COMPUTER SCIENCE

Overview and Contact Information
Computer science is an exciting field with applications across 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 on 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, developing video games and computer animations, 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
Valerie Barr, Chair
Wendy Queiros, Academic Department Coordinator
207 Clapp Laboratory
413-538-2420
https://www.mtholyoke.edu/acad/computerscience

Faculty
This area of study is administered by the Department of Computer Science:
Valerie Barr, Jean E. Sammet Professor of Computer Science
Lisa Ballesteros, Associate Professor of Computer Science, Teaching Fall Only
Audrey Lee St. John, Associate Professor of Computer Science, On Leave 2018-2019
Barbara Lerner, Associate Professor of Computer Science, Teaching Spring Only

Peter Klemperer, Assistant Professor of Computer Science
Heather Pon-Barry, Clare Boothe Luce Assistant Professor of Computer Science
Eitan Mendelowitz, Visiting Assistant Professor of Data Science
Daniel Sheldon, Five College Assistant Professor of Computer Science
Janet Slocum, Lab Instructor, Visiting Lecturer in Computer Science
Carolyn Anderson, Visiting Instructor in Computer Science; UMass Teaching Associate
Yuvraj Singla, Visiting Instructor in Computer Science; UMass Teaching Associate

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>Programming Paradigms and Software Development</td>
<td>4</td>
</tr>
<tr>
<td>COMSC-312</td>
<td>Algorithms</td>
<td>4</td>
</tr>
<tr>
<td>COMSC-322</td>
<td>Operating Systems</td>
<td>4</td>
</tr>
<tr>
<td>Three</td>
<td>additional computer science courses:</td>
<td>12</td>
</tr>
<tr>
<td>courses:</td>
<td>Two of these must be at the 300 level (8 credits)</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>The third may be at either the 200 level or 300</td>
<td></td>
</tr>
<tr>
<td></td>
<td>level (4 credits)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mathematics (4 credits)</td>
<td></td>
</tr>
<tr>
<td>MATH-232</td>
<td>Discrete Mathematics</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Total Credits</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

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:

<table>
<thead>
<tr>
<th>Code</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH-232</td>
<td>Discrete Mathematics</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Total Credits</td>
<td>20</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
Course Offerings

**COMSC-100 An Introduction to Computer Science**

*Fall. Credits: 4*

An introduction to basic computer science concepts with a focus on Python programming to manipulate images and sounds. Lectures will cover topics such as the origins of computing, computer architecture, artificial intelligence, and robotics. Programming exercises will use the Python language to do image and sound manipulation.

*Applies to requirement(s): Math Sciences

J. Slocum

*Notes: Students may not take this course after Computer Science 106 or 151.*

**COMSC-103 Networks**

*Not Scheduled for This Year. Credits: 4*

How do opinions, fads, and political movements spread through society? What makes food webs and financial markets robust? What are the technological, political, and economic forces that play in online communities? This course examines connections between the social, technological, and natural worlds through the lens of networks. Students will learn basics of graph theory and game theory and apply them to build mathematical models of processes that take place in networks.

*Applies to requirement(s): Math Sciences

The department

**COMSC-106 Fundamentals of Applied Computing**

*Not Scheduled for This Year. 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 or 151. Students may not take Computer Science 100 after taking 106, but may take 151.*

*Notes: Course does not count toward the Computer Science major or minor.*

**COMSC-109 iDesign Studio**

*Fall and Spring. 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.

*Course does not count toward the Computer Science major or minor.*

**COMSC-132 Engineering for Everyone**

*Spring. Credits: 4*

Engineers change the world we live in every day by developing and improving nearly every aspect of our lives. In this course, we will study the interaction of technology and society and how the engineering design process helps shape the world we live in. Engineering comprises many disciplines, but one common theme is the engineering design process: research, problem definition, feasibility, conceptualization, prototyping, and testing. In this class, students will learn the engineering design process through application to contemporary technological and societal issues put into practice with pitch presentations, design reviews, prototypes, and written reports.

*Applies to requirement(s): Math Sciences

P. Klemperer

*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-151 Introduction to Computational Problem Solving
COMSC-151AA Introduction to Computational Problem Solving: 'Algorithmic Arts'
Spring. Credits: 4
Introduction to the field of computer science with a computer art theme. 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
E. Mendelowitz
Coreq: COMSC-151AAL.
Notes: Additional seats will open for all students after first year students have registered.

COMSC-151AR Introduction to Computational Problem Solving: 'Artificial Intelligence'
Spring. Credits: 4
Introduction to the field of computer science with a theme of artificial intelligence. 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
V. Barr
Coreq: COMSC-151ARL.
Notes: Additional seats will open for all students after first year students have registered.

COMSC-151DS Introduction to Computational Problem Solving: 'Data Science'
Fall. Credits: 4
Introduction to the field of computer science with a data science theme. Introduces students to algorithms, basic data structures, and programming techniques. Also introduces the skills, techniques, and tools needed to collect, prepare, analyze, and visualize data to quantitatively ask and answer questions. Through readings, discussions, case studies, and projects, students will explore a breadth of subjects including programming for data manipulation, the presentation and representation of data, statistics and machine learning, "Big Data," and the ethics of working with data at scale.
Applies to requirement(s): Math Sciences
E. Mendelowitz
Coreq: COMSC-151DSL.

COMSC-151HC Introduction to Computational Problem Solving: 'Humanities Computing'
Fall. Credits: 4
Introduction to the field of computer science with a theme of humanities computing. 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
V. Barr
Coreq: COMSC-151HCL.

COMSC-151MD Introduction to Computational Problem Solving: 'Computers in Medical Technology'
Fall and Spring. Credits: 4
Introduction to the field of computer science with a theme of computing in medicine. 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
J. Slocum
Coreq: COMSC-151MDL.
Notes: Additional seats will open for all students after first year students have registered.

COMSC-201 Advanced Problem-Solving and Elementary Data Structures
Fall. 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
H. Pon-Barry, 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
Spring. Credits: 4
This course builds on the basic programming concepts learned in Computer Science 101, 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
V. Barr
Prereq: COMSC-101 or COMSC-151 with a grade of C or better. Coreq: COMSC-205L.
Advisory: This course cannot be taken by students who have completed COMSC-201 or COMSC-211.

COMSC-211 Advanced Data Structures
Fall and Spring. 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
V. Barr, E. Mendelowitz
Prereq: COMSC-201.
COMSC-215 Software Design  
_Spring._ 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, J. Slocum_  
_Prereq: COMSC-201. Coreq: COMSC-221L._

COMSC-225 Programming Paradigms and Software Development  
_Not Scheduled for This Year._ Credits: 4  
Computational approaches to problems follow an iterative design process, refining an abstract solution to a final implementation. This includes identifying or creating an abstract model for the problem, choosing appropriate data structures and algorithms or heuristics and working within the constraints of a particular programming paradigm or environment. In this course, students will put this design process into practice through programming projects, using additional tools beyond those in COMSC-151 and COMSC-205. This course is programming intensive.  
_Applies to requirement(s): Math Sciences_  
_The department_  
_Prereq: COMSC-205._

COMSC-226 Engineering Robotic Systems  
_Spring._ Credits: 4  
This intermediate-level course presents a hands-on introduction to robotics. Each participant will construct and modify a robot controlled by an Arduino-compatible 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. Participants will use the Makerspace facilities to fabricate and demonstrate their robots.  
_Applies to requirement(s): Math Sciences_  
_P. Klemperer_  
_Instructor permission required._  
_Prereq: COMSC-109 or COMSC-201._

COMSC-243 Topic  
COMSC-243EM Topic: 'Embodied Interaction'  
_Fall._ 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_  
_E. Mendelowitz_  
_Prereq: COMSC-211._

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 and MATH-232._

COMSC-312 Algorithms  
_Fall and Spring._ Credits: 4  
How does Mapquest 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 will explore graph algorithms, greedy algorithms, divide-and-conquer, dynamic programming, and network flow. We will learn to recognize when to apply each of these strategies as well as to evaluate the expected runtime costs of the algorithms we design.  
_Applies to requirement(s): Math Sciences_  
_B. Lerner, D. Sheldon_  
_Prereq: COMSC-211 and MATH-232._
COMSC-316 Software Practicum
*Not Scheduled for This Year. Credits: 4*
Tired of writing programs that nobody ever uses? Then, this is the course for you. Software enables enterprises to carry out previously tedious or impossible tasks, but many organizations lack the resources to develop needed software. You will apply your programming skills to develop and deliver software to meet the requirements of a client from the community. You will learn critical communication skills required to work with a client, work as a team with classmates, and experience the software lifecycle from requirements elicitation through delivery. You will synthesize many topics learned in courses as well as new technologies required to complete the project. Programming intensive.
*Applies to requirement(s): Math Sciences*
*The department*
*Instructor permission required.*
*Prereq: COMSC-215.*

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, file-handling systems, memory management, virtual memory, resource scheduling, multiprocessing, deadlocks, concurrent processes, protection, and design principles. Course emphasis: understanding the effects of operating system design on computer system performance.
*This course is programming intensive.*
*Applies to requirement(s): Math Sciences*
*B. Lerner, Y. Singla*
*Prereq: COMSC-211 and COMSC-221.*

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*
*Prereq: COMSC-211 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*
*D. Sheldon*
*Instructor permission required.*
*Prereq: COMSC-211, MATH-232, and a Calculus course (MATH-101, MATH-102, or MATH-203).*
*Advisory: Preference will be given to seniors in need of a final 300-level elective.*

COMSC-336 Intelligent Information Retrieval
*Not Scheduled for This Year. Credits: 4*
Introduces the basic concepts, methodologies, and research findings in information retrieval. Special topics include Web searching, cross-language retrieval, data mining, and data extraction. Completion of this course will provide the necessary foundation to work in today's business environment where competitive advantage is obtained by retrieving needed information.
*Applies to requirement(s): Math Sciences*
*L. Ballesteros*
*Prereq: COMSC-211.*

COMSC-341 Topics
COMSC-341DC Topics: 'Distributed Systems Engineering'
*Fall. 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*
*p. klemperer*
*Prereq: COMSC-201.*

COMSC-341NL Topics: 'Natural Language Processing'
*Fall. Credits: 4*
This seminar provides an introduction to natural language processing, the discipline of getting computers to understand human language. We will cover core ideas and algorithms relevant to both speech processing and text processing, with emphasis on applications in human-computer natural language interaction. Students will design and complete an open-ended final project.
*Applies to requirement(s): Math Sciences*
*H. Pon-Barry*
*Prereq: COMSC-211.*
COMSC-341SP Topics: 'Computer Security & Privacy'
Not Scheduled for This Year. Credits: 4
Data security is an everyday concern for people but also for high-profile targets like the United States Director of National Intelligence, the United States Federal Bureau of Investigation, Target and Home Depot. In the course we will study the security vulnerabilities that make attacks possible and actions that can be taken to mitigate them. This course will introduce a variety of security topics: identifying software security vulnerabilities, malicious software, cryptography, authentication, access-control, networking, risk-analysis, usability, and ethics. Students will complete readings, in-class discussions, and hands-on programming activities.
Applies to requirement(s): Math Sciences
P. Klemperer
Prereq: COMSC-211.

COMSC-343 Programming Language Design and Implementation
Fall. 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
C. Anderson
Prereq: COMSC-211.

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