

**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**

Audrey St. John, Chair  
Wendy Queiros, Academic Department Coordinator  
200 Clapp Laboratory  
413-538-2420  
https://www.mtholyoke.edu/acad/computerscience (https://www.mtholyoke.edu/acad/computerscience/)

**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:  
Valerie Barr, Jean E. Sammet Professor of Computer Science, Teaching  
Spring Only  
Audrey Lee St. John, Professor of Computer Science  
Barbara Lerner, Professor of Computer Science  
Lisa Ballesteros, Associate Professor of Computer Science  
Heather Pon-Barry, Associate Professor of Computer Science, Teaching  
Fall Only  
Murphy McCauley, Assistant Professor of Computer Science  
Melody Su, Assistant Professor of Computer Science  
Shruti Biswal, Visiting Lecturer in Computer Science  
Ozan Erat, Visiting Lecturer in Computer Science

**Requirements for the Major**

A minimum of 40 credits:

<table>
<thead>
<tr>
<th>Code</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMSC-151</td>
<td>Introduction to Computational Problem Solving</td>
<td>4</td>
</tr>
<tr>
<td>COMSC-205</td>
<td>Data Structures</td>
<td>4</td>
</tr>
<tr>
<td>COMSC-221</td>
<td>Introduction to Computing Systems</td>
<td>4</td>
</tr>
<tr>
<td>COMSC-225</td>
<td>Software Design and Development</td>
<td>4</td>
</tr>
<tr>
<td>COMSC-312</td>
<td>Algorithms</td>
<td>4</td>
</tr>
</tbody>
</table>
Requirements for the Minor
A minimum of 20 credits:

<table>
<thead>
<tr>
<th>Code</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMSC-322</td>
<td>Operating Systems</td>
<td>4</td>
</tr>
<tr>
<td>Two additional computer science courses: 2</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Two of these must be at the 300 level (8 credits)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The third may be at either the 200 level or 300 level (4 credits)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mathematics (4 credits)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATH-232</td>
<td>Discrete Mathematics</td>
<td>4</td>
</tr>
<tr>
<td>Humanities and Social Sciences 5, 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beyond those used to fulfill the College’s distribution requirements:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>One additional designated Humanities course 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>One additional designated Social Science course 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Credits</td>
<td></td>
<td>40</td>
</tr>
</tbody>
</table>

1. Any COMSC-151 offering, for example COMSC-151AA, COMSC-151AR, COMSC-151DS, COMSC-151HC, or COMSC-151MD
2. Independent study courses do not count as electives
3. Students may also count PHIL-225 as an elective because of its strong connection to computer science
4. 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
5. These courses can also count towards the College’s Outside the Major requirement or towards the requirements of a second major
6. The ungraded option cannot be elected after declaration of the major in courses used to meet these requirements
7. 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.

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 any COMSC-151 offering. This is a programming-intensive course introducing the use of computers as a problem-solving tool. There are a variety of COMSC-151 offerings (for example, COMSC-151AA, COMSC-151AR, COMSC-151DS, COMSC-151HC, and COMSC-151MD) which all cover the same essential material but with differing motivating examples.

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 concepts 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, COMSC-108, or COMSC-109).

Course Offerings
COMSC-100 Computing and the Digital World
Fall. 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
S. Biswal
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
S. Biswal, 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 and Spring. 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
Instructor permission required.
Notes: Half semester.

COMSC-108 Computing and Dance
Not Scheduled for This Year. Credits: 4
Designers are continually innovating ways of incorporating technology into today's world, from apps that monitor physical activity to 3D-printed toe shoes to dancing avatars trained via Machine Learning. The recent emergence of low-cost, user-friendly components makes this new world of design accessible to a broad community. In this course, students will think critically about technologies that can enhance dance technique and performance. Through a sequence of hands-on workshops on electronics basics and microcontroller programming, students will gain the surprisingly minimal level of comfort and background necessary to learn tools to produce prototypes and address these dance-related technologies.
Applies to requirement(s): Math Sciences
L. Ballesteros

COMSC-109 iDesign Studio
Not Scheduled for This Year. Credits: 4
Designers are continually innovating ways of incorporating technology into today's world, from projections of butterflies on Grammy performance dresses to "smart" purses that sense when your wallet is missing. The recent emergence of low-cost, user-friendly components is making this new world of design accessible to a broad community. In this course, students will think critically about products already in the marketplace and will be given the tools to create their own designs. A sequence of hands-on workshops on electronics basics and microcontroller programming will provide the surprisingly minimal level of comfort and background in technology required to produce prototypes of these designs.
Applies to requirement(s): Math Sciences
The department

COMSC-120 Introduction to R
Spring. Credits: 1
An introduction to the programming language R and how it can be used for statistical analysis and visualization of data. Students will learn how to write basic R programs that can read, write, and manipulate data. They will make use of R functions for executing common statistical analysis and learn how to display the results using graphs and charts. Through a series of projects, students will get experience with writing their own functions, learn how to make use of R documentation and how to extend their own knowledge of the language.
Applies to requirement(s): Meets No Distribution Requirement
V. Barr
Notes: Credit/no credit grading.

COMSC-121 Object-Oriented Programming
Not Scheduled for This Year. Credits: 1
This course will introduce object-oriented programming to students who have a foundation in Python programming and are interested in continuing on to COMSC-205 Data Structures. It includes coverage of classes, objects, methods, and sub-typing.
Applies to requirement(s): Meets No Distribution Requirement
The department
Prereq: COMSC-150 or computer science placement test.

COMSC-122 Java Programming Language
Not Scheduled for This Year. Credits: 1
This course will teach the Java programming language to students who already have programming ability in another object-oriented programming language.
Applies to requirement(s): Meets No Distribution Requirement
The department
Prereq: COMSC-205PY.

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
The department
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-150 Introduction to Computer Science**

*Fall and Spring.* 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*

L. Ballesteros, A. St. John

*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**

Thematic introduction to the field of computer science. Draws on problems found in the thematic focus of each topics course. All topics courses within COMSC-151 cover the same concepts and skills and satisfy requirements in the Computer Science major and minor as well as the Data Science major. 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-151AA Introduction to Computational Problem Solving: ‘Algorithmic Arts’**

*Not Scheduled for This Year. Credits: 4*

Introduces students to algorithms, basic data structures, and programming techniques. Explores computation as an artistic medium, examining a range of computational art practices. By combining aspects of a studio art course, a media art survey, and an introductory computing lab, course participants will develop a solid foundation in computer programming approaches and techniques as they pertain to art production as well as an understanding of their emerging importance in the contemporary art world.

*Applies to requirement(s): Math Sciences*

The department

Coreq: COMSC-151AA L

*Notes: Additional seats will open for all students after first year students have registered.*

**COMSC-151AR Introduction to Computational Problem Solving: ‘Artificial Intelligence’**

*Not Scheduled for This Year. Credits: 4*

Introduces students to algorithms, basic data structures, and programming techniques, and basic methods from artificial intelligence. Includes discussion of foundational papers in AI. Programming exercises will explore what is necessary in order to get computers to operate in ways that seem intelligent such as in game play or solving puzzles.

*Applies to requirement(s): Math Sciences*

The department

Coreq: COMSC-151AR L

*Notes: Additional seats will open for all students after first year students have registered.*

**COMSC-151CP Introduction to Computational Problem Solving: ‘Computing Principles’**

*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*

B. Lerner, A. St. John

Coreq: COMSC-151CP L

**COMSC-151DS Introduction to Computational Problem Solving: ‘Big Data’**

*Fall. Credits: 4*

Introduces students to algorithms, basic data structures, and programming techniques, and focuses on data collection, preparation, analysis. Explores programming for data manipulation, the presentation and representation of data, and the ethics of working with data at scale.

*Applies to requirement(s): Math Sciences*

B. Lerner

Coreq: COMSC-151DSL

**COMSC-151EN Introduction to Computational Problem Solving: ‘Environmental Studies’**

*Not Scheduled for This Year. Credits: 4*

This course introduces students to algorithms, basic data structures, and programming techniques. Students will explore using computing to interpret data relating to global temperature changes, ocean currents, earthquakes, and water quality.

*Applies to requirement(s): Math Sciences*

The department

Coreq: COMSC-151ENL

**COMSC-151HC Introduction to Computational Problem Solving: ‘Humanities Computing’**

*Not Scheduled for This Year. Credits: 4*

Introduces students to algorithms, basic data structures, and programming techniques. Students will explore solving problems that arise in humanities disciplines: various forms of text analysis, image manipulation, animation, and sound manipulation.

*Applies to requirement(s): Math Sciences*

The department

Coreq: COMSC-151HCL

**COMSC-151MD Introduction to Computational Problem Solving: ‘Computers in Medical Technology’**

*Not Scheduled for This Year. Credits: 4*

Introduces students to algorithms, basic data structures, and programming techniques. Students will explore solving problems that arise in using computers to interpret biological data such as DNA sequences, cancer tumor shape/size, and cardiac waveforms.

*Applies to requirement(s): Math Sciences*

The department

Coreq: COMSC-151MDL

*Notes: Additional seats will open for all students after first year students have registered.*

**COMSC-151SG Introduction to Computational Problem Solving: ‘Computing for Social Good’**

*Fall. Credits: 4*

Introduces students to algorithms, basic data structures, and programming techniques. Includes discussion of the ways in which computing can improve human lives and society, such as improving accessibility for people with disabilities, or helping organize a rescue team during an emergency.

*Applies to requirement(s): Math Sciences*

B. Lerner

Coreq: COMSC-151SG L
COMSC-161 Introduction to Computer Science Part 2: Object-Oriented Programming
Fall and Spring. Credits: 2
This course builds on the programming concepts learned in COMSC-150, covering object-oriented programming and introducing the Java programming language.
A. St. John
Prereq: COMSC-150 or placement test.
Advisory: Students may not take this course after Computer Science 151.
Notes: Half semester. The combination of COMSC-150 and COMSC-161 serves as an alternate prerequisite route for COMSC-205 Data Structures.

COMSC-201 Advanced Problem-Solving and Elementary Data Structures
Not Scheduled for This Year. Credits: 4
This course builds on the basic programming concepts learned in Computer Science 101. Emphasis is on developing the skills needed to write more sophisticated programs. This includes strategies to aid in assuring the correctness of programs through the use of assertions and unit testing as well as advanced Java features such as inheritance, polymorphism, and network programming. We will also introduce some widely used data structures such as vectors and linked lists. This course is programming-intensive.
Applies to requirement(s): Math Sciences
The department
Prereq: COMSC-101 with a grade of C or better. Coreq: COMSC-201L.
Notes: Students must select a lab with the same instructor as the lecture.

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
O. Erat
Prereq: COMSC-151 (with a grade of C or better) or COMSC-150 (with a grade of C or better) and either COMSC-121 or COMSC-161. Coreq: COMSC-205L.
Advisory: This course cannot be taken by students who have completed COMSC-201 or COMSC-211.

COMSC-205PY Data Structures (in Python)
Not Scheduled for This Year. Credits: 4
This course builds on the basic programming concepts learned in Computer Science 150 and Computer Science 121, 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.
Applies to requirement(s): Math Sciences
The department
Prereq: COMSC-150 (with grade of C or better) and COMSC-121; or COMSC-151 (with grade of C or better).
Notes: Students taking this course will need to take COMSC-122 Java Programming Language before continuing with COMSC-225.

COMSC-211 Advanced Data Structures
Not Scheduled for This Year. Credits: 4
Using Java. Solving problems with computers is accomplished by writing programs that operate on data to produce a desired result. The way data is organized and presented to the program can significantly affect its efficiency and simplicity and can sometimes determine whether or not a program can be written to solve the problem at all. This course presents ways of organizing data into ‘data structures’ and analyzes how structuring the data can improve program performance. This course is programming intensive.
Applies to requirement(s): Math Sciences
The department
Prereq: COMSC-201.

COMSC-215 Software Design
Not Scheduled for This Year. Credits: 4
Building large software systems introduces new challenges to software development. Appropriate design decisions early in the development of large software can make a major difference in developing software that is correct and maintainable. In this course, students will learn techniques and tools to help them address these problems and develop larger software projects, improving their skills in designing, writing, debugging, and testing software. Topics include design patterns, UML, designing for maintainability, software architecture, and designing concurrent and fault tolerant systems. Programming intensive.
Applies to requirement(s): Math Sciences
B. Lerner
Prereq: COMSC-201.

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, COMSC-205, or COMSC-205PY; 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
S. Biswal, Y. Su
Prereq: COMSC-205 (with a grade of C or better), or COMSC-205PY (with a grade of C or better).
Advisory: Students who have taken COMSC-215 may not take COMSC-225.
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

Prereq: COMSC-151 or COMSC-201.

Instructor permission required.

COMSC-243 Topic

COMSC-243EM Topic: ‘Embodied Interaction’

Not Scheduled for This Year. Credits: 4
This class will expose students to programming techniques used in computer-based interactive art including real-time graphics, data visualization, human-computer interaction, sensor networks, computer vision, and physical computing through analysis of existing computational art and synthesis of original works. The course will place particular emphasis on embodied interaction – interaction that uses sensors to react to the whole body. Weekly assignments and reading will serve to reinforce concepts from lectures, build technical skills, and develop a personal aesthetic.

Applies to requirement(s): Math Sciences

The department

Prereq: COMSC-226.

COMSC-243MS Topic: ‘Modeling and Simulation’

Not Scheduled for This Year. Credits: 4
This class will expose students to modeling and simulation of physical systems. Drawing on examples from a number of different disciplines, the course will cover modeling and analyzing a physical system, using models to predict behavior. Students will strengthen programming skills and learn additional computational skills necessary for simulation in areas such as population growth, disease spread, heat transfer, projectile motion.

Applies to requirement(s): Math Sciences

The department

Prereq: COMSC-151 or COMSC-201.

COMSC-243SW Topic: ‘Computing Systems Workshop’

Spring. 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

Instructor permission required.

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 PASCAL? This seminar 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 Universal Turing Machine, and computability.

Applies to requirement(s): Math Sciences

The department

Prereq: COMSC-201 or COMSC-205; MATH-232.

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

O. Erat

Prereq: COMSC-201 or COMSC-205 or COMSC-205PY; MATH-232.
COMSC-316 Developing Innovative Software
Spring. 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-215 or 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 either COMSC-211 or COMSC-225.

COMSC-331 Computer Graphics
Not Scheduled for This Year. Credits: 4
The creation of pictorial images using a computer. Topics include drawing of two- and three-dimensional scenes using OpenGL and other graphical environments; transformations of objects (translations, scalings, rotations, shearings) using homogeneous coordinates; creating perspective in three-dimensional drawing; algorithms for enhancing realism and visual effect; and ray tracing. Students will complete a number of graphics projects based on readings and class discussion. This course is programming intensive.
Applies to requirement(s): Math Sciences
Prereq: COMSC-205 or COMSC-221, and at least one of the following: MATH-203, MATH-211, or MATH-232.

COMSC-334 Artificial Intelligence
Fall. 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
Instructor permission required.
Prereq: COMSC-225 and MATH-232.
Advisory: Apply for permission using this form: https://forms.gle/nar9XSgTD0jpjZEL7

COMSC-335 Machine Learning
Fall and 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
Y. Su
Instructor permission required.
Prereq: A grade of C or better in one of COMSC-205 or COMSC-205PY or COMSC-211, 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. Use this form to apply for permission: https://forms.gle/gmPFFyokU1a64QB69

COMSC-341 Topics
COMSC-341CC Topics: 'Compiler Design'
Spring. 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
V. Barr
Instructor permission required.
Prereq: COMSC-221, COMSC-225, and COMSC-312.

COMSC-341CP Topics: 'Cyber-Physical Systems'
Not Scheduled for This Year. Credits: 4
Tired of mixing test tubes by hand, counting ant colonies, or transcribing for hours? Automation and instrumentation advance scientific research, freeing us from tasks that are dirty, dangerous or boring while improving precision and repeatability. Advances in mobile processor design make it easier to add computing-based automation to "dumb" devices. Student teams will create innovative tools for teaching and research, focusing on tools that advance teaching and research around the college and studying embedded computing topics including reliability, testing and qualification, signal processing, real-time systems, collaborative design, and learning rapid prototyping in the Makerspace.
Applies to requirement(s): Math Sciences
The department
Prereq: COMSC-221.
COMSC-341DC Topics: ‘Distributed Systems Engineering’
Not Scheduled for This Year. Credits: 4
How does Google respond to search queries so quickly? How does the power grid maintain stability when a tree falls on a wire? Distributed systems solve big problems by facilitating cooperation between independent agents towards a common goal. This course covers major principles of distributed systems: resource contention, concurrent action, scheduling, and communicating. Students will put theory into practice designing, implementing, and debugging distributed systems. This course is programming intensive.
Applies to requirement(s): Math Sciences
The department
Prereq: COMSC-205.

COMSC-341GP Topics: ‘Game Programming’
Fall. 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.
Advisory: Use this form: https://forms.gle/nar9XSGTDjpOBZEL7 to apply for permission.

COMSC-341NL Topics: ‘Natural Language Processing’
Not Scheduled for This Year. 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
The department
Prereq: COMSC-221 and COMSC-312.

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
The department
Prereq: COMSC-205 or COMSC-211.

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
The department
Prereq: COMSC-225.

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