COMPUTER SCIENCE

Overview and Contact Information

Computer science is an exciting field with applications to many disciplines across the humanities, social sciences, and sciences. The main role of a computer scientist is that of a problem solver. A degree in the field signifies formal training in computational and analytical approaches to problem solving as well as the skills necessary to develop software to tackle new challenges. These computational approaches can be applied to a wide spectrum of problems, including protein folding and flexibility, modeling and forecasting bird migration, improving the capabilities of search engines to retrieve the most relevant documents, understanding how the connectedness provided by social networks impact the lives we lead, supporting scientists in the management and analysis of the data they collect, and more. In truth, it is difficult to think of a scenario in which the tools acquired in computer science do not provide a powerful advantage.

Honors

To graduate with honors in computer science, a student must complete a project and write an accompanying thesis. This is often a full year commitment, during which the student works closely with a faculty member to explore a topic in depth by reading research papers, writing programs, and experimenting with ideas. Preliminary research usually begins in the summer following the junior year, with the student submitting and defending a thesis proposal early in the fall of the senior year. Upon department approval of this proposal, the student will complete the research during the senior year, writing and defending the thesis in the spring. Some honors students attend conferences and/or coauthor papers with their mentors.

See Also

- Data Science (http://catalog.mtholyoke.edu/areas-study/data-science/)
- Engineering (http://catalog.mtholyoke.edu/areas-study/engineering/)

Contact Information

Lisa Ballesteros, Co-chair
Barbara Lerner, Co-chair
Eva Diaz, Academic Department Coordinator

200 Clapp Laboratory
413-538-2420
https://www.mtholyoke.edu/academics/find-your-program/computer-science (https://www.mtholyoke.edu/academics/find-your-program/computer-science/)

Learning Goals

The computer science curriculum is designed to encourage students to:

- Develop the critical thinking skills to solve problems by designing and implementing algorithms.
- Develop the analytical skills for reasoning about algorithmic complexity.
- Be able to design, implement, test, and document computer programs that solve substantial computational problems.
- Build skills for developing a working understanding of a complex code base and ability to effectively modify it.
- Be able to think at multiple levels of detail and abstraction.
- Develop a foundation that allows and encourages learning new and relevant skills and technologies as the field evolves.
- Understand the interplay between theory and practice.
- Understand the interplay between software and hardware.
- Be able to communicate clearly in written and oral form.
- Be able to work effectively on a team.

More specifically, students are expected to master the following concepts:

- Computer programming — including working knowledge of at least two programming languages in different paradigms.
- Data structures: an ability to use and implement fundamental abstract data types including queues, stacks, balanced search trees, hash tables, and graphs.
- Abstraction to manage complexity.
- Recursion and induction.
- Algorithmic problem-solving: an ability to design, code, analyze, and prove the correctness of algorithms using recursive divide-and-conquer, dynamic programming, and greedy approaches.
- The conceptual organization of computers—including both computer architecture (the hardware level) and operating system issues.
- Data storage on a computer.
- Applications of computing: an understanding of how computers, algorithms, programs, and/or data structures are used in several application areas.

Faculty

This area of study is administered by the Department of Computer Science:

Audrey Lee St. John, Professor of Computer Science

Barbara Lerner, Professor of Computer Science

Lisa Ballesteros, Jean E. Sammet Associate Professor of Computer Science

Heather Pon-Barry, Associate Professor of Computer Science

Alyxander Burns, Assistant Professor of Computer Science

Tony Liu, Assistant Professor of Computer Science

Murphy McCauley, Assistant Professor of Computer Science

Melody Su, Assistant Professor of Computer Science

Diane Uwacu, Assistant Professor of Computer Science

Requirements for the Major

A minimum of 40 credits:

Code	Title	Credits
Computer Science (36 credits)		
COMSC-151	Introduction to Computational Problem Solving	4
COMSC-205	Data Structures	4
COMSC-221	Introduction to Computing Systems	4
COMSC-225	Software Design and Development	4
COMSC-312	Algorithms	4

COMSC-322	Operating Systems	4
Three additional computer science courses: [1]		12
Two of these must be at the 300 level (8 credits)		
The third may be at either the 200 level or 300 level (4 credits)		
Mathematics (4 credits)		
MATH-232	Discrete Mathematics [2]	4
Humanities and Social Sciences [3, 4]		
Beyond those used to fulfill the College's distribution requirements:		
One additional designated Humanities course [5]		
One additional designated Social Science course [5]		
Total Credits		**40**

1. Independent study courses do not count as electives
2. Computer science majors who elect a mathematics or statistics minor may not count MATH-232 for credit in both mathematics or statistics and computer science
3. These courses can also count towards the College's Outside the Major requirement or towards the requirements of a second major and therefore are not counted directly toward the credits required for the major.
4. The ungraded option cannot be elected after declaration of the major in courses used to meet these requirements.
5. Designated courses are those that have been classified to meet the College's applicable distribution requirement.

Additional Specifications

- The skills and abstract reasoning of mathematics are especially important in computer science. It is strongly recommended that students take additional mathematics courses (at least through MATH-101 and MATH-102). MATH-211, is very useful for some fields, like machine learning and computer graphics.
- Students planning to pursue an advanced degree in computer science should include in their plans additional computer science courses and independent research leading to a thesis.

Requirements for the Minor

A minimum of 20 credits:

Code	Title	Credits
Computer Science		
COMSC-151	Introduction to Computational Problem Solving [1]	4
COMSC-205	Data Structures	4
COMSC-225	Software Design and Development	4
Two additional computer science courses (8 credits), including:		8
One at the 300 level		
The second can be at either the 200 level or 300 level		
Total Credits		**20**

1. COMSC-150 plusCOMSC-161 can substitute for COMSC-151

Course Advice

The Computer Science department offers multiple ways to begin study of computer science, suitable for students considering a computer science major or minor, those who may want to use computing within another major, and those who are curious about computer science and want to start with an introduction that is not programming-intensive.

For students considering a major or minor in computer science:

If you are interested in a major or minor in computer science, the recommended entry point is COMSC-151. This is a programming-intensive course introducing the use of computers as a problem-solving tool.

Students with prior programming experience may take a placement test to determine eligibility to skip the above and start with COMSC-205.

For students interested in programming but undecided about a computer science major or minor:

Some semesters, the department offers an alternative entry point: COMSC-150. This course focuses on the core constructs used in many programming languages, but does not include the topic of object-oriented programming. If a student then wishes to continue with more computer science study, COMSC-150 should be followed by COMSC-161; the combination of the two is equivalent to COMSC-151.

For student curious about computer science:

For students interested in exploring computer science with less emphasis on programming, the department typically offers at least one non-major course each semester (for example, COMSC-100, COMSC-106, or COMSC-109).

Course Offerings

COMSC-100 Computing and the Digital World
Not Scheduled for This Year. Credits: 4
An introduction to basic computer science concepts. Lectures will cover topics such as the origins of computing, computer architecture, artificial intelligence, and privacy and security. There will be some programming exercises.
Applies to requirement(s): Math Sciences
The department
Advisory: No prior study of computer science is expected. Students may not take this course after Computer Science 106, 150, or 151.
Notes: Course does not count toward the Computer Science major or minor.

COMSC-106 Fundamentals of Applied Computing
Spring. Credits: 4
Have you ever used Google's image search tool and wondered how the search results were found? Why is it so difficult for a computer to "see" as we do? Computer scientists are actively researching how to approach this challenge of "computer vision." This course will introduce the fundamentals of applied computing using computer vision as a motivating theme. Students will learn foundations of programming (in the Python programming language) before working with computational tools more independently.
Applies to requirement(s): Math Sciences
A. St. John
Advisory: No prior study of computer science is expected. Students may not take this course after Computer Science 100, 150, or 151.
Notes: Course does not count toward the Computer Science major or minor.

COMSC-107 iDesign Learning Lab
Fall. Credits: 2
When charting a path through college and beyond, a metacognitive framework can provide scaffolding for intentional reflection. Situated in the Fimbel Maker & Innovation lab, this course will leverage tangible activities to ground discussions on factors and strategies that impact learning. For example, embedding a microcontroller to create motion-sensitive lights in a ballet skirt parallels the cycle of self-regulated learning. No prior experience with electronics or computer science is assumed, and students will work with hands-on tutorials that teach the basics required to develop their own interactive technology projects.
A. St. John
Restrictions: Course limited to sophomores, juniors and seniors
Notes: Half semester.

COMSC-109 iDesign Learning Incubator
Spring. Credits: 4
Learning to make and making to learn - this course challenges students to engage in both by being situated in the Fimbel Maker & Innovation Lab. Hands-on introductory activities provide the surprisingly minimal level of comfort and background required to produce prototypes of interactive technology products. These tangible experiences are leveraged to prompt students to reflect on factors and strategies that impact effective learning. For example, developing a music box parallels the cycle of self-regulated learning, while designing a greeter robot grounds discussion on the role of belongingness in a learning environment.
Applies to requirement(s): Math Sciences
A. St. John
Advisory: No prior experience with electronics or computer science is assumed.

COMSC-112 Topics in Computer Science Problem Solving

COMSC-112AE Topics in Computer Science: 'AI Ethics'
Not Scheduled for This Year. Credits: 2
Artificial intelligence (AI) is rapidly changing our world, from the way we drive to the way we communicate. But what are the ethical implications of these changes? In this course, we will explore the ethical challenges and opportunities posed by AI. We will discuss topics such as data privacy, fairness, bias, accountability, and transparency. We will also examine the role of AI in society and its potential impact on our values and our way of life.
Applies to requirement(s): Meets No Distribution Requirement
D. Uwacu
Advisory: No prior knowledge of AI is required.

COMSC-132 Engineering for Everyone
Spring. Credits: 4
Engineers change the world we live in every day by developing technologies that influence nearly every aspect of our lives. In this course, we will study how engineered things shape the world we live in. Students will engage in a team-based, hands-on engineering design project, from brainstorming solutions to a contemporary problem, to building, testing, and iterating design solutions. In the process, students will learn basic programming and fabrication skills. We will reflect together on the ethics of engineering design, and leave with a more nuanced understanding of the ways technology and society interact. Who decides what technologies matter? What is a "good" technological solution, and for whom is it "good"?
Crosslisted as: PHYS-132
Applies to requirement(s): Math Sciences
M. Su
Advisory: This course has no prerequisites and is recommended for all students interested in engineering and technology.
Notes: Students interested in continuing with the Engineering Nexus are strongly recommended to take the course.

COMSC-133 Topic Problem Solving

COMSC-133DV Data Visualization: Design and Perception
Not Scheduled for This Year. Credits: 4
Data visualizations such as graphs, charts, and infographics are everywhere! But creating data visualizations which communicate effectively is not a simple task. In this introductory course, students will explore how design influences the ways that data are understood and how they can use this to craft effective visualizations for different types of data. Experience working with data, design, or data visualization are not expected; students will work on a series of projects which will build these skills over the semester.
Applies to requirement(s): Math Sciences
A. Burns

COMSC-150 Introduction to Computer Science
Not Scheduled for This Year. Credits: 4
Introduction to the field of computer science. Introduces students to Python programming including algorithms, basic data structures (lists, dictionaries), and programming techniques. Does not include object-oriented programming.
Applies to requirement(s): Math Sciences
H. Ghosh, The department
Notes: Students pursuing a Computer Science major or minor or the Data Science major should take COMSC-161 following this course. Alternatively, students may wish to consider the more programming-intensive course COMSC-151.

COMSC-151 Introduction to Computational Problem Solving
Fall and Spring. Credits: 4
Introduces students to algorithms, basic data structures, and programming techniques. Students learn computing principles by exploring problems drawn from a broad set of domains, such as cryptography, data analysis and games.
Applies to requirement(s): Math Sciences
A. Burns, H. Pon-Barry, D. Uwacu
Coreq: COMSC-151L.
Notes: This course is programming-intensive and includes the topic of object-oriented programming. Students may wish to consider COMSC-150 as an alternative entry point that does not cover the topic of object-oriented programming.

COMSC-161 Introduction to Computer Science Part 2: Object-Oriented Programming

Not Scheduled for This Year. Credits: 2

This course builds on the programming concepts learned in COMSC-150, covering object-oriented programming and introducing the Java programming language.

A. Burns

Prereq: COMSC-150 or placement test.

Advisory: Students may not take this course after Computer Science 151. Just getting started with computer science? We recommend taking the CS Problem Solving Assessment. You can access it via Gradescope with Entry Code RWG253.

Notes: Half semester. The combination of COMSC-150 and COMSC-161 serves as an alternate prerequisite route for COMSC-205 Data Structures.

COMSC-205 Data Structures

Fall and Spring. Credits: 4

This course builds on the basic programming concepts learned in Computer Science 151, shifting the focus to the organization of data in order to improve efficiency and simplicity of programs. Topics include the study of abstract data types and data structures (such as linked lists, stacks, queues, and binary trees). This course is programming-intensive and introduces the Java programming language.

Applies to requirement(s): Math Sciences

B. Lerner, T. Liu

Prereq: One of the following 1) COMSC-151 (with a grade of C or better), 2) COMSC-161 (with a grade of C or better), 3) COMSC-150 (with a grade of C or better) and COMSC-121. Coreq: COMSC-205L.

Advisory: This course cannot be taken by students who have completed COMSC-201 or COMSC-211.

COMSC-221 Introduction to Computing Systems

Fall and Spring. Credits: 4

This course looks at the inner workings of a computer and computer systems. It is an introduction to computer architecture. Specific topics include assembly language programming, memory, and parallelism. This course is programming intensive.

Applies to requirement(s): Math Sciences

L. Ballesteros

Prereq: COMSC-201 or COMSC-205; and MATH-232. Coreq: COMSC-221L.

Advisory: The department recommends, but does not require, that students take COMSC-225 prior to COMSC-221.

COMSC-225 Software Design and Development

Fall and Spring. Credits: 4

Building large software systems introduces new challenges to software development. Appropriate design decisions and programming methodology can make a major difference in developing software that is correct and maintainable. In this course, students will learn techniques and tools that are used to build correct and maintainable software, improving their skills in designing, writing, debugging, and testing software. Topics include object-oriented design, testing, design patterns, and software architecture. This course is programming intensive.

Applies to requirement(s): Math Sciences

B. Lerner, M. Su

Prereq: COMSC-205 (with a grade of C or better).

COMSC-226 Engineering Robotic Systems

Spring. Credits: 4

This intermediate-level course presents a hands-on introduction to robotics. Each student will construct and modify a robot controlled by an Arduino-like microcontroller. Topics include kinematics, inverse kinematics, control-theory, sensors, mechatronics, and motion planning. Material will be delivered through one weekly lecture and one weekly guided laboratory. Assignments include a lab-preparatory homework, guided lab sessions, and out-of-class projects that build upon the in-class sessions. Students have access to the Fimbel Maker and Innovation lab for fabricating and demonstrating their robots.

Applies to requirement(s): Math Sciences

M. Su

Restrictions: This course is limited to first-years and sophomores.

Prereq: COMSC-150 or COMSC-151.

COMSC-235 Applications of Machine Learning

Fall. Credits: 4

This course provides a practical and conceptual introduction to machine learning. Through programming projects and work with real-world data, we will study the motivations behind common machine learning algorithms, and the properties that determine whether or not they will work well for a particular task. We will also study practical applications of these algorithms to problems in areas such as speech, language, social sciences, and biology. Topics may include: supervised learning, classification, regression, clustering, decision trees, support vector machines, Naïve Bayes, neural networks and reinforcement learning.

Applies to requirement(s): Math Sciences

H. Pon-Barry

Prereq: COMSC-205.

Advisory: Students may not take this course after COMSC-335.

COMSC-243 Topic

COMSC-243GP Topic: 'Introduction to Game Programming'

Fall. Credits: 4

Video games are not only engaging to play, but challenging and fun to program. Many games are in fact simulations: they define a game world, and model that world and the interactions of elements in it. To program games, one must excel at this type of modeling while simultaneously handling real-time input and output to create a compelling experience. This project-based course explores techniques at the heart of game programming. By working through design and coding activities both with a team and independently, students will strengthen their core programming skills, their ability to model complex problems, and their skills for developing software in a team. This course is programming-intensive.

Applies to requirement(s): Math Sciences

J. McCauley

Prereq: COMSC-225.

COMSC-243HC Topic: 'Human-Computer Interaction'
Not Scheduled for This Year. Credits: 4
Human-computer interaction (HCI) is a multidisciplinary field exploring the relationships between people and computers. This broad area includes the study of topics such as how people interact with technology, how design impacts what people do with technology, and methodologies for designing new technologies that really works for people. In this class, students will learn about and apply human-centered design principles, employ common research methods in the field, and explore recent literature in the field.
Applies to requirement(s): Math Sciences
A. Burns
Prereq: COMSC-205 (may take concurrently). Prereq: COMSC-205 (may take concurrently).

COMSC-243HR Topic: 'Human-Robot Interaction'
Spring. Credits: 4
Human-Robot Interaction is an interdisciplinary field that examines a broad set of questions about robots that are designed to interact with humans (e.g., educational, assistive, and service robots). How does the behavior and appearance of a robot change how humans perceive and interact with it? How can we design and program robots that are natural, trustworthy, and effective? In this course, students learn the algorithmic foundations of interactive robots, gain experience building and evaluating interactive robots, and read and present scholarly research papers. Class time is split between lecture, presentations by students, discussions, and hands-on activities.
Applies to requirement(s): Math Sciences
H. Pon-Barry
Prereq: COMSC-205.

COMSC-243ST Topic: 'Introduction to Search Technologies'
Spring. Credits: 4
The vast amount of unstructured and structured data on the web and in organizational databases has increased the need for approaches to processing large volumes of text. Such analyses help researchers and businesses to gain insights -- that would otherwise be too resource- and time-consuming to do manually -- into issues such as how much a consumer can be expected to spend in a particular context, the rise of hate groups and their impact on social media, or to whom a newly discovered manuscript may be attributed. In this course, students are introduced to tools and techniques used to gain these insights, such as Map-Reduce and Sentiment Analysis, in the context of Natural Language Processing and search technologies (e.g., Google).
Applies to requirement(s): Math Sciences
L. Ballesteros
Prereq: COMSC-150 or COMSC-151.

COMSC-243SW Topic: 'Computing Systems Workshop'
Not Scheduled for This Year. Credits: 4
Beneath the polished surface of high-level programming languages like Python and consumer devices like gaming consoles and smartphones lie the elemental parts of computer systems -- elements like hardware components, operating systems, and digital logic. This course will use a hands-on approach combining both hardware and software as a way of understanding such systems at a low level. Students will have the chance to construct various tangible projects using Raspberry Pi computers and will have access to the Fimbel Maker & Innovation lab. Specific topics will touch on low level data representation, sound generation, and the classic Nintendo Entertainment System.
Applies to requirement(s): Math Sciences
J. McCauley
Prereq: COMSC-150 or COMSC-151.

COMSC-295 Independent Study
Fall and Spring. Credits: 1 - 4
The department
Instructor permission required.

COMSC-311 Theory of Computation
Not Scheduled for This Year. Credits: 4
Are there any limits to what computers can do? Does the answer to this question depend on whether you use a PC or a Mac? Is C more powerful than Python? This course explores these questions by investigating several models of computation, illustrating the power and limitations of each of these models, and relating them to computational problems and applications. Topics include finite state automata, pushdown automata, grammars, Turing machines, the Halting Problem, and NP-completeness.
Applies to requirement(s): Math Sciences
A. St. John
Prereq: COMSC-205 and MATH-232.
Notes: This 3-minute student-created video gives an overview: https://www.youtube.com/watch?v=SV57Yv8BXBc.

COMSC-312 Algorithms
Fall and Spring. Credits: 4
How does Google Maps find the best route between two locations? How do computers help to decode the human genome? At the heart of these and other complex computer applications are nontrivial algorithms. While algorithms must be specialized to an application, there are some standard ways of approaching algorithmic problems that tend to be useful in many applications. Among other topics, we explore graph algorithms, greedy algorithms, divide-and-conquer, dynamic programming, and network flow. Students learn to recognize when to apply each of these strategies as well as to evaluate the expected runtime costs of the algorithms they design.
Applies to requirement(s): Math Sciences
A. St. John, D. Uwacu
Prereq: COMSC-205 and MATH-232.

COMSC-316 Developing Innovative Software
Not Scheduled for This Year. Credits: 4
Tired of writing programs that nobody ever uses? Then, this is the course for you. Many people come up with novel ideas for software, but lack the resources or ability to develop the software. Students will apply their programming skills to develop and deliver software based on the requirements of a client. Students will learn critical communication skills required to work with a client, work in teams with classmates, and experience the software lifecycle from requirements elicitation through delivery. Students will synthesize many topics learned in prior courses as well as explore new technologies required to complete a specific project. Programming intensive.
Applies to requirement(s): Math Sciences
B. Lerner
Prereq: COMSC-225.

COMSC-322 Operating Systems
Fall and Spring. Credits: 4
An introduction to the issues involved in orchestrating the use of computer resources. Topics include operating system evolution, memory management, virtual memory, resource scheduling, multiprogramming, deadlocks, concurrent processes, protection, and design principles. Course emphasis: understanding the implications of OS design on the programs you run and write (i.e., on their security, performance, etc.). **This course is programming intensive**.
Applies to requirement(s): Math Sciences
J. McCauley
Prereq: COMSC-221 and COMSC-225.

COMSC-334 Artificial Intelligence
Spring. Credits: 4
Artificial Intelligence, as a field, has grown from its humble beginnings in science fiction to become one of the broadest fields in computer science, encompassing an incredibly wide array of topics. One of the common threads between these topics is "How do we build computer systems which exhibit logic and reason?" or rather "How do we build systems which can solve problems intelligently without resorting to brute force?" We'll cover a few major topics in this course, most notably search, logical reasoning, and planning as well as game playing/theory, uncertain reasoning, and graphical models. This course is programming intensive.
Applies to requirement(s): Math Sciences
L. Ballesteros, J. Burroni
Prereq: COMSC-225 and MATH-232.

COMSC-335 Machine Learning
Spring. Credits: 4
How does Netflix learn what movies a person likes? How do computers read handwritten addresses on packages, or detect faces in images? Machine learning is the practice of programming computers to learn and improve through experience, and it is becoming pervasive in technology and science. This course will cover the mathematical underpinnings, algorithms, and practices that enable a computer to learn. Topics will include supervised learning, unsupervised learning, evaluation methodology, and Bayesian probabilistic modeling. Students will learn to program in MATLAB or Python and apply course skills to solve real world prediction and pattern recognition problems. Programming Intensive.
Applies to requirement(s): Math Sciences
M. Su
Prereq: A grade of C or better in COMSC-205, MATH-232, and a Calculus course (MATH-101, MATH-102, or MATH-203).
Advisory: Preference will be given to Computer Science seniors in need of a final 300-level elective and Data Science seniors.

COMSC-341 Topics

COMSC-341CC Topics: 'Compiler Design'
Not Scheduled for This Year. Credits: 4
Principles and practices for the design and implementation of compilers and interpreters. Will cover the stages of the compilation and execution process: lexical analysis; parsing; symbol tables; type systems; scope; semantic analysis; intermediate representations; run-time environments and interpreters; code generation; program analysis and optimization; and garbage collection. Students will construct a full compiler.
Applies to requirement(s): Math Sciences
The department
Prereq: COMSC-221, COMSC-225, and COMSC-312.

COMSC-341CD Topics: 'Causal Inference for Data Science'
Spring. Credits: 4
You might have heard the phrase "correlation is not causation" - but then, what is causation? For example, how did scientists determine that smoking causes lung cancer? This course will explore how to ask and answer causal questions using data. We will learn the fundamentals of estimating cause-and-effect relationships, drawing from foundations in computer science, statistics, and economics. We will also use modern data science tools coding in Python to run simulations, work with data, and communicate our findings. Students will get hands-on experience thinking through causal study design and analyzing data across real-world applications in healthcare, public policy, education, and more. This course will have a substantial mathematical component, building off probability concepts seen in MATH-232.
Applies to requirement(s): Math Sciences
T. Liu
Prereq: COMSC-205 and MATH-232.

COMSC-341CV Topics: 'Computer Vision'
Fall. Credits: 4
This course provides an introduction to image analysis and 3D interpretation from image data. It uncovers the mystery behind standard techniques in image processing like filtering, edge detection, stereo vision, flow, etc. Math lovers, this course is for you! Throughout the semester, each student will develop their own computer vision library through programming assignments. Furthermore, students will learn about newer, advanced machine-learning-based computer vision algorithms.
Applies to requirement(s): Math Sciences
M. Su
Prereq: COMSC-205, MATH-211, and Calculus (MATH-101, MATH-102, or MATH-203), all with grade of C or better.

COMSC-341GP Topics: 'Game Programming'
Not Scheduled for This Year. Credits: 4
Video games are not only fun to play but interesting and challenging to program, involving elements that are useful in programming other sorts of systems as well. They incorporate graphics, audio, and animation, must model relatively complex systems, and often have relatively strict requirements on timing. In this course, we explore techniques behind game implementation by implementing some of our own. This course is programming (and gaming) intensive.
Applies to requirement(s): Math Sciences
J. McCauley
Instructor permission required.
Prereq: COMSC-225 and either MATH-100 or equivalent as indicated by the math placement test or completion of a higher level math course.

COMSC-341NL Topics: 'Natural Language Processing'
Fall. Credits: 4
This course provides an introduction to natural language processing, the discipline of enabling computers to process and understand human language. We will learn fundamental techniques for automated text and speech analysis and understanding, with insights from linguistics. Students will get hands-on practice implementing computational algorithms, reading scholarly research articles and will design and carry out an independent final project.
Applies to requirement(s): Math Sciences
H. Pon-Barry
Prereq: COMSC-225, MATH-232, and a Calculus course (MATH-101, MATH-102, or MATH-203).

COMSC-341NP Topics: 'Intro to Networking Architecture and Protocols'
Not Scheduled for This Year. Credits: 4
This course is an introduction to computer networking with a focus on the Internet. At the high level, we will emphasize concepts and principles which have contributed to the Internet's success scaling from its modest beginnings to a system used by over half of the world's population. At the low level, we will survey techniques, technologies and protocols that underlie networks, as well as key protocols built atop these networks. Specific topics include layering, routing, addressing, reliable delivery, congestion control, DNS, HTTP, and others.
Applies to requirement(s): Math Sciences
J. McCauley
Prereq: COMSC-221 and COMSC-312.

COMSC-341RP Topics: 'Robotics Planning Algorithms'
Spring. Credits: 4
Robotics planning is a fundamental skill for developing autonomous robots. This course will introduce students to the fundamental concepts and algorithms of robotics planning. Students will learn how to apply the concepts they have learned in Data Structures to implement and analyze the performance of popular planning algorithms. Students will also learn about the advancements and gaps that exist today in robotics navigation, manipulation, and collaboration.
Applies to requirement(s): Math Sciences
D. Uwacu
Prereq: COMSC-205 (may be taken concurrently). Prereq: COMSC-205 (may be taken concurrently).
Advisory: This course is ideal for students who are interested in developing autonomous robots. It is also a good course for students who are interested in learning more about the latest advancements in robotics planning research.

COMSC-341RT Topics: 'Applied Rigidity Theory'
Not Scheduled for This Year. Credits: 4
Suppose you were asked to: control a formation of robots to cooperatively carry an object; engineer a bridge using metal beams and bolts; or develop a web app to assist in drug design. The area of "rigidity theory" provides tools for approaching any of these tasks. We start with fundamental results from classical rigidity theory, culminating in a combinatorial characterization of rigid 2D "bar-and-joint frameworks" and a graph-theoretic algorithm. We then shift to specific applications drawn from domains such as robotics, structural engineering and computational biology. In small groups, students investigate a research question and produce an associated pedagogical or computational tool.
Applies to requirement(s): Math Sciences
A. St. John
Prereq: COMSC-205 and MATH-232.

COMSC-341TE Topics: 'Text Technologies for Data Science'
Not Scheduled for This Year. Credits: 4
This course focuses on text analysis and technologies. We look at the challenges of working with massive amounts of unstructured vs semi-structured vs structured data. In that context, we explore some of the ways that statistical analyses are applied to things like search, categorization e.g. spam filtering, recommender systems, plagiarism detection, and hidden message finding.
Applies to requirement(s): Math Sciences
L. Ballesteros
Prereq: COMSC-205.

COMSC-341VB Topics: 'Technology for the Visually Impaired and Blind'
Fall. Credits: 4
Life for the visually impaired and blind (VIB) can be very challenging. Both low-tech devices, such as white canes, and high-tech devices, such as Siri, can help VIB people overcome some of these challenges. In this course, we will read about, discuss, and experiment with devices across the low tech to high tech spectrum, to understand how they work and how effective they are. We will also look at technology being created to support VIB programmers, data scientists, and other professionals in their work. We will learn the routine things we should be doing to make websites and apps accessible to the VIB, and also stretch our imaginations and awareness about what may be possible in the future.
Applies to requirement(s): Math Sciences
B. Lerner
Prereq: COMSC-225.

COMSC-343 Programming Language Design and Implementation
Not Scheduled for This Year. Credits: 4
Ever wonder why there are so many semicolons in Java programs, or what it would mean for a language to not be object-oriented? In this course, we will explore issues related to the design and implementation of programming languages. Along the way, we will discover answers to these questions and more. Topics will include syntax, semantics, runtime support for languages as well as an introduction to functional programming.
Applies to requirement(s): Math Sciences
B. Lerner
Prereq: COMSC-225.

COMSC-395 Independent Study
Fall and Spring. Credits: 1 - 8
The department
Instructor permission required.