learn data wrangling, cleaning, exploratory data analysis (EDA), unsupervised learning, and the construction of interactive dashboards. The methodology emphasizes hands-on programming, use of modern data science libraries, and open-source dashboard frameworks. Through assignments and a milestone-based project, students develop reproducible data analysis pipelines and communicate results visually. The course focuses on analytical methods, data processing techniques, and visual representation of data. Prerequisite: CS 2308 with grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Topics
Grade Mode: Standard Letter

CS 4379H. Cryptography.
This course introduces the fundamental paradigms and principles of modern cryptography, with emphasis on formal definitions, mathematical foundations, and algorithmic constructions. Topics include symmetric and public#key cryptographic primitives, hash functions, digital signatures, and message authentication codes for secure systems. Students examine commonly defined security properties, the formal models used to specify them, and standard techniques for evaluating whether cryptographic constructions satisfy these properties. The course emphasizes rigorous reasoning using algebraic and number#theoretic tools and introduces basic cryptanalysis methods for assessing algorithmic strength. By the end of the course, students will be prepared to read and interpret a substantial portion of contemporary cryptography research literature and widely used technical standards. Prerequisite: CS 3358 with grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Topics
Grade Mode: Standard Letter

CS 4379K. Introduction to Autonomous Robotics.
This course introduces programming, algorithms, and artificial intelligence techniques used in autonomous robotic systems. Topics include motion control, state estimation using Kalman filters, localization with particle filters, computer vision, object detection, task and motion planning, deep reinforcement learning, and multirobot coordination. The course also examines applications such as autonomous vehicles and analyzes the technical and societal considerations associated with intelligent robotic systems. Emphasis is placed on understanding algorithmic foundations, system integration, and practical implementation in real#world robotic platforms. Prerequisite: CS 3358 with grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Topics
Grade Mode: Standard Letter

CS 4379Q. Introduction to Recommender Systems.
This course introduces students to the concepts and techniques of recommender systems for delivering personalized content, products, and information in modern, AI-driven applications. Major recommendation approaches are covered, including content-based filtering, neighborhood- and model-based collaborative filtering, matrix factorization, sequence-aware and context-aware methods, and hybrid architectures used in contemporary web, mobile, and streaming platforms. Students implement complete recommendation pipelines in Python using real interaction and rating data, with emphasis on data preprocessing, candidate generation, ranking, offline evaluation, and the design of A/B experiments and online evaluation strategies. By the end of the course, students are able to apply state-of-the-art recommendation algorithms, analyze and explain their performance using appropriate metrics, and critically evaluate issues such as privacy, fairness, transparency, and emerging regulatory and platform policies that shape recommender systems in today's digital ecosystem. Prerequisite: CS 4347 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Dif Tui- Science & Engineering|Topics
Grade Mode: Standard Letter

CS 4379Y. Introduction to Green Computing.
This course introduces undergraduate students to the principles, methods, and technologies used to analyze and improve the energy efficiency of computing systems. Topics include energy#efficient hardware design, software techniques for reducing power consumption, energy and carbon#aware scheduling, resource management, and power measurement and profiling. Students examine data center efficiency metrics, trade#offs between performance and energy use, and the computational characteristics of machine learning workloads. Emphasis is placed on quantitative evaluation, system#level analysis, and evidence#based comparison of design alternatives. The course treats sustainability#related considerations as technical and economic constraints within computing systems, enabling students to assess design decisions using established metrics and tools rather than prescriptive policy positions. Prerequisite: [CS 2325 or CS 3339] and CS 3358 with grades of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Dif Tui- Science & Engineering|Topics
Grade Mode: Standard Letter

CS 4380. Parallel Programming.
This course explores practical aspects of parallel programming, including multi-core processors and shared-memory programming, GPUs and accelerator programming, and distributed-memory machines and message-passing programming. Topics include MPI, POSIX threads, OpenMP, CUDA, HIP, loop parallelization, parallel algorithms, amorphous data parallelism, atomic operations, prefix sums, performance measurement, parallelism bugs, and case studies of parallel programs. Students gain applied knowledge and skills by developing, testing, and evaluating the performance of parallel software on various shared- and distributed-memory systems. Prerequisite: [CS 2325 or CS 3339] and CS 3360 with grades of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4381. Practical Game Development.
This course explores the design and implementation of computer games through an in-depth study of game engine architecture, physics simulation, software design patterns, and artificial intelligence techniques. Students examine topics such as hierarchical state machines, genetic algorithms, state-space search algorithms, A* pathfinding, goal-based planning, and fuzzy logic, focusing on how these methods create responsive and believable game behavior. The course also emphasizes the integration of graphics engines, physics systems, and AI components to form cohesive game architectures, applying key software design principles such as modularity, scalability, and maintainability. Prerequisite: CS 3398 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4388. Computer Graphics.
This course examines the fundamental concepts and algorithms of computer graphics, including geometric transformations, scene representation, lighting and shading models, texture mapping, and animation. Students analyze mathematical and computational techniques used to construct and manipulate 2D and 3D graphical content. Topics include rendering pipelines, projection methods, and real-time visualization. Emphasis is placed on the implementation and evaluation of interactive graphics systems and the representation of visual data in digital environments. The course provides a foundation for studying modern graphics technologies and their applications in computing. Prerequisite: CS 3358 and [MATH 1317 or MATH 2321 or MATH 2417 or MATH 2471 or MATH 2472] with grades of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 4395. Independent Study in Computer Science.
This course allows interested undergraduate students to pursue an individual independent study under the supervision of a computer science faculty member. The course content depends on the chosen project but typically includes an in-depth study of the selected topic, a literature review, the implementation of an idea, the evaluation of said idea, and the presentation and a write-up of the project and its results. The selected topic must not already be covered in substantial detail in any other course offered by the Department of Computer Science. Prerequisite: Instructor approval.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Dif Tui- Science & Engineering
Grade Mode: Credit/No Credit

CS 4398. Software Engineering Project.
This course serves as an undergraduate-level capstone experience in software engineering, with a focus on model-driven development and object-oriented methods. Students work in teams to design, implement, verify, and validate a small but complete software system. The course integrates core topics including UML-based modeling, agile processes such as eXtreme Programming, requirements specification through use cases, dynamic and statechart modeling, and the Model-View-Controller architecture. Emphasis is placed on systematic test case derivation and automated regression testing using JUnit. Students also gain hands-on experience with version control via Git and apply structured, team-based development practices to manage complexity, coordinate activities, and deliver a high-quality software product within realistic project constraints. Prerequisite: CS 3398 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 5100. Advanced Computer Science Internship.
This course provides CS master's students on-the-job training supervised by computer professionals in industry internship programs. During the internship, students apply what they have learned in the classroom toward the internship work. They are required to submit a mid-term and a final report of their internship work, describing what they have learned as a result of the internship and what curriculum additions or improvements they would suggest as a result of the internship experience. Students need the approval of the department in advance to take this internship course. Prerequisite: Instructor approval.
1 Credit Hour. 0 Lecture Contact Hours. 20 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Graduate Assistantship|Exclude from Graduate GPA
Grade Mode: Leveling/Assistantships

CS 5199B. Thesis.
This course provides continued enrollment for graduates engaged in thesis research and writing in computer science. Work is conducted under the direct supervision of a thesis advisor and involves activities necessary for completing the thesis, such as data collection, analysis, preparation of written dissertation chapters, and oral defense of the thesis. Candidates may participate in systems implementation, computational research, software engineering, or other approved investigative approaches as appropriate to their study. Enrollment may be needed for each long semester while conducting research or writing to maintain steady progress until the thesis is submitted for binding.
1 Credit Hour. 1 Lecture Contact Hour. 0 Lab Contact Hours.
Grade Mode: Credit/No Credit

CS 5299B. Thesis.
This course provides continued enrollment for graduates engaged in thesis research and writing in computer science. Work is conducted under the direct supervision of a thesis advisor and involves activities necessary for completing the thesis, such as data collection, analysis, preparation of written dissertation chapters, and oral defense of the thesis. Candidates may participate in systems implementation, computational research, software engineering, or other approved investigative approaches as appropriate to their study. Enrollment may be needed for each long semester while conducting research or writing to maintain steady progress until the thesis is submitted for binding.
2 Credit Hours. 2 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Credit/No Credit

CS 5300. Professional Development of Graduate Assistants.
This course develops the professional, pedagogical, and technical skills required of master's-level graduate instructional and teaching assistants in Computer Science. Through weekly seminars, guest speakers, and structured practice, students explore effective teaching strategies for lower-division CS courses, ethical and legal responsibilities, classroom and laboratory management, and core technical support skills relevant to department labs. Activities such as a formal teaching presentation, peer feedback, and written reflections emphasize clear communication, professionalism, and effective support of undergraduate learners. This course does not earn graduate degree credit.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Graduate Assistantship|Exclude from Graduate GPA
Grade Mode: Leveling/Assistantships

CS 5301. Programming Practicum.
This course introduces graduate students to foundational programming concepts and practices for computing coursework. Topics include procedural and object-oriented programming, problem decomposition, program design, testing, debugging, and implementation in C++, with selected programming idioms from Java and Python. Students analyze and implement core data structures such as arrays, linked lists, stacks, queues, trees, and graphs, along with basic algorithms including searching, sorting, recursion, depth-first search, and breadth-first search. The course also examines contemporary programming workflows involving AI-assisted coding, code review, debugging, and responsible evaluation of AI-generated code.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from Graduate GPA|Leveling
Grade Mode: Leveling/Assistantships

CS 5302. Foundations of Data Structures and Algorithm Design.
This course examines classic data structures and the analysis of related algorithms through the lens of abstract data types. Students analyze the behavior of fundamental structures, including lists, stacks, queues, trees, graphs, and hash tables, while evaluating their impact on algorithmic performance regarding time and space complexity. Learners implement recursion and its various applications alongside elementary algorithms for sorting, searching, and hashing. Participants evaluate the trade-offs between dynamic and array-based implementations to determine appropriate structures for specific computational problems. By examining generic programming and heap structures, the course provides a comprehensive foundation for advanced programming and algorithm design.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from Graduate GPA
Grade Mode: Leveling/Assistantships

CS 5303. Foundations of Computer Architecture.
This course provides foundational instruction in computer architecture for graduate students requiring reinforcement of core concepts. Topics include arithmetic logic units, instruction set architectures, datapath and control design, pipelining, multiprocessing, input/output systems, memory hierarchies, virtual memory, low level programming techniques, and performance evaluation. Students examine architectural tradeoffs through analytical methods and quantitative performance analysis. Emphasis is placed on understanding how hardware structures implement computation and influence system behavior.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Exclude from Graduate GPA
Grade Mode: Leveling/Assistantships

CS 5305. Foundations of Operating Systems.
This course serves as a foundation course for computer science master's students who need reinforcement of fundamental concepts covered by CS 4328. The course examines the principles and design of modern operating systems. Topics include process and thread management, CPU scheduling, synchronization, interprocess communication, deadlocks, memory management, virtual memory, file systems, I/O, virtualization and cloud computing. Students explore system calls, kernel structure, and security mechanisms. Emphasis is placed on the interaction between hardware and software and on performance trade-offs. Programming assignments provide hands-on experience with system-level concepts using a Unix/Linux environment.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from Graduate GPA
Grade Mode: Leveling/Assistantships

CS 5306. Advanced Operating Systems.
This course examines the principles and design of modern operating systems. Topics include process and thread management, CPU scheduling, synchronization, interprocess communications, deadlocks, memory management, virtual memory, file systems, I/O, virtualization and cloud computing. Students explore system calls, kernel structure, and security mechanisms. Emphasis is placed on the interaction between hardware and software and on performance trade-offs. Programming assignments provide hands-on experience with system-level concepts using a Unix/Linux environment.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5310. Network and Communication Systems.
This course provides a study of network and communication systems. The course consists of three parts: foundations of data communications, essentials of computer networks and protocols, TCP/IP programming. Topics for data communications include types of networks, data types and their properties, network and application quality of services, modulations and multiplexing, and issues of signal transmissions. The computer networks and protocol part covers the core elements of computer networks and protocols such as sliding window protocols, LANs, Internet and TCP/IP. The programming part introduces TCP/IP networking socket APIs.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5313. Machine Learning and Applications.
This course provides a rigorous technical foundation in machine learning theory, algorithms, and real-world applications for graduate students. It covers supervised and unsupervised learning, high-dimensional statistical modeling, and advanced ML/AI topics, including ensemble methods and generative and discriminative models. Students learn to address regression and classification tasks using data-driven paradigms, including clustering, ensemble learning, and dimensionality reduction, with emphasis on mathematical derivations, algorithmic implementation, and challenges such as the curse of dimensionality. Methodology emphasizes systematic experimentation, optimization, and critical evaluation using modern machine learning tools and reproducible workflows. Students complete a domain-driven project that integrates full data science pipelines with novel or advanced modeling approaches in domains such as healthcare, finance, and vision.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Dif Tui- Science & Engineering
Grade Mode: Standard Letter

CS 5315. Responsible and Trustworthy AI.
This course examines the foundational principles and practices associated with responsible and trustworthy Artificial Intelligence (AI), introducing AI Engineering and approaches to developing AI systems with attention to reliability and risk. Topics include robustness, explainability, privacy, fairness, bias, and the use of generative AI models and machine learning in production environments. Students analyze the benefits, limitations, and trade-offs of these concepts and their integration into AI development. The course also reviews recent advancements and examines technical, regulatory, and ethical challenges within the AI domain. Prerequisite: CS 5313 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5316. Data Mining.
This course examines fundamental concepts, core methodologies, and recent developments in data mining. Typical topics include but are not limited to classification, cluster analysis, frequent pattern mining and related approaches for discovering structure and patterns in large#scale data sets. Relevant research training and practice opportunities are also provided. Through programming assignments and projects, students will enhance their understanding of theoretical concepts and gain hands-on experience on practical techniques. They will also explore popular data mining tool packages such as Weka, Orange, KNIME and RapidMiner.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5318. Principles of Programming Languages.
This course focuses on the principles of programming languages. Topics covered include programming paradigms, concepts of programming languages, formal syntax and semantics, and language implementation issues. Students examine principles used in specifying, designing, and implementing programming languages and review major paradigms including imperative, object-oriented, functional, logic, and concurrent programming. The course also explores how language features influence software development and implementation.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5325. Reinforcement Learning.
This course examines the foundational principles and modern methods of reinforcement learning, in which agents learn sequential decision-making through interaction with an environment based on reward signals. Topics include Markov decision processes, dynamic programming, Monte Carlo methods, temporal-difference learning, policy gradient methods, and deep reinforcement learning. Students investigate algorithms such as Q-learning, SARSA, actor-critic architectures, and proximal policy optimization, and explore applications including game-playing agents, robotics, and reinforcement learning from human feedback. Emphasis is placed on both theoretical analysis and practical implementation using Python, PyTorch, and OpenAI Gymnasium. Prerequisite: CS 5313 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5329. Algorithm Design and Analysis.
This course examines advanced algorithm design principles and computational complexity. Essential topics include core design techniques (such as divide-and-conquer, dynamic programming, and greedy methods), advanced algorithms for sorting, searching, and graphs, and the theory of NP-completeness. Students complete theoretical problem-solving and applied programming assignments to analyze the time and space complexity of algorithms and verify their correctness. Through these activities, students develop the advanced algorithmic toolkit necessary to design highly efficient and scalable solutions for a wide range of complex computational problems.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5331. Crafting Compilers.
This course provides a study of the design and implementation of modern compilers, with an emphasis on foundational principles and practical system construction. It covers the compilation pipeline, including lexical analysis, parsing, semantic analysis, and code generation. Students also learn about various optimization techniques. The course also introduces students to the challenges in compiling for modern architectures and the role of compilers in enabling high-performance execution. Through a sequence of incrementally structured projects, students build a compiler from the ground up, where each stage extends prior components to form a complete system, providing hands-on experience in constructing complex software systems and understanding how compilers translate high-level abstractions into efficient machine-level execution.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5332. Database Theory and Design.
This course examines the organization and management of data using relational database systems. Topics include data modeling, the Entity-Relationship model, and translating application data requirements into well-structured relational schemas. Fundamental design principles such as functional dependencies, normalization, and integrity constraints are introduced. Students gain practical experience creating databases and retrieving data through interactive SQL queries and programmatic access. Relational algebra, which provides the formal foundation for SQL, is also covered, along with underlying DBMS implementation technologies and advanced non-relational data models.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5334. Advanced Internet Information Processing.
This course integrates web programming languages and data storage techniques for advanced information processing in Internet applications. Topics include database and big data support, as well as application-specific information processing algorithms. Students examine methods for building web-based information processing systems using languages such as PHP, Java, Java Servlets/JSP, and Python. The course also covers design considerations, system integration, and performance aspects of web applications, with practical implementation exercises to support applied learning.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5338. Formal Languages.
This course covers the fundamental concepts and advanced topics in formal languages and automata theory. Through a combination of theoretical study and practical projects, students will learn how to analyze formal languages such as regular, context free, decidable and semi-decidable languages. Students will also learn how to create formal grammars such as regular and context free grammars, and design abstract machines such as finite state machines, pushdown automata and Turing machines. Other topics include Church-Turing thesis, Halting problem, reduction, computability and complexity.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5341. Advanced Network Programming.
This course covers some of the advanced concepts and programming skills in computer networks. The course looks into the details of the TCP/IP protocol suite, in particular, the IP, TCP, and UDP protocols. Students will learn TCP/IP based programming skills based on socket API. Main topics covered include advanced TCP/IP, API, multicasting and broadcasting, reliable communications, advanced I/O functions and options. The course extends the programming part of CS 5310 and gets students prepared for more advanced TCP/IP network programming.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5342. Robotics and Autonomous Systems.
This course provides a technical study of robotics and autonomous systems with an emphasis on algorithmic foundations and system implementation. Topics include robot kinematics, feedback control, probabilistic state estimation, perception using deep learning, motion planning, and coordination in multi-robot systems. Students evaluate real#world applications, including autonomous vehicles, and assess documented system#level impacts using empirical and scholarly sources. The course emphasizes hands-on development and testing of robotic systems through project-based learning, simulation environments, and real robots. Prerequisite: CS 5329 with grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5343. Wireless Communications and Networks.
This course covers the fundamental aspects of wireless communications and wireless and mobile networks. Topics include the electromagnetic spectrum and propagation modes; antenna properties such as radiation patterns, gain, and loss; and key challenges in wireless communication, including interference and fading. The course also examines signal encoding and digitization techniques such as pulse code modulation and delta modulation, as well as the principles, advantages, and limitations of spread spectrum methods. Cellular network design is addressed, including architecture, operation, and handoff management.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5346. Advanced Artificial Intelligence.
This course provides an introduction to Artificial Intelligence (AI) and generative AI, focusing on the analysis, design, implementation, and evaluation of AI systems. Topics include machine learning, knowledge representation, intelligent search, probabilistic reasoning, natural language processing, planning and decision-making, adversarial search, and generative models such as large language models, GANs, and VAEs. Students examine methods for developing and evaluating AI systems using established performance metrics and experimental methodologies. The course also addresses scalability, knowledge-based systems, and the analysis of technical, societal, and reliability considerations associated with AI systems.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5351. Parallel Processing.
This course explores practical aspects of parallel processing, including multi-core CPUs and shared-memory programming, GPUs and accelerator programming, and distributed-memory computers and message-passing programming. The lectures cover MPI, POSIX threads, OpenMP, CUDA, HIP, loop parallelization, parallel algorithms, amorphous data parallelism, atomic operations, prefix sums, performance measurement, parallelism bugs, and case studies of parallel programs. Students are given opportunities to gain applied knowledge and skills by developing, testing, and evaluating the performance of parallel software on various shared- and distributed-memory systems.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5352. Distributed Computing.
This course examines essential aspects of distributed and cloud computing. Topics include the history and evolution of distributed systems, architectures and models, system transparency, distributed time and clock synchronization, global states, and safety and fairness properties. The course also covers inter-process communication, concurrency control and atomicity, failure detection and recovery, fault tolerance including Byzantine failures, distributed consensus and coordination, remote method invocation, naming, and security. Students also examine cloud application development and case studies of distributed and cloud-based systems.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5361. Generative Artificial Intelligence.
This course introduces generative artificial intelligence, focusing on models such as autoencoders, variational autoencoders (VAEs), generative adversarial networks (GANs), and diffusion-based methods. Topics include text, image, and audio generation, as well as applications such as data augmentation and synthetic data generation. Students implement and evaluate generative models through practical assignments. The course also examines technical considerations related to bias, synthetic media, and intellectual property, along with methods used to assess generated outputs across different application contexts. Prerequisite: CS 5313 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5369A. Data Science And Visualization.
This course introduces fundamental and practical techniques for data science and visualization in Python, with a focus on end-to-end analytics pipelines. Topics include data wrangling, data cleaning, exploratory data analysis, unsupervised learning, and the design of interactive dashboards. Students work with multi-source datasets and implement reproducible workflows using contemporary data science libraries, version control, and dashboard frameworks. The course also examines pipeline design, data representation, and visualization techniques, along with introductory concepts in environment management and workflow organization. Consideration is given to how design and infrastructure choices relate to interpretability, reliability, and downstream analytical applications.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Topics
Grade Mode: Standard Letter

CS 5369Q. Recommender Systems.
This course provides an advanced graduate-level treatment of recommender systems, focusing on algorithms and research methods for personalized content and decision support. Topics include content-based and collaborative filtering, matrix factorization, sequence- and context-aware models, and hybrid architectures used in large-scale environments. Students design and implement recommendation pipelines in Python using interaction data, with emphasis on candidate generation, ranking, offline evaluation metrics, and analysis of online experiments such as A/B tests. The course also examines methods for evaluating system performance and analyzing the effects of design choices in recommender systems.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Topics
Grade Mode: Standard Letter

CS 5369Y. Green Computing.
This course provides a graduate-level introduction to computing approaches related to energy efficiency and resource usage. Topics include energy-efficient hardware and system design, software optimization techniques, and methods for analyzing energy consumption in computing systems. The course examines data center efficiency, resource management strategies, and scheduling approaches based on energy-related constraints. Students use measurement and profiling tools to evaluate system performance and energy usage. Case studies are used to analyze trade-offs among performance, resource utilization, and system design choices.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Course Attribute(s): Exclude from 3-peat Processing|Topics
Grade Mode: Standard Letter

CS 5375. Multimedia Computing.
This course introduces the fundamental concepts and techniques of digital multimedia. It covers the representation and compression of text, audio, images, and video, along with multimedia systems, applications, transmission, and the role of standardization in ensuring interoperability and efficient media exchange. Topics include multimedia systems and applications; graphics and image representation; image and video processing; visual perception and color models; lossless and lossy compression, including frequency-domain concepts; multimedia compression standards; and emerging technologies such as virtual and augmented reality.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5378. Advanced Computer Security.
This course examines advanced concepts in computer security, from abstract security policies to the design and analysis of secure systems and networks. Topics include security models, cryptographic techniques, common vulnerabilities in modern systems, and established security practices. Students analyze how attacks are conducted, how vulnerabilities are identified, and how defensive mechanisms are implemented and evaluated. The course also introduces methods for assessing emerging security challenges and reviewing current research in computer security. Emphasis is placed on understanding system security from both theoretical and practical perspectives.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5388. Advanced Computer Graphics.
This course examines the fundamental concepts and algorithms of computer graphics with an emphasis on advanced analysis and implementation. Topics include geometric transformations, scene representation, lighting and shading models, texture mapping, and animation. Students analyze and design complex graphics systems through the application of mathematical and computational techniques. Additional emphasis is placed on performance considerations, system-level design, and evaluation of rendering pipelines and visual representations. The course includes topics related to real-time graphics and modern visual computing systems.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5391. Survey of Software Engineering.
This course examines the software development life cycle with an emphasis on system analysis, design, and implementation practices used in contemporary software engineering. Topics include requirements engineering, architectural design, development methodologies based on data#flow and object#oriented models, and techniques for verification and validation. The course also introduces professional standards and ethical frameworks relevant to software engineering practice, treating ethical issues as objects of analysis rather than prescriptive rules. Students engage with both industry practices and research#informed approaches to understand tradeoffs, design decisions, and evaluation methods used in modern software systems.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5392. Formal Methods in Software Engineering.
This course examines formal methods in software engineering with a focus on static program analysis and verification techniques. Topics include propositional and first-order logic, soundness and completeness, computability concepts, and formal proof methods. The course covers model checking, temporal logics such as linear time temporal logic (LTL) and computation tree logic (CTL), and specification of system properties including safety and liveness. Additional topics include satisfiability-based verification, symbolic analysis, and approaches to analyzing concurrent systems. Students engage with tools and methods used to evaluate software correctness and analyze the capabilities and limitations of formal verification techniques. Prerequisite: CS 5391 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5393. Software Quality.
This course examines dynamic program analysis algorithms and their role in software quality assessment. Topics include software correctness, reliability, and robustness; testing methodologies such as unit, integration, and regression testing; and techniques including mutation testing, data flow analysis, and symbolic evaluation. The course also covers dynamic validation approaches, dependency analysis, and challenges in object-oriented testing. Students analyze quality assurance strategies and apply formal and empirical methods to evaluate software systems. Prerequisite: CS 5391 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5394. Advanced Software Engineering Project.
This course serves as a graduate-level capstone experience in software engineering. Students work in teams to design, implement, verify, and validate a software system of substantial scope. The course integrates concepts from the software engineering curriculum, including requirements analysis, architectural design, process modeling, and verification and validation. Emphasis is placed on structured, team-based development using contemporary development approaches aligned with established software architectures and processes. Students apply systematic methods to manage complexity, coordinate team activities, and evaluate software quality within realistic project constraints. Prerequisite: CS 5391 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5395. Independent Study in Advanced Computer Science.
This course provides graduate students with the opportunity to pursue an individualized area of advanced study in computer science not covered by the existing curriculum. Students engage in independent work under faculty supervision, focusing on advanced topics or research-oriented study. The course includes investigation of a defined topic and may involve implementation, analysis, or theoretical exploration. Students present their work in an oral presentation addressing the central topic of study. Prerequisite: Instructor Approval.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Credit/No Credit

CS 5396. Advanced Software Engineering Processes and Methods.
This course examines structured software engineering processes and methods, including object-oriented, aspect-oriented, feature-oriented, Cleanroom, PSP, TSP, Scrum, XP, and related approaches. Topics include process selection, software quality considerations, and the relationship between system requirements and development methodologies. The course also addresses tools and techniques for developing complex software systems, including interactive, mobile, and distributed applications. Students review literature on software engineering processes, methods, and tools, and analyze approaches to integrating process models with contemporary software development practices, including automated and agent-assisted development. Prerequisite: CS 5391 with a grade of "C" or better.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Standard Letter

CS 5399A. Thesis A.
This course represents a student's initial thesis enrollment for graduate students engaged in thesis research and development in computer science. Work is conducted under the supervision of a thesis advisor and involves activities necessary for completing the thesis, such as data collection, analysis, and preparation of written dissertation chapters. Students may engage in systems implementation, computational research, software engineering, or other approved investigative approaches appropriate to their area of study.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Credit/No Credit

CS 5399B. Thesis B.
This course provides continued enrollment for graduate students engaged in thesis research and writing in computer science. Work is conducted under the supervision of a thesis advisor and involves activities necessary for completing the thesis, including data collection, analysis, preparation of written dissertation chapters, and oral defense. Students may engage in systems implementation, computational research, software engineering, or other approved investigative approaches appropriate to their area of study.
3 Credit Hours. 3 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Credit/No Credit

CS 5599B. Thesis B.
This course provides continued enrollment for graduate students engaged in thesis research and writing in computer science. Work is conducted under the supervision of a thesis advisor and involves activities necessary for completing the thesis, including data collection, analysis, preparation of written dissertation chapters, and oral defense. Students may engage in systems implementation, computational research, software engineering, or other approved investigative approaches appropriate to their area of study.
5 Credit Hours. 5 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Credit/No Credit

CS 5999B. Thesis B.
This course provides continued enrollment for graduate students engaged in thesis research and writing in computer science. Work is conducted under the supervision of a thesis advisor and involves activities necessary for completing the thesis, including data collection, analysis, preparation of written dissertation chapters, and oral defense. Students may engage in systems implementation, computational research, software engineering, or other approved investigative approaches appropriate to their area of study.
9 Credit Hours. 9 Lecture Contact Hours. 0 Lab Contact Hours.
Grade Mode: Credit/No Credit