Department of Computer Science

Alexandru Nicolau, Department Chair
Rina Dechter, Department Vice Chair
3019 Donald Bren Hall
949-824-1546
http://www.cs.uci.edu/

Overview

With 40 full-time faculty members, 200+ graduate students, and more than 800 undergraduates, we provide a world-class research environment spanning not only the core areas of computer science — including computer architecture, system software, networking and distributed computing, data and information systems, the theory of computation, artificial intelligence, and computer graphics — but also highly interdisciplinary programs, such as biomedical informatics, data mining, security and privacy, and ubiquitous computing.

The diverse research interests of our faculty are reflected directly in our educational programs. Computer Science faculty teach most of the undergraduate and graduate courses for the degree programs in both Computer Science and Information & Computer Science. We jointly offer with our colleagues in The Henry Samueli School of Engineering an undergraduate degree in Computer Science and Engineering, as well as the graduate program in Networked Systems. We also have a major in Computer Game Science, jointly offered with the Department of Informatics.

Our department collaborates with many other institutions in the United States and abroad, and its doors are always open to a multitude of visitors and collaborators from all corners of the globe.

On This Page:

• Undergraduate Major in Computer Science
  • Admissions
  • Requirements for the B.S. Degree in Computer Science
  • Sample Program of Study
  • Undergraduate Major in Computer Science and Engineering

Undergraduate Major in Computer Science

The Computer Science major emphasizes the principles of computing that underlie our modern world, and provides a strong foundational education to prepare students for the broad spectrum of careers in computing. This major can serve as preparation for either graduate study or a career in industry. Students receive a solid background in low-level architecture and systems; middle-level infrastructure, algorithms, and mathematical foundations. This is a highly flexible degree that allows students to explore a broad range of topics in modern computing. In order to achieve some focus in their upper-division studies, students are required to satisfy the requirements for one of the eight specializations described below. Additional electives can be used to satisfy a second specialization or obtain a broader understanding of the field.

Algorithms. This specialization focuses on fundamental computational techniques, including their analysis and applications to topics in computer vision, computer games, graphics, artificial intelligence, and information retrieval. Topics include data structures, graph and network algorithms, computational geometry, probabilistic algorithms, complexity theory, and cryptography.

Architecture and Embedded Systems. This specialization integrates principles of embedded systems, software, hardware, computer architecture, distributed systems and networks, and prepares students to design and create efficient hardware/software architectures for emerging application areas. Students in this specialization will build upon a strong foundation in software and hardware and learn how to design networked embedded systems, and efficient computer architectures for a diverse set of application domains such as gaming, visualization, search, databases, transaction processing, data mining, and high-performance and scientific computing.

Bioinformatics. This specialization introduces students to the interdisciplinary intersection of biology and medicine with computer science and information technology. Students who complete the specialization will understand biomedical computing problems from the computer science perspectives, and be able to design and develop software that solves computational problems in biology and medicine.

General Computer Science. This specialization allows students to acquire a well-rounded knowledge of computer science that may be tailored to their individual interests. Students choose 11 upper-division computer science courses, including two project courses. This specialization will appeal to those who are interested in a broad education in computer science, or who wish to create their own unique specialization not found in the current list of (other) specializations under this major.

Information. This specialization is intended to prepare students for working with and developing a wide variety of modern data and information systems. Topics covered by this concentration include database management, information retrieval, Web search, data mining, and data-intensive computing.
Intelligent Systems. This specialization will introduce students to the principles underlying intelligent systems, including topics such as representing human knowledge, building automated reasoning systems, developing intelligent search techniques, and designing algorithms that adapt and learn from data. Students in this specialization will use these principles to solve problems across a variety of applications such as computer vision, information retrieval, data mining, automated recommender systems, bioinformatics, as well as individually designed projects.

Networked Systems. This specialization focuses on Internet architecture, Internet applications, and network security. It also encourages students to learn about operating systems, databases, search, programming, embedded systems, and performance.

Systems and Software. This specialization deals with principles and design of systems and software. It emphasizes the interaction between software and the computing infrastructure on which it runs and the performance impact of design decisions. Core topics include the hardware/software interface, languages and compilers, operating systems, parallel and distributed computing. Elective topics include networking, security, graphics, and databases.

Visual Computing. This specialization encompasses the digital capture, processing, synthesis and display of visual data such as images and video. This specialization includes computer vision, image processing, and graphics, and covers such topics as the representation of 3D objects, visual recognition of objects and people, interactive and photo-realistic image rendering, and physics and perception of light and color.

The Department also offers a joint undergraduate degree in Computer Science and Engineering, in conjunction with The Henry Samueli School of Engineering; information is available in the Interdisciplinary Studies section of the Catalogue.

Admissions
Freshman Applicants: See the Undergraduate Admissions section.

Transfer Applicants:
Junior-level applicants who satisfactorily complete course requirements will be given preference for admission. Applicants must satisfy the following requirements:

1. Complete one year of approved college-level math, preferably courses in calculus equivalent to UCI's MATH 2A - MATH 2B; if not available, one year of coursework equivalent to other major-related math courses is acceptable.
2. Completion of one year of transferable computer science courses with at least one course involving concepts such as those found in Java, Python, C++, or other object-oriented or high-level programming language.

NOTE: The introductory sequence in ICS has moved to Python. The Bren School of ICS strongly encourages all participants to become familiar with this programming language prior to matriculation. Additional computer science courses beyond the two required are strongly recommended, particularly those that align with the major(s) of interest. Java is used extensively in the curriculum; therefore, transfer students should plan to learn it by studying on their own or by completing a Java-related programming course prior to their first quarter at UCI.

Additional courses beyond those required for admission must be taken to fulfill the lower-division degree requirements, as many are prerequisites for upper-division courses. For some transfer students, this may mean that it will take longer than two years to complete their degree.

Requirements for the B.S. Degree in Computer Science
All students must meet the University Requirements.

Major Requirements

Lower-division

<table>
<thead>
<tr>
<th>A. Core</th>
<th>B. Complete:</th>
</tr>
</thead>
<tbody>
<tr>
<td>I&amp;C SCI 31</td>
<td>MATH 2A- 2B</td>
</tr>
<tr>
<td>I&amp;C SCI 32</td>
<td>Single-Variable Calculus</td>
</tr>
<tr>
<td>I&amp;C SCI 33</td>
<td>and Single-Variable Calculus</td>
</tr>
<tr>
<td>I&amp;C SCI 51</td>
<td>I&amp;C SCI 6B</td>
</tr>
<tr>
<td>I&amp;C SCI 51</td>
<td>Boolean Algebra and Logic</td>
</tr>
<tr>
<td>I&amp;C SCI 90</td>
<td>I&amp;C SCI 6D</td>
</tr>
<tr>
<td>IN4MATX 43</td>
<td>Discrete Mathematics for Computer Science</td>
</tr>
<tr>
<td>I&amp;C SCI 53L</td>
<td>Principles in System Design Library</td>
</tr>
<tr>
<td>I&amp;C SCI 45C</td>
<td>Principles in System Design</td>
</tr>
<tr>
<td>I&amp;C SCI 46</td>
<td>Data Structure Implementation and Analysis</td>
</tr>
<tr>
<td>I&amp;C SCI 53</td>
<td>Introductory Computer Organization</td>
</tr>
<tr>
<td>I&amp;C SCI 53</td>
<td>Programming in C/C++ as a Second Language</td>
</tr>
<tr>
<td>I&amp;C SCI 50</td>
<td>New Students Seminar</td>
</tr>
<tr>
<td>I&amp;C SCI 43</td>
<td>Introduction to Software Engineering</td>
</tr>
</tbody>
</table>

Introduction to Programming
Programming with Software Libraries
Intermediate Programming
Data Structure Implementation and Analysis
Principles in System Design
Principles in System Design Library
Single-Variable Calculus
Boolean Algebra and Logic
Discrete Mathematics for Computer Science
I&C SCI 6N
or MATH 3A
STATS 67
C. Two courses approved for General Education category II except those offered by CSE, Economics, ICS, or Mathematics. University Studies courses can be used with the approval of the CS Vice Chair for Undergraduate Studies.

Upper-division

A. Core
COMPSCI 161
I&C SCI 139W

B. Electives: Select eleven of the following:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN4MATX 102</td>
<td>Concepts of Programming Language II</td>
</tr>
<tr>
<td>IN4MATX 113</td>
<td>Requirements Analysis and Engineering</td>
</tr>
<tr>
<td>IN4MATX 115</td>
<td>Software Testing, Analysis, and Quality Assurance</td>
</tr>
<tr>
<td>IN4MATX 121</td>
<td>Software Design: Applications</td>
</tr>
<tr>
<td>IN4MATX 122</td>
<td>Software Design: Structure and Implementation</td>
</tr>
<tr>
<td>IN4MATX 123</td>
<td>Software Architecture</td>
</tr>
<tr>
<td>IN4MATX 131</td>
<td>Human Computer Interaction</td>
</tr>
<tr>
<td>IN4MATX 133</td>
<td>User Interaction Software</td>
</tr>
<tr>
<td>IN4MATX 134</td>
<td>Project in User Interaction Software</td>
</tr>
<tr>
<td>I&amp;C SCI 160</td>
<td>Graphics Processors and Game Platforms</td>
</tr>
<tr>
<td>I&amp;C SCI 161</td>
<td>Game Engine Lab</td>
</tr>
<tr>
<td>I&amp;C SCI 162</td>
<td>Modeling and World Building</td>
</tr>
</tbody>
</table>

C. The upper-division electives must satisfy the following criteria:

1. Specialization. Select and satisfy the requirements for one of the specializations below:

Algorithms

COMPSCI 162
COMPSCI 163
COMPSCI 164
COMPSCI 165
COMPSCI 167
COMPSCI 169

Architecture and Embedded Systems: four courses from the following list:

COMPSCI 144
COMPSCI 145A-145B
COMPSCI 151
COMPSCI 152
COMPSCI 153
COMPSCI 154

Recommended electives:

COMPSCI 131
COMPSCI 142A
COMPSCI 143A

Bioinformatics: three courses from the following list:

COMPSCI 184A
and two courses from:

COMPSCI 172B
COMPSCI 184B
COMPSCI 184C
COMPSCI 189
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 121</td>
<td>Information Retrieval</td>
</tr>
<tr>
<td>COMPSCI 122A</td>
<td>Introduction to Data Management</td>
</tr>
<tr>
<td>COMPSCI 178</td>
<td>Machine Learning and Data-Mining</td>
</tr>
<tr>
<td></td>
<td>and four courses from:</td>
</tr>
<tr>
<td>I&amp;C SCI 45J</td>
<td>Programming in Java as a Second Language</td>
</tr>
<tr>
<td>COMPSCI 122B</td>
<td>Project in Databases and Web Applications</td>
</tr>
<tr>
<td>COMPSCI 125</td>
<td>Next Generation Search Systems</td>
</tr>
<tr>
<td>COMPSCI 132</td>
<td>Computer Networks</td>
</tr>
<tr>
<td>COMPSCI 134</td>
<td>Computer and Network Security</td>
</tr>
<tr>
<td>COMPSCI 141</td>
<td>Concepts in Programming Languages I</td>
</tr>
<tr>
<td>COMPSCI 142A</td>
<td>Compilers and Interpreters</td>
</tr>
<tr>
<td>COMPSCI 143A</td>
<td>Principles of Operating Systems</td>
</tr>
<tr>
<td>COMPSCI 163</td>
<td>Graph Algorithms</td>
</tr>
<tr>
<td>COMPSCI 165</td>
<td>Project In Algorithms And Data Structures</td>
</tr>
<tr>
<td>COMPSCI 167</td>
<td>Introduction to Applied Cryptography</td>
</tr>
<tr>
<td>COMPSCI 179</td>
<td>Algorithms for Probabilistic and Deterministic Graphical Models</td>
</tr>
<tr>
<td></td>
<td>at least one of which must be:</td>
</tr>
<tr>
<td>COMPSCI 122B</td>
<td>Project in Databases and Web Applications</td>
</tr>
<tr>
<td>or COMPSCI 125</td>
<td>Next Generation Search Systems</td>
</tr>
<tr>
<td>or COMPSCI 179</td>
<td></td>
</tr>
<tr>
<td><strong>Intelligent Systems</strong></td>
<td></td>
</tr>
<tr>
<td>COMPSCI 171</td>
<td>Introduction to Artificial Intelligence</td>
</tr>
<tr>
<td>COMPSCI 175</td>
<td>Project in Artificial Intelligence</td>
</tr>
<tr>
<td>COMPSCI 178</td>
<td>Machine Learning and Data-Mining</td>
</tr>
<tr>
<td></td>
<td>and at least one course from:</td>
</tr>
<tr>
<td>COMPSCI 177</td>
<td>Applications of Probability in Computer Science</td>
</tr>
<tr>
<td>or COMPSCI 179</td>
<td>Algorithms for Probabilistic and Deterministic Graphical Models</td>
</tr>
<tr>
<td></td>
<td>and at least one course from:</td>
</tr>
<tr>
<td>COMPSCI 162</td>
<td>Formal Languages and Automata</td>
</tr>
<tr>
<td>COMPSCI 163</td>
<td>Graph Algorithms</td>
</tr>
<tr>
<td>COMPSCI 164</td>
<td>Computational Geometry and Geometric Modeling</td>
</tr>
<tr>
<td>COMPSCI 168</td>
<td>Network Optimization</td>
</tr>
<tr>
<td>COMPSCI 169</td>
<td>Introduction to Optimization</td>
</tr>
<tr>
<td></td>
<td>and at least one course from:</td>
</tr>
<tr>
<td>COMPSCI 116</td>
<td>Computational Photography and Vision</td>
</tr>
<tr>
<td>COMPSCI 121</td>
<td>Information Retrieval</td>
</tr>
<tr>
<td>COMPSCI 125</td>
<td>Next Generation Search Systems</td>
</tr>
<tr>
<td>COMPSCI 174</td>
<td>Bioinformatics</td>
</tr>
<tr>
<td>COMPSCI 184B</td>
<td>Probabilistic Modeling of Biological Data</td>
</tr>
<tr>
<td><strong>Networked Systems</strong></td>
<td></td>
</tr>
<tr>
<td>COMPSCI 132</td>
<td>Computer Networks</td>
</tr>
<tr>
<td>COMPSCI 133</td>
<td>Advanced Computer Networks</td>
</tr>
<tr>
<td>COMPSCI 134</td>
<td>Computer and Network Security</td>
</tr>
<tr>
<td>COMPSCI 143A</td>
<td>Principles of Operating Systems</td>
</tr>
<tr>
<td>Recommended electives:</td>
<td></td>
</tr>
<tr>
<td>One course from:</td>
<td></td>
</tr>
<tr>
<td>COMPSCI 122B</td>
<td>Project in Databases and Web Applications</td>
</tr>
<tr>
<td>COMPSCI 143B</td>
<td>Project in Operating System Organization</td>
</tr>
<tr>
<td>Two courses from:</td>
<td></td>
</tr>
<tr>
<td>COMPSCI 122A</td>
<td>Introduction to Data Management</td>
</tr>
<tr>
<td>COMPSCI 131</td>
<td>Parallel and Distributed Computing</td>
</tr>
<tr>
<td>COMPSCI 137</td>
<td>Internet Applications Engineering</td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Name</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>COMPSCI 167</td>
<td>Introduction to Applied Cryptography</td>
</tr>
<tr>
<td>COMPSCI 145A</td>
<td>Embedded Software</td>
</tr>
<tr>
<td>COMPSCI 163</td>
<td>Graph Algorithms</td>
</tr>
<tr>
<td>COMPSCI 169</td>
<td>Introduction to Optimization</td>
</tr>
</tbody>
</table>

**Systems and Software: three courses from the following list:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 131</td>
<td>Parallel and Distributed Computing</td>
</tr>
<tr>
<td>COMPSCI 141</td>
<td>Concepts in Programming Languages I</td>
</tr>
<tr>
<td>COMPSCI 142A</td>
<td>Compilers and Interpreters</td>
</tr>
<tr>
<td>COMPSCI 142B</td>
<td>Language Processor Construction</td>
</tr>
<tr>
<td>COMPSCI 143A</td>
<td>Principles of Operating Systems</td>
</tr>
<tr>
<td>COMPSCI 143B</td>
<td>Project in Operating System Organization</td>
</tr>
</tbody>
</table>

**Recommended electives:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 132</td>
<td>Computer Networks</td>
</tr>
<tr>
<td>COMPSCI 134</td>
<td>Computer and Network Security</td>
</tr>
<tr>
<td>COMPSCI 144</td>
<td>High-performance Computers and Program Optimization</td>
</tr>
<tr>
<td>COMPSCI 152</td>
<td>Computer Systems Architecture</td>
</tr>
</tbody>
</table>

**Visual Computing: four courses from the following list:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 111</td>
<td>Digital Image Processing</td>
</tr>
<tr>
<td>COMPSCI 112</td>
<td>Computer Graphics</td>
</tr>
<tr>
<td>COMPSCI 114</td>
<td>Projects in Advanced 3D Computer Graphics</td>
</tr>
<tr>
<td>COMPSCI 116</td>
<td>Computational Photography and Vision</td>
</tr>
<tr>
<td>COMPSCI 117</td>
<td>Project in Computer Vision</td>
</tr>
<tr>
<td>I&amp;C SCI 162</td>
<td>Modeling and World Building</td>
</tr>
</tbody>
</table>

**General CS track**

Any 11 upper-division COMPSCI courses

2. **Project Courses. Choose at least two project courses from the following list:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 113</td>
<td>Computer Game Development</td>
</tr>
<tr>
<td>COMPSCI 114</td>
<td>Projects in Advanced 3D Computer Graphics</td>
</tr>
<tr>
<td>COMPSCI 117</td>
<td>Project in Computer Vision</td>
</tr>
<tr>
<td>COMPSCI 122B</td>
<td>Project in Databases and Web Applications</td>
</tr>
<tr>
<td>COMPSCI 122C</td>
<td>Principles of Data Management</td>
</tr>
<tr>
<td>COMPSCI 133</td>
<td>Advanced Computer Networks</td>
</tr>
<tr>
<td>COMPSCI 142B</td>
<td>Language Processor Construction</td>
</tr>
<tr>
<td>COMPSCI 143B</td>
<td>Project in Operating System Organization</td>
</tr>
<tr>
<td>COMPSCI 145A-145B</td>
<td>Embedded Software and Embedded Software Laboratory</td>
</tr>
<tr>
<td>COMPSCI 153</td>
<td>Logic Design Laboratory</td>
</tr>
<tr>
<td>COMPSCI 154</td>
<td>Computer Design Laboratory</td>
</tr>
<tr>
<td>COMPSCI 165</td>
<td>Project In Algorithms And Data Structures</td>
</tr>
<tr>
<td>COMPSCI 175</td>
<td>Project in Artificial Intelligence</td>
</tr>
<tr>
<td>COMPSCI 189</td>
<td>Project in Bioinformatics</td>
</tr>
<tr>
<td>IN4MATX 134</td>
<td>Project in User Interaction Software</td>
</tr>
</tbody>
</table>

3. **Electives. The remainder of the upper-division electives, in conjunction with the specialization courses and project courses must equal a total of 11 courses. Choose the remainder of the electives from the following list:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN4MATX 102</td>
<td>Concepts of Programming Language II</td>
</tr>
<tr>
<td>IN4MATX 113</td>
<td>Requirements Analysis and Engineering</td>
</tr>
<tr>
<td>IN4MATX 115</td>
<td>Software Testing, Analysis, and Quality Assurance</td>
</tr>
<tr>
<td>IN4MATX 121</td>
<td>Software Design: Applications</td>
</tr>
<tr>
<td>IN4MATX 122</td>
<td>Software Design: Structure and Implementation</td>
</tr>
<tr>
<td>IN4MATX 123</td>
<td>Software Architecture</td>
</tr>
<tr>
<td>IN4MATX 131</td>
<td>Human Computer Interaction</td>
</tr>
<tr>
<td>IN4MATX 133</td>
<td>User Interaction Software</td>
</tr>
</tbody>
</table>
### Sample Program of Study — Computer Science

#### Freshman

<table>
<thead>
<tr>
<th>Fall</th>
<th>Winter</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>I&amp;C SCI 31</td>
<td>I&amp;C SCI 32</td>
<td>I&amp;C SCI 33</td>
</tr>
<tr>
<td>MATH 2A</td>
<td>MATH 2B</td>
<td>IN4MATX 43</td>
</tr>
<tr>
<td>WRITING 39A</td>
<td>WRITING 39B</td>
<td>I&amp;C SCI 6B</td>
</tr>
<tr>
<td>I&amp;C SCI 90</td>
<td>GE III</td>
<td>WRITING 39C</td>
</tr>
</tbody>
</table>

#### Sophomore

<table>
<thead>
<tr>
<th>Fall</th>
<th>Winter</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>I&amp;C SCI 51</td>
<td>I&amp;C SCI 46</td>
<td>CS Spec/Elective</td>
</tr>
<tr>
<td>I&amp;C SCI 6D</td>
<td>I&amp;C SCI 53</td>
<td>STATS 67</td>
</tr>
<tr>
<td>I&amp;C SCI 45C</td>
<td>I&amp;C SCI 53L</td>
<td>GE III</td>
</tr>
<tr>
<td>I&amp;C SCI 6N</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Junior

<table>
<thead>
<tr>
<th>Fall</th>
<th>Winter</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 161</td>
<td>CS Spec./Elective</td>
<td>CS Spec./Elective</td>
</tr>
<tr>
<td>Science Elective</td>
<td>CS Spec./Elective</td>
<td>CS Spec./Elective</td>
</tr>
<tr>
<td>GE III</td>
<td>I&amp;C SCI 139W</td>
<td>GE VI</td>
</tr>
<tr>
<td>GE VII</td>
<td>GE VIII</td>
<td></td>
</tr>
</tbody>
</table>

#### Senior

<table>
<thead>
<tr>
<th>Fall</th>
<th>Winter</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS Spec./Elective</td>
<td>CS Spec./Elective</td>
<td>CS Spec./Elective</td>
</tr>
<tr>
<td>CS Spec./Elective</td>
<td>CS Spec./Elective</td>
<td>CS Spec./Elective</td>
</tr>
<tr>
<td>GE IV</td>
<td>GE IV</td>
<td></td>
</tr>
</tbody>
</table>

### NOTES:

1. Students are advised that this sample program lists the minimum requirements; it is possible that students may have to take additional courses to prepare for required courses.

2. The lower-division writing requirement must be completed by the end of the seventh quarter at UCI.

3. This is only a sample plan. Course offerings may be moved due to unforeseen circumstances. It is strongly recommended that students meet with an academic advisor to create an academic plan tailored to meet their specific areas of interest.

### Undergraduate Major in Computer Science and Engineering (CSE)

This program is administered jointly by the Department of Computer Science in the Bren School of ICS, and the Department of Electrical Engineering and Computer Science (EECS) in The Henry Samueli School of Engineering. For information, see the Interdisciplinary Studies section of the Catalogue.

### Requirements for the B.S. Degree in Computer Science and Engineering

All students must meet the University Requirements.

Major Requirements: See the Interdisciplinary Studies section.

### Minor in Bioinformatics

The minor provides a focused study of bioinformatics to supplement a student’s major program of study and prepares students for a profession, career, or academic pursuit in which biomedical computing is an integral part but is not the primary focus. The Bioinformatics minor contributes to students’ competence in computing applied to biomedical problems and data, as well as exposing them to the fundamentals of the life sciences from a computer science perspective. The minor allows students sufficient flexibility to pursue courses that complement their major field or address specific interests.

Students who complete the minor requirements will be able to do the following: synthesize computer science, quantitative methods, and biological science; understand the synergistic set of reciprocal influences between life and computational sciences and technologies; discuss biomedical computing problems and corresponding computer science perspectives; and employ principles, methods, and technologies fundamental to biomedical computing.

### Requirements

Select one of the following:

- I&C SCI 21- 22 Introduction to Computer Science I
  and Introduction to Computer Science II
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>I&amp;C SCI 31/CSE 41</td>
<td>Introduction to Programming</td>
</tr>
<tr>
<td>I&amp;C SCI 32/CSE 42</td>
<td>Programming with Software Libraries</td>
</tr>
<tr>
<td>I&amp;C SCI 33/CSE 43</td>
<td>Intermediate Programming</td>
</tr>
<tr>
<td>BIO SCI 93</td>
<td>From DNA to Organisms</td>
</tr>
<tr>
<td>COMPSCI 183/BIO SCI M123</td>
<td>Introduction to Computational Biology</td>
</tr>
<tr>
<td>COMPSCI 184A</td>
<td>Representations and Algorithms for Molecular Biology</td>
</tr>
<tr>
<td>COMPSCI 184B</td>
<td>Probabilistic Modeling of Biological Data</td>
</tr>
<tr>
<td>or COMPSCI 184C</td>
<td>Computational Systems Biology</td>
</tr>
<tr>
<td>COMPSCI 189</td>
<td>Project in Bioinformatics</td>
</tr>
</tbody>
</table>

NOTE: A maximum of two courses may be taken Pass/Not Pass toward a minor. Bren School majors should refer to the Majors/Minors Restrictions catalog section before attempting to minor in Bioinformatics. Students who are considering a major in Computer Science or Computer Science and Engineering must complete the major requirements for a letter grade.

On This Page:
- Master of Science in Computer Science
- Doctor of Philosophy in Computer Science
  - Research Project for the Ph.D. Degree
  - Advancement to Candidacy Examination
  - Doctoral Dissertation Topic Defense
  - Doctoral Dissertation and Final Examination
- Graduate Program in Mathematical and Computational Biology

### Graduate Program in Computer Science

The field of Computer Science is concerned with the design, analysis, and implementation of computer systems as well as the use of computation as it is applied to virtually every field of study and use in the everyday world. Computer systems can range in scope from small embedded systems to the Internet as a whole. Research in computer science involves mathematical analysis, empirical experimentation, and the implementation of prototype systems. Core research areas include artificial intelligence and machine learning, bioinformatics, computer architecture, embedded systems, graphics and visual computing, databases and information management, multimedia, networked and distributed systems, programming languages and compilers, security and cryptography, design and analysis of algorithms, scientific computing, and ubiquitous computing.

The M.S. and Ph.D. degrees in Computer Science (CS) are broad and flexible programs which offer students opportunities for graduate study in the full spectrum of intellectual activity in computer science.

### Master of Science in Computer Science

The course requirements for the M.S. are identical to those of the Ph.D., although completion plans differ. For general information about admissions, the M.S. completion plan options, visit the Bren School of ICS Graduate Programs section of the Catalogue.

### Doctor of Philosophy in Computer Science

#### Required Courses

Each student must complete at least 47 units of course work with an average GPA of at least 3.5 for Ph.D. students and 3.0 for M.S. students. In addition, students must receive at least a B in each course counted toward filling these requirements. The set of core and elective courses chosen by a student must be approved by the student’s research advisor before advancement to candidacy. Faculty associated with each research area will provide suggested curricula for that area to guide students in their selection of courses. These curricula will also help Ph.D. students to prepare for their candidacy examination (see below) which must be taken in a specific research area.

Students must complete three quarters of COMPSCI 200S, four core courses, and seven elective courses. The course requirements are as follows:

Students must select four areas from the list of seven areas given below. From each area, they must select at least one of the courses listed for that area.

#### Data Structures and Algorithms

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 260</td>
<td>Fundamentals of the Design and Analysis of Algorithms</td>
</tr>
<tr>
<td>COMPSCI 261</td>
<td>Data Structures</td>
</tr>
<tr>
<td>COMPSCI 263</td>
<td>Analysis of Algorithms</td>
</tr>
</tbody>
</table>

#### Architecture/Embedded Systems
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 250A</td>
<td>Computer Systems Architecture</td>
</tr>
<tr>
<td>COMPSCI 244</td>
<td>Introduction to Embedded and Ubiquitous Systems</td>
</tr>
<tr>
<td><strong>System Software</strong></td>
<td></td>
</tr>
<tr>
<td>COMPSCI 241</td>
<td>Advanced Compiler Construction</td>
</tr>
<tr>
<td>COMPSCI 243</td>
<td>High-Performance Architectures and Their Compilers</td>
</tr>
<tr>
<td>COMPSCI 230</td>
<td>Distributed Computer Systems</td>
</tr>
<tr>
<td><strong>Artificial Intelligence</strong></td>
<td></td>
</tr>
<tr>
<td>COMPSCI 271</td>
<td>Introduction to Artificial Intelligence</td>
</tr>
<tr>
<td>COMPSCI 273A</td>
<td>Machine Learning</td>
</tr>
<tr>
<td><strong>Networks/Multimedia</strong></td>
<td></td>
</tr>
<tr>
<td>COMPSCI 232</td>
<td>Computer and Communication Networks</td>
</tr>
<tr>
<td>COMPSCI 203</td>
<td>Network and Distributed Systems Security</td>
</tr>
<tr>
<td>COMPSCI 212</td>
<td>Multimedia Systems and Applications</td>
</tr>
<tr>
<td><strong>Database Systems</strong></td>
<td></td>
</tr>
<tr>
<td>COMPSCI 222</td>
<td>Principles of Data Management</td>
</tr>
<tr>
<td>COMPSCI 223</td>
<td>Transaction Processing and Distributed Data Management</td>
</tr>
<tr>
<td><strong>Scientific and Visual Computing</strong></td>
<td></td>
</tr>
<tr>
<td>COMPSCI 206</td>
<td>Principles of Scientific Computing</td>
</tr>
<tr>
<td>COMPSCI 211A</td>
<td>Visual Computing</td>
</tr>
</tbody>
</table>

Seven elective courses from any set of CS, Informatics, or Statistics courses, including the above core courses, but excluding COMPSCI 290, COMPSCI 298, COMPSCI 299, or any course with a suffix of “S.”

Two of these courses can be graduate courses offered by a department outside of ICS, with written consent of the advisor (M.S. students must obtain written consent from the Computer Science Vice Chair for Graduate Studies).

Two of the courses can be undergraduate courses from the following list:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 111</td>
<td>Digital Image Processing</td>
</tr>
<tr>
<td>COMPSCI 112</td>
<td>Computer Graphics</td>
</tr>
<tr>
<td>COMPSCI 122A</td>
<td>Introduction to Data Management</td>
</tr>
<tr>
<td>COMPSCI 132</td>
<td>Computer Networks</td>
</tr>
<tr>
<td>COMPSCI 142A</td>
<td>Compilers and Interpreters</td>
</tr>
<tr>
<td>COMPSCI 143A</td>
<td>Principles of Operating Systems</td>
</tr>
<tr>
<td>COMPSCI 152</td>
<td>Computer Systems Architecture</td>
</tr>
<tr>
<td>COMPSCI 161</td>
<td>Design and Analysis of Algorithms</td>
</tr>
<tr>
<td>COMPSCI 171</td>
<td>Introduction to Artificial Intelligence</td>
</tr>
<tr>
<td>COMPSCI 178</td>
<td>Machine Learning and Data-Mining</td>
</tr>
<tr>
<td>I&amp;C SCI 161</td>
<td>Game Engine Lab</td>
</tr>
<tr>
<td>I&amp;C SCI 162</td>
<td>Modeling and World Building</td>
</tr>
<tr>
<td>I&amp;C SCI 163</td>
<td>Mobile and Ubiquitous Games</td>
</tr>
<tr>
<td>I&amp;C SCI 166</td>
<td>Game Design</td>
</tr>
</tbody>
</table>

Students may not retake courses they have used toward an undergraduate degree and receive credit toward the graduate requirements.

No more than two undergraduate courses or COMPSCI 295 may be taken to satisfy elective course requirements.

Ph.D. students are required to serve as teaching assistants for at least two quarters.

**Research Project for the Ph.D. Degree**

Doctoral students must find a faculty advisor and successfully complete a research project with that faculty member by the end of their second year. In coordination with this project the student must also take at least one independent studies course (COMPSCI 299) with their faculty advisor. The objective of the research project is to demonstrate early in the program the student’s ability to carry out basic research in computer science.

Finally, the student must present the outcome of the research in a technical report, which must be approved by the advisor. The project may or may not be a stepping-stone toward a dissertation, and must be completed by the end of the second year, and prior to advancement to candidacy.
Advancement to Candidacy Examination
The objective of the candidacy examination is to demonstrate in-depth knowledge of an area of computer science and readiness to carry out independent research at the doctoral level in that area. The student must complete all pre-candidacy course requirements and the research project prior to advancing to candidacy. All requirements for candidacy including the candidacy examination must be completed by the end of the third year (or, for students entering the program with an M.S. in Computer Science, by the end of the second year). If the student does not pass on the first trial, the student will be allowed until the end of the first quarter of the fourth year to advance to candidacy. Consult the ICS Graduate Office for policies regarding committee membership. The format is an oral examination during which the student is tested on knowledge relevant to the chosen area of specialization. Each area is defined by a set of topics and reading list, which are maintained by the Computer Science Department faculty. New areas or changes to existing areas must be approved by a majority vote of the CS faculty in accordance with the Department’s bylaws. The current areas include the following: Algorithms and Data Structures; Computer Architecture and Embedded Systems; Database Systems and Multimedia; Computer Networks; Distributed Systems; Artificial Intelligence and Machine Learning; Informatics in Biology and Medicine; Computer Graphics and Visual Computing; Cryptography and Computer Security; Computational Neuroscience; Scientific Computing; Systems Software.

The examination is graded pass or fail. In order to pass, the Candidacy Committee must unanimously approve the final outcome. In the case of a fail, the examination may be retaken once. Students who fail on the second try will be recommended for disqualification from the doctoral program.

Doctoral Dissertation Topic Defense
The student must produce a substantial written document representing the dissertation plan. This must include the proposed dissertation abstract, a dissertation outline, and a detailed plan for completing the work. A dissertation defense committee is formed in accordance with UCI Senate regulations. The dissertation committee must unanimously approve the student’s proposal. At the discretion of the student’s advisor, the student may be required to give an oral presentation of the proposed plan to the committee. This must be completed by the end of the fourth year. It is expected that this will be done at least a year prior to the final examination and before most of the dissertation research and writing are undertaken. The idea is for students to demonstrate that they have a clear plan for carrying out the research for their dissertation. It also gives the student an understanding of what will be expected for final approval of the dissertation.

Doctoral Dissertation and Final Examination
Ph.D. students are required to complete a Ph.D. dissertation in accordance with Academic Senate regulations. In addition, they must pass an oral dissertation defense which consists of a public seminar presenting results followed by a private examination by the doctoral committee and other interested members of the Computer Science Department faculty.

Students entering the Ph.D. program with an M.S. in Computer Science must advance to candidacy within two years. All others must advance within three years. The normative time for completion of the Ph.D. is five years, and the maximum time permitted is seven years.

Graduate Program in Mathematical and Computational Biology
The graduate program in Mathematical and Computational Biology (MCB) is a one-year “gateway” program designed to function in concert with selected graduate programs, including the Ph.D. in Computer Science. The time to degree for students entering the Ph.D. program in Computer Science from MCB begins when the student first transfers to the Computer Science program. Detailed information is available at the Mathematical, Computational and Systems Biology website (http://mcsb.uci.edu) and in the School of Biological Sciences section of the Catalogue.

Faculty
Shannon L. Alfaro, M.S. University of California, Irvine, Lecturer of Computer Science
Animashree Anandkumar, Ph.D. Cornell University, Assistant Professor of Electrical Engineering and Computer Science; Computer Science (statistical inference and learning of graphical models, scalable network algorithms)
Nader Bagherzadeh, Ph.D. University of Texas at Austin, Professor of Electrical Engineering and Computer Science; Computer Science (parallel processing, computer architecture, computer graphics, VLSI design)
Pierre F. Baldi, Ph.D. California Institute of Technology, UCI Chancellor’s Professor of Computer Science; Biological Chemistry; Biomedical Engineering; Developmental and Cell Biology (bioinformatics, computational biology)
Lubomir Bic, Ph.D. University of California, Irvine, Professor of Computer Science; Biomedical Engineering; Electrical Engineering and Computer Science (parallel and distributed computing, mobile agents)
Elaheh Bozorgzadeh, Ph.D. University of California, Los Angeles, Associate Professor of Computer Science; Electrical Engineering and Computer Science (design automation and synthesis for embedded systems, VLSI CAD, reconfigurable computing)
Michael Carey, Ph.D. University of California, Berkeley, Donald Bren Professor of Information & Computer Sciences and Professor of Computer Science
Pai H. Chou, Ph.D. University of Washington, Professor of Electrical Engineering and Computer Science; Computer Science (embedded systems, wireless sensor systems, medical devices, real-time systems, hardware/software co-synthesis)
Rina Dechter, Ph.D. University of California, Los Angeles, Professor of Computer Science
Michael B. Dillencourt, Ph.D. University of Maryland, College Park, Professor of Computer Science

James P. Dourish, Ph.D. University College London, Professor of Informatics; Computer Science (human-computer interaction, computer-supported cooperative work)

Nikil D. Dutt, Ph.D. University of Illinois at Urbana-Champaign, UCI Chancellor’s Professor of Computer Science; Electrical Engineering and Computer Science (embedded systems, computer architecture, electronic design automation, software systems, brain-inspired architectures and computing)

Magda S. El Zarki, Ph.D. Columbia University, Professor of Computer Science; Electrical Engineering and Computer Science; Informatics (telecommunications, networks, wireless communication, video transmission)

David A. Eppstein, Ph.D. Columbia University, UCI Chancellor’s Professor of Computer Science

Julian Feldman, Ph.D. Carnegie Institute of Technology, Professor Emeritus of Computer Science

Charless C. Fowlkes, Ph.D. University of California, Berkeley, Associate Professor of Computer Science; Cognitive Sciences; Electrical Engineering and Computer Science (computer vision, machine learning, computational biology)

Michael S. Franz, Ph.D. Swiss Federal Institute of Technology in Zurich, Professor of Computer Science; Electrical Engineering and Computer Science (systems software, particularly compilers and virtual machines, trustworthy computing, software engineering)

Daniel H. Frost, M.S. University of California, Irvine, Senior Lecturer of Computer Science; Informatics (artificial intelligence, software engineering, computer graphics, teaching of programming)

Tony D. Givargis, Ph.D. University of California, Riverside, Professor of Computer Science; Informatics (embedded systems, platform-based system-on-a-chip design, low-power electronics)

Michael T. Goodrich, Ph.D. Purdue University, UCI Chancellor’s Professor of Computer Science; Electrical Engineering and Computer Science (computer security, algorithm design, data structures, Internet algorithmics, geometric computing, graphic drawing)

Richard H. Granger, Ph.D. Yale University, Professor Emeritus of Computer Science

Ian G. Harris, Ph.D. University of California, San Diego, Associate Professor of Computer Science; Electrical Engineering and Computer Science (hardware/software covalidation, manufacturing test)

Wayne B. Hayes, Ph.D. University of Toronto, Associate Professor of Computer Science

Dan S. Hirschberg, Ph.D. Princeton University, Professor of Computer Science; Electrical Engineering and Computer Science (analyses of algorithms, concrete complexity, data structures, models of computation)

Alexander T. Ihler, Ph.D. Massachusetts Institute of Technology, Associate Professor of Computer Science

Sandra S. Irani, Ph.D. University of California, Berkeley, Professor of Computer Science

Ramesh Chandra Jain, Ph.D. Indian Institute of Technology Kharagpur, Donald Bren Professor of Information & Computer Sciences and Professor of Computer Science

Stanislaw M. Jarecki, Ph.D. Massachusetts Institute of Technology, Professor of Computer Science

Scott A. Jordan, Ph.D. University of California, Berkeley, Professor of Computer Science; Electrical Engineering and Computer Science (pricing and differentiated services in the Internet, resource allocation in wireless networks, telecommunications policy)

Dmitri V. Kalashnikov, Ph.D. Purdue University, Associate Adjunct Professor of Computer Science

David G. Kay, J.D. Loyola Marymount University, Senior Lecturer of Informatics; Computer Science (computer law, computer science education)

Dennis F. Kibler, Ph.D. University of California, Irvine, Professor Emeritus of Computer Science

Raymond O. Klefstad, Ph.D. University of California, Irvine, Lecturer of Computer Science

Alfred Kobza, Ph.D. University of Vienna, Professor of Informatics; Computer Science (user modeling, human-computer interaction, artificial intelligence, cognitive science, interdisciplinary computer science)

Jeffrey Krichmar, Ph.D. George Mason University, Professor of Cognitive Sciences; Computer Science

Fadi J. Kurdahi, Ph.D. University of Southern California, Director, Center for Embedded Computer Systems and Professor of Electrical Engineering and Computer Science; Computer Science (VLSI system design, design automation of digital systems)
Richard H. Lathrop, Ph.D. Massachusetts Institute of Technology, Professor of Computer Science; Biomedical Engineering (modeling structure and function, machine learning, intelligent systems and molecular biology, protein structure/function prediction)

Marco Levorato, Ph.D. University of Padua, Assistant Professor of Computer Science; Electrical Engineering and Computer Science

Chen Li, Ph.D. Stanford University, Professor of Computer Science

Kwei-Jay Lin, Ph.D. University of Maryland, College Park, Professor of Electrical Engineering and Computer Science; Computer Science (real-time systems, distributed systems, service-oriented computing)

Cristina V. Lopes, Ph.D. Northeastern University, Professor of Informatics; Computer Science (programming languages, acoustic communications, operating systems, software engineering)

George S. Lueker, Ph.D. Princeton University, Professor Emeritus of Computer Science

Aditi Majumder, Ph.D. University of North Carolina at Chapel Hill, Professor of Computer Science; Electrical Engineering and Computer Science (novel displays and cameras for computer graphics and visualization, human-computer interaction, applied computer vision)

Athina Markopoulou, Ph.D. Stanford University, Associate Professor of Electrical Engineering and Computer Science; Computer Science (networking —reliability and security, multimedia networking, measurement and control, design and analysis of network protocols and algorithms, internet reliability and security, multimedia streaming, network measurements and control)

Gopi Menakshisundaram, Ph.D. University of North Carolina at Chapel Hill, Professor of Computer Science; Electrical Engineering and Computer Science (geometry and topology for computer graphics, image-based rendering, object representation, surface reconstruction, collision detection, virtual reality, telepresence)

Sharad Mehrotra, Ph.D. University of Texas at Austin, Professor of Computer Science

Eric D. Mjolsness, Ph.D. California Institute of Technology, Professor of Computer Science; Mathematics (applied mathematics, mathematical biology, modeling languages)

Alexandru Nicolau, Ph.D. Yale University, Department Chair and Professor of Computer Science; Electrical Engineering and Computer Science (architecture, parallel computation, programming languages and compilers)

Donald J. Patterson, Ph.D. University of Washington, Associate Professor of Informatics; Computer Science (ubiquitous computing, pervasive computing, human-computer interaction, artificial intelligence, intelligent context for situated computing)

Richard Pattis, M.S. Stanford University, Senior Lecturer of Computer Science; Informatics (MicroWorlds for teaching programming, debugging, computational tools for non-computer scientists)

Deva Kannan Ramanan, Ph.D. University of California, Berkeley, Associate Professor of Computer Science

Amelia C. Regan, Ph.D. University of Texas at Austin, Professor of Computer Science

Isaac D. Scherson, Ph.D. Weizmann Institute of Science, Professor of Computer Science; Electrical Engineering and Computer Science (parallel computing architectures, massively parallel systems, parallel algorithms, interconnection networks, performance evaluation)

Babak Shahbaba, Ph.D. University of Toronto, Associate Professor of Statistics; Computer Science

Phillip C-Y Sheu, Ph.D. University of California, Berkeley, Professor of Electrical Engineering and Computer Science; Biomedical Engineering; Computer Science (database systems, interactive multimedia systems)

Alice Silverberg, Ph.D. Princeton University, Professor of Mathematics; Computer Science (algebra and number theory)

Patrick J. Smyth, Ph.D. California Institute of Technology, Professor of Computer Science; Statistics

Alexander W. Thornton, B.S. University of California, Irvine, Lecturer of Computer Science

Gene Y. Tsudik, Ph.D. University of Southern California, UCI Chancellor's Professor of Computer Science

Koijro Umezaki, M.A. Dartmouth College, Associate Professor of Music; Computer Science

Alexander Veenkenbaum, Ph.D. University of Illinois at Urbana-Champaign, Professor of Computer Science

Nalini Venkatasubramanian, Ph.D. University of Illinois at Urbana-Champaign, Professor of Computer Science

Max Welling, Ph.D. Utrecht University, Professor of Computer Science; Statistics
Courses

COMPSCI 111. Digital Image Processing. 4 Units.
Introduction to the fundamental concepts of digital signal and image processing as applicable in areas such as multimedia, graphics, AI, data mining, databases, vision, or video games. Topics include image representation, space- and frequency-domain transformations, filters, segmentation, and compression.
Prerequisite: (I&C SCI 23 or CSE 23 or I&C SCI 46 or CSE 46) and I&C SCI 6D and (MATH 6G or MATH 3A or I&C SCI 6N). I&C SCI 23 with a grade of C or better. CSE 23 with a grade of C or better. I&C SCI 46 with a grade of C or better. CSE 46 with a grade of C or better. I&C SCI 6D with a grade of C or better. MATH 6G with a grade of C or better. MATH 3A with a grade of C or better. I&C SCI 6N with a grade of C or better.

COMPSCI 112. Computer Graphics. 4 Units.
Introduction to the fundamental principles of 3D computer graphics including polygonal modeling, geometric transformations, visibility algorithms, illumination models, texturing, and rasterization. Use of an independently-learned 3D graphics API to implement these techniques.
Prerequisite: (I&C SCI 22 or CSE 22 or I&C SCI H22 or I&C SCI 33 or CSE 43) and (I&C SCI 45C or CSE 45C) and (MATH 6G or MATH 3A or I&C SCI 6N). I&C SCI 22 with a grade of C or better. CSE 22 with a grade of C or better. I&C SCI H22 with a grade of C or better. I&C SCI 33 with a grade of C or better. CSE 43 with a grade of C or better. I&C SCI 45C with a grade of C or better. CSE 45C with a grade of C or better. MATH 6G with a grade of C or better. MATH 3A with a grade of C or better. I&C SCI 6N with a grade of C or better.

COMPSCI 113. Computer Game Development. 4 Units.
Introduction to the principles of interactive 2D and 3D computer game development. Concepts in computer graphics, algorithms, software engineering, art and graphics, music and sound, story analysis, and artificial intelligence are presented and are the basis for student work.
Prerequisite: COMPSCI 112 or COMPSCI 171 or IN4MATX 121 or ART 106B or I&C SCI 163 or I&C SCI 166.
Same as IN4MATX 125.

COMPSCI 114. Projects in Advanced 3D Computer Graphics. 4 Units.
Projects in advanced 3D graphics such as illumination, geometric modeling, visualization, and animation. Topics may include physically based and global illumination, solid modeling, curved surfaces, multiresolution modeling, image-based rendering, basic concepts of animation, and scientific visualization.
Prerequisite: COMPSCI 112 and (I&C SCI 45C or CSE 45C). I&C SCI 45C with a grade of C or better. CSE 45C with a grade of C or better. Recommended: COMPSCI 161 or CSE 161 or COMPSCI 164 or COMPSCI 165.

COMPSCI 115. Computer Simulation. 4 Units.
Discrete event-driven simulation; continuous system simulation; basic probability as pertaining to input distributions and output analysis; stochastic and deterministic simulation; static and dynamic system simulation.
Prerequisite: I&C SCI 6B and (MATH 6G or I&C SCI 6N) and STATS 67 and I&C SCI 51 and (I&C SCI 52 or IN4MATX 43). I&C SCI 6B with a grade of C or better. MATH 6G with a grade of C or better. I&C SCI 6N with a grade of C or better. STATS 67 with a grade of C or better. I&C SCI 51 with a grade of C or better. I&C SCI 52 with a grade of C or better. IN4MATX 43 with a grade of C or better.
Restriction: Upper-division students only.

COMPSCI 116. Computational Photography and Vision. 4 Units.
Introduces the problems of computer vision through the application of computational photography. Specific topics include photo-editing (image warping, compositing, hole filling), panoramic image stitching, and face detection for digital photographs.
Prerequisite: I&C SCI 6D and (MATH 6G or MATH 3A or I&C SCI 6N) and MATH 2B and (I&C SCI 23 or CSE 23 or I&C SCI 46 or CSE 46). I&C SCI 6D with a grade of C or better. MATH 6G with a grade of C or better. MATH 3A with a grade of C or better. I&C SCI 6N with a grade of C or better. MATH 2B with a grade of C or better. I&C SCI 23 with a grade of C or better. CSE 23 with a grade of C or better. I&C SCI 46 with a grade of C or better. CSE 46 with a grade of C or better.
COMPSCI 117. Project in Computer Vision. 4 Units.
Students undertake construction of a computer vision system. Topics may include automatically building 3D models from photographs, searching photo collections, robot navigation, and human motion tracking.

Prerequisite: I&C SCI 6D and (MATH 6G or MATH 3A or I&C SCI 6N) and MATH 2B and (I&C SCI 23 or CSE 23 or I&C SCI H23 or I&C SCI 46 or CSE 46) and (COMPSCI 112 or COMPSCI 116 or COMPSCI 171 or COMPSCI 178). I&C SCI 6D with a grade of C or better. MATH 6G with a grade of C or better. MATH 2B with a grade of C or better. I&C SCI 23 with a grade of C or better. CSE 23 with a grade of C or better. I&C SCI H23 with a grade of C or better. I&C SCI 46 with a grade of C or better. CSE 46 with a grade of C or better.

COMPSCI 121. Information Retrieval. 4 Units.
An introduction to information retrieval including indexing, retrieval, classifying, and clustering text and multimedia documents.

Prerequisite: (IN4MATX 45 or I&C SCI 46 or CSE 46 or ((I&C SCI 33 or CSE 43) and I&C SCI 45J)) and (STATS 7 or STATS 67). IN4MATX 45 with a grade of C or better. I&C SCI 46 with a grade of C or better. CSE 46 with a grade of C or better. I&C SCI 33 with a grade of C or better. CSE 43 with a grade of C or better. I&C SCI 45J with a grade of C or better.

Same as IN4MATX 141.

COMPSCI 122A. Introduction to Data Management. 4 Units.
Introduction to the design of databases and the use of database management systems (DBMS) for applications. Topics include entity-relationship modeling for design, relational data model, relational design theory, and Structured Query Language (SQL) programming.

(Design units: 1)

Prerequisite: I&C SCI 23 or CSE 23 or I&C SCI H23 or I&C SCI 46 or CSE 46 or IN4MATX 45 or I&C SCI 33 or CSE 43 or EECS 114. I&C SCI 23 with a grade of C or better. CSE 23 with a grade of C or better. I&C SCI H23 with a grade of C or better. I&C SCI 46 with a grade of C or better. CSE 46 with a grade of C or better. I&C SCI 33 with a grade of C or better. EECS 114 with a grade of C or better.

Same as EECS 116.

Restriction: School of Information and Computer Sciences majors and Computer Engineering majors have first consideration for enrollment.

COMPSCI 122B. Project in Databases and Web Applications. 4 Units.
Introduces students to advanced database technologies and Web applications. Topics include but are not limited to database connectivity (ODBC/JDBC), extending databases using stored procedures, database administration, Web servers, Web programming languages (Java servlets, XML, Ajax, and mobile platforms).

Prerequisite: (COMPSCI 122A or EECS 116) and I&C SCI 45J.

COMPSCI 122C. Principles of Data Management. 4 Units.
Covers fundamental principles underlying data management systems. Content includes key techniques including storage management, buffer management, record-oriented file system, access methods, query optimization, and query processing.

Prerequisite: COMPSCI 122A and COMPSCI 143A and COMPSCI 152.

Concurrent with COMPSCI 222.

COMPSCI 125. Next Generation Search Systems. 4 Units.
Discusses concepts and techniques related to all aspects of search systems. After considering basic search technology and the state-of-art systems, rapidly developing techniques for multimedia search, local search, event-search, and video-on-demand are explored.

Prerequisite: I&C SCI 21 or CSE 21 or IN4MATX 41 or I&C SCI 31.

Restriction: Upper-division students only.

Concurrent with COMPSCI 225.

COMPSCI 131. Parallel and Distributed Computing. 4 Units.
Parallel and distributed computer systems. Parallel programming models. Common parallel and distributed programming issues. Specific topics include parallel programming, performance models, coordination and synchronization, consistency and replication, transactions, fault tolerance.

Prerequisite: (I&C SCI 53 and I&C SCI 53L) or COMPSCI 143A.
COMPSCI 132. Computer Networks. 4 Units.
Computer network architectures, protocols, and applications. Internet congestion control, addressing, and routing. Local area networks. Multimedia networking.

(Design units: 2)
Prerequisite: EECS 55 or STATS 67.
Same as EECS 148.
Restriction: Computer Engineering and Computer Science and Engineering majors have first consideration for enrollment.

COMPSCI 133. Advanced Computer Networks. 4 Units.
Fundamental principles in computer networks are applied to obtain practical experience and skills necessary for designing and implementing computer networks, protocols, and network applications. Various network design techniques, simulation techniques, and UNIX network programming are covered.
Prerequisite: COMPSCI 132.

COMPSCI 134. Computer and Network Security. 4 Units.
Overview of modern computer and networks security, attacks, and countermeasures. Authentication, identification, data secrecy, data integrity, authorization, access control, computer viruses, network security. Also covers secure e-commerce and applications of public key methods, digital certificates, and credentials.
Prerequisite: I&C SCI 6D and (I&C SCI 33 or CSE 43 or I&C SCI 22 or CSE 22 or IN4MATX 42) and (COMPSCI 122A or EECS 116 or COMPSCI 132 or COMPSCI 143A or CSE 104).

COMPSCI 137. Internet Applications Engineering. 4 Units.
Concepts in Internet applications engineering with emphasis on the Web. Peer-to-Peer and Interoperability. Topics include HTTP and REST, Remote Procedure/Method Calls, Web Services, data representations, content distribution networks, identity management, relevant W3C/IETF standards, and relevant new large-scale computing styles.
Prerequisite: COMPSCI 132.
Same as IN4MATX 124.
Restriction: Upper-division students only.

COMPSCI 141. Concepts in Programming Languages I. 4 Units.
In-depth study of several contemporary programming languages stressing variety in data structures, operations, notation, and control. Examination of different programming paradigms, such as logic programming, functional programming and object-oriented programming; implementation strategies, programming environments, and programming style. Course may be offered online.
Prerequisite: (IN4MATX 42 or I&C SCI 51 or CSE 31 or EECS 31) and (IN4MATX 45 or I&C SCI 46 or CSE 46 or I&C SCI 33 or CSE 43). IN4MATX 42 with a grade of C or better. I&C SCI 51 with a grade of C or better. CSE 31 with a grade of C or better. EECS 31 with a grade of C or better. IN4MATX 45 with a grade of C or better. I&C SCI 46 with a grade of C or better. CSE 46 with a grade of C or better. I&C SCI 33 with a grade of C or better. CSE 43 with a grade of C or better.
Same as IN4MATX 101, CSE 141.

COMPSCI 142A. Compilers and Interpreters. 4 Units.
Introduction to the theory of programming language processors covering lexical analysis, syntax analysis, semantic analysis, intermediate representations, code generation, optimization, interpretation, and run-time support.
Prerequisite: CSE 141 or COMPSCI 141 or IN4MATX 101.
Same as CSE 142.

COMPSCI 142B. Language Processor Construction. 4 Units.
Project course which provides working laboratory experience with construction and behavior of compilers and interpreters. Students build actual language processors and perform experiments which reveal their behaviors.
Prerequisite: COMPSCI 142A or CSE 142.
COMPSCI 143A. Principles of Operating Systems. 4 Units.
Principles and concepts of process and resource management, especially as seen in operating systems. Processes, memory management, protection, scheduling, file systems, and I/O systems are covered. Concepts illustrated in the context of several well-known systems. Course may be offered online. Course may be offered online.
Prerequisite: (I&C SCI 23 or CSE 23 or I&C SCI 46 or CSE 46) and (I&C SCI 51 or EECS 31 or CSE 31).
Overlaps with EECS 111.

COMPSCI 143B. Project in Operating System Organization. 4 Units.
Detailed specification and design of critical components of an actual operating system including a memory manager, a process server, and a file/I/O subsystem. Hardware/software tradeoffs. Emphasis on logical organization of system and communication.
Prerequisite: COMPSCI 143A or CSE 104.

COMPSCI 144. High-performance Computers and Program Optimization. 4 Units.
Analyzes the relationship between computer architecture and program optimization. High-performance and parallelizing compilers for RISC, Superscalar, and VLIW architectures are discussed.
Prerequisite: I&C SCI 51. Recommended: COMPSCI 142A. I&C SCI 51 with a grade of C or better.

COMPSCI 145A. Embedded Software. 4 Units.
Principles of embedded computing systems: embedded systems architecture, hardware/software components, system software and interfacing, real-time operating systems, hardware/software co-development, and communication issues. Examples of embedded computing in real-world application domains. Simple programming using an embedded systems development environment.
Corequisite: COMPSCI 145B.
Prerequisite: CSE 46 or I&C SCI 46 or CSE 23 or I&C SCI 23 or I&C SCI 51 or CSE 31 or EECS 31.
Same as CSE 145A.

COMPSCI 145B. Embedded Software Laboratory. 2 Units.
Laboratory section to accompany CSE 145A or COMPSCI 145A.
(Design units: 0)
Corequisite: CSE145A or COMPSCI 145A.
Same as CSE 145B.

COMPSCI 146. Programming in Multitasking Operating Systems. 4 Units.
User- and systems-level programming of modern Internet-connected, multi-user, multitasking operating systems. Shells, scripting, filters, pipelines, programmability, extensibility, concurrency, inter-process communication. Concrete examples of a modern operating system (such as, but not necessarily, Unix programmed in C) are used.
Prerequisite: (I&C SCI 23 or CSE 23 or I&C SCI 46 or CSE 46) and I&C SCI 51. I&C SCI 23 with a grade of C or better. CSE 23 with a grade of C or better. I&C SCI 46 with a grade of C or better. CSE 46 with a grade of C or better. I&C SCI 51 with a grade of C or better. Recommended: COMPSCI 143A.

COMPSCI 151. Digital Logic Design. 4 Units.
Prerequisite: (I&C SCI 33 or CSE 43 or I&C SCI 23 or CSE 43) and I&C SCI 51 and I&C SCI 6B and I&C SCI 6D. I&C SCI 33 with a grade of C or better. CSE 43 with a grade of C or better. I&C SCI 23 with a grade of C or better. CSE 43 with a grade of C or better. I&C SCI 51 with a grade of C or better.

COMPSCI 152. Computer Systems Architecture. 4 Units.
Design of computer elements; ALU, control unit, and arithmetic circuits. Memory hierarchy and organization. Caches. Function unit sharing and pipelining. I/O and interrupt processing. RTL and behavioral modeling using hardware description languages. Microprocessor organization and implementation techniques.
Prerequisite: COMPSCI 151.
Overlaps with I&C SCI 160, EECS 112.
COMPSCI 153. Logic Design Laboratory. 4 Units.
Introduction to standard integrated circuits. Construction and debugging techniques. Design of digital systems using LSI and MSI components. Practical use of circuits in a laboratory environment, including implementation of small digital systems such as arithmetic modules, displays, and timers.
Prerequisite: COMPSCI 151.

COMPSCI 154. Computer Design Laboratory. 4 Units.
Underlying primitives of computer instruction sets. Principles of microprogramming. Microprogramming. Microprograms written for one or more systems. Typical microprogramming applications discussed and implemented or simulated.
Prerequisite: Prerequisite or corequisite: COMPSCI 151.

COMPSCI 151. Design and Analysis of Algorithms. 4 Units.
Techniques for efficient algorithm design, including divide-and-conquer and dynamic programming, and time/space analysis. Fast algorithms for problems applicable to networks, computer games, and scientific computing, such as sorting, shortest paths, minimum spanning trees, network flow, and pattern matching.
Prerequisite: (I&C SCI 23 or CSE 23 or I&C SCI 46 or CSE 46) and I&C SCI 6B and I&C SCI 6D and MATH 2B. I&C SCI 23 with a grade of C or better. CSE 23 with a grade of C or better. I&C SCI 46 with a grade of C or better. CSE 46 with a grade of C or better.

COMPSCI 152. Formal Languages and Automata. 4 Units.
Formal aspects of describing and recognizing languages by grammars and automata. Parsing regular and context-free languages. Ambiguity, nondeterminism. Elements of computability; Turning machines, random access machines, undecidable problems, NP-completeness.
Prerequisite: (I&C SCI 23 or CSE 23 or I&C SCI 46 or CSE 46) and MATH 2A and MATH 2B and I&C SCI 6B and I&C SCI 6D. I&C SCI 23 with a grade of C or better. CSE 23 with a grade of C or better. I&C SCI 46 with a grade of C or better. CSE 46 with a grade of C or better.
Same as LINGUIS 102.

COMPSCI 153. Graph Algorithms. 4 Units.
Algorithms for solving fundamental problems in graph theory. Graph representations, graph traversal, network flow, connectivity, graph layout, matching problems.
Prerequisite: COMPSCI 161 or CSE 161.

COMPSCI 154. Computational Geometry and Geometric Modeling. 4 Units.
Algorithms and data structures for computational geometry and geometric modeling, with applications to game and graphics programming. Topics: convex hulls, Voronoi diagrams, algorithms for triangulation, motion planning, and data structures for geometric searching and modeling of 2D and 3D objects.
Prerequisite: I&C SCI 23 or CSE 23 or I&C SCI H23 or I&C SCI 46 or CSE 46. I&C SCI 23 with a grade of C or better. CSE 23 with a grade of C or better. I&C SCI H23 with a grade of C or better. I&C SCI 46 with a grade of C or better. CSE 46 with a grade of C or better.

COMPSCI 155. Project In Algorithms And Data Structures. 4 Units.
Design, implementation, execution, and analysis of algorithms for problems such as sorting, searching, data compression, and data encryption. Time-space-structure trade-offs.
Prerequisite: COMPSCI 161 or CSE 161. Recommended: I&C SCI 45C OR CSE 45C.

COMPSCI 156. Introduction to Applied Cryptography. 4 Units.
An introduction to the essential aspects of applied cryptography, as it is used in practice. Topics include classical cryptography, block ciphers, stream ciphers, public-key cryptography, digital signatures, one-way hash functions, basic cryptographic protocols, and digital certificates and credentials.
Prerequisite: COMPSCI 161 or CSE 161.
Restriction: Upper-division students only.

COMPSCI 157. Network Optimization. 4 Units.
Network modeling techniques and related algorithms for solving large-scale integer programming problems. Exact methods and heuristic techniques. Applications include computer and communications networks and transportation and logistics networks.
Restriction: Upper-division students only.
COMPSCI 169. Introduction to Optimization. 4 Units.
Prerequisite: (I&C SCI 6N or MATH 3A or MATH 6G) and STATS 67.
Concurrent with COMPSCI 268.

COMPSCI 171. Introduction to Artificial Intelligence. 4 Units.
Different means of representing knowledge and uses of representations in heuristic problem solving. Representations considered include predicate logic, semantic nets, procedural representations, natural language grammars, and search trees.
Corequisite: STATS 67.
Prerequisite: (I&C SCI 23 or CSE 23 or I&C SCI 46 or CSE 46) and MATH 2B.

COMPSCI 172B. Neural Networks and Deep Learning . 4 Units.
Neural network and deep learning from multiple perspectives. Theory of parallel distributed processing systems, algorithmic approaches for learning from data in various manners, applications to difficult problems in AI from computer vision, to natural language understanding, to bioinformatics and chemoinformatics.
Prerequisite: STATS 120A and STATS 120B, or MATH 121A, or COMPSCI 178, or COMPSCI 273A, or equivalents.
Concurrent with COMPSCI 274C.

COMPSCI 174. Bioinformatics. 4 Units.
Introduces fundamental problems in biology that lend themselves to computational approaches. The lectures present the necessary biological background to understand the importance of the problem and the data available for algorithmic analysis.
Prerequisite: COMPSCI 171. COMPSCI 171 with a grade of C or better.

COMPSCI 175. Project in Artificial Intelligence. 4 Units.
Construction of a working artificial intelligence system. Evaluation of capabilities of the system including impact of knowledge representation.
Prerequisite: COMPSCI 171.

COMPSCI 177. Applications of Probability in Computer Science. 4 Units.
Application of probability to real-world problems in computer science. Typical topics include analysis of algorithms and graphs, probabilistic language models, network traffic modeling, data compression, and reliability modeling.
Prerequisite: MATH 2B and STATS 67 and I&C SCI 6B and I&C SCI 6D and (MATH 6G or MATH 3A or I&C SCI 6N).

COMPSCI 178. Machine Learning and Data-Mining. 4 Units.
Introduction to principles of machine learning and data mining applied to real-world datasets. Typical applications include spam filtering, object recognition, and credit scoring.
Prerequisite: I&C SCI 6B and I&C SCI 6D and (I&C SCI 6N or MATH 6G or MATH 3A) and MATH 2B and (STATS 7 or STATS 67).

COMPSCI 179. Algorithms for Probabilistic and Deterministic Graphical Models. 4 Units.
Graphical model techniques dealing with probabilistic and deterministic knowledge representations. Focuses on graphical models, such as constraint networks, Bayesian networks and Markov networks that have become a central paradigm for knowledge representation and reasoning in Artificial Intelligence and general computer science.
Prerequisite: I&C SCI 23 or CSE 23 or I&C SCI 46 or (CSE 46 and MATH 2A and MATH 2B and STATS 67)
Restriction: Prerequisite required

COMPSCI 183. Introduction to Computational Biology. 4 Units.
Prerequisite: MATH 2D or MATH 2J or STATS 7 or STATS 8.
Same as BIO SCI M123.
Concurrent with MOL BIO 223.
COMPSCI 184A. Representations and Algorithms for Molecular Biology. 4 Units.
Introduction to computational methods in molecular biology, aimed at those interested in learning about this interdisciplinary area. Covers computational approaches to understanding and predicting the structure, function, interactions, and evolution of DNA, RNA, proteins, and related molecules and processes.
Prerequisite: BIO SCI M123 or COMPSCI 183.
Concurrent with COMPSCI 284A.

COMPSCI 184B. Probabilistic Modeling of Biological Data. 4 Units.
A unified Bayesian probabilistic framework for modeling and mining biological data. Applications range from sequence (DNA, RNA, proteins) to gene expression data. Graphical models, Markov models, stochastic grammars, structure prediction, gene finding, evolution, DNA arrays, single- and multiple-gene analysis.
Prerequisite: COMPSCI 184A.
Concurrent with COMPSCI 284B.

COMPSCI 184C. Computational Systems Biology. 4 Units.
Prerequisite: COMPSCI 184A.
Concurrent with COMPSCI 284C.

COMPSCI 189. Project in Bioinformatics. 4 Units.
Teaches problem definition and analysis, data representation, algorithm design, component integration, solution validation, and testability with teams specifying, designing, building, and testing a solution to a bioinformatics problem. Lectures include engineering values, discussions, and ethical ramifications of biomedical computing issues.
Prerequisite: COMPSCI 184A. COMPSCI 184A with a grade of C or better.

COMPSCI 190. Special Topics in Information and Computer Science. 4 Units.
Studies in selected areas of Information and Computer Science. Topics addressed vary each quarter.
Prerequisite: Prerequisites vary.
Repeatability: Unlimited as topics vary.

COMPSCI H198. Honors Research. 4 Units.
Directed independent research in computer science for honors students.
Prerequisite: Satisfactory completion of the Lower-Division Writing requirement.
Repeatability: May be repeated for credit unlimited times.
Restriction: Upper-division students only. Bren School of ICS Honors Program or Campuswide Honors Program students only.

COMPSCI 199. Individual Study. 2-5 Units.
Individual research or investigation with Computer Science faculty.
Repeatability: May be repeated for credit unlimited times.

COMPSCI 200S. Seminar in Computer Science Research. 1 Unit.
Graduate colloquium series. Includes weekly talks by notable computer scientists.
Grading Option: Satisfactory/unsatisfactory only.
Repeatability: May be repeated for credit unlimited times.

COMPSCI 201. Foundations of Cryptographic Protocols. 4 Units.
Explores fundamental cryptographic tools, including encryption, signatures, and identification schemes. Students are introduced to the provable security paradigm of modern cryptography, focusing on understanding of security properties provided by cryptographic tools, and on proving security (or insecurity) of cryptographic constructions.
Prerequisite: COMPSCI 260 or COMPSCI 263.
COMPSCI 202. Applied Cryptography. 4 Units.
Design and analysis of algorithms for applied cryptography. Topics include symmetric and asymmetric key encryption, digital signatures, one-way hash functions, digital certificates and credentials, and techniques for authorization, non-repudiation, authentication, identification, data integrity, proofs of knowledge, and access control.

Prerequisite: COMPSCI 260 and COMPSCI 263.

COMPSCI 203. Network and Distributed Systems Security. 4 Units.
Modern computer and networks security: attacks and countermeasures, authentication, identification, data secrecy, data integrity, authorization, access control, computer viruses, network security. Group communication and multicast security techniques. Covers secure e-commerce and applications of public key methods, digital certificates, and credentials.

Prerequisite: EECS 148 or COMPSCI 132.
Same as NET SYS 240.

COMPSCI 206. Principles of Scientific Computing. 4 Units.
Overview of widely used principles and methods of numerical and scientific computing, including basic concepts and computational methods in linear algebra, optimization, and probability.

Prerequisite: Basic courses in multivariate calculus, linear algebra, and probability.
Overlaps with STATS 230.

COMPSCI 211A. Visual Computing. 4 Units.
Fundamentals of image processing (convolution, linear filters, spectral analysis), vision geometry (projective geometry, camera models and calibration, stereo reconstruction), radiometry (color, shading, illumination, BRDF), and visual content synthesis (graphics pipeline, texture-, bump-, mip-mapping, hidden surface removal, anti-aliasing).

COMPSCI 211B. Computer Graphics and Visualization. 4 Units.
Interactive 3D graphics rendering pipeline, illumination and shading, ray tracing, texture-, bump-, mip-mapping, hidden surface removal, anti-aliasing, multiresolution representations, volume rendering techniques, iso-surface extraction.

Prerequisite: COMPSCI 211A.

COMPSCI 212. Multimedia Systems and Applications. 4 Units.
Organization and structure of modern multimedia systems; audio and video encoding/compression; quality of service concepts; scheduling algorithms for multimedia; resource management in distributed and multimedia systems; multimedia protocols over high-speed networks; synchronization schemes; multimedia applications; and teleservices.

Prerequisite: (COMPSCI 143A and COMPSCI 161) or B.S. degree in Computer Science. Recommended: COMPSCI 131 and COMPSCI 132 and COMPSCI 133.

COMPSCI 213. Introduction to Visual Perception. 4 Units.
Introduction to the process of human visual perception. Offers the physiological and psychophysical approach to understand vision, introducing concepts of perception of color, depth, movement. Examples of quantification and application of these models in computer vision, computer graphics, multimedia, HCI.

Prerequisite: MATH 121A.

COMPSCI 216. Image Understanding. 4 Units.
The goal of image understanding is to extract useful semantic information from image data. Course covers low-level image and video processing techniques, feature descriptors, segmentation, objection recognition, and tracking.

Prerequisite: I&C SCI 6D and (I&C SCI 6N or MATH 6G or MATH 3A) and MATH 2B and I&C SCI 46.

COMPSCI 217. Light and Geometry in Computer Vision. 4 Units.
Examines the issues of light transport and multiview geometry in computer vision. Applications include camera calibration, 3D understanding, stereo reconstruction, and illumination estimation.

Prerequisite: I&C SCI 6D and (I&C SCI 6N or MATH 6G or MATH 3A) and MATH 2B and I&C SCI 46 and COMPSCI 211A.
COMPSCI 221. Information Retrieval, Filtering, and Classification. 4 Units.
Algorithms for the storage, retrieval, filtering, and classification of textual and multimedia data. The vector space model, Boolean and probabilistic queries, and relevance feedback. Latent semantic indexing; collaborative filtering; and relationship to machine learning methods.

Prerequisite: COMPSCI 161 and COMPSCI 171 and (I&C SCI 6N or MATH 3A or MATH 6G).

Same as IN4MATX 225.

Restriction: Graduate students only.

COMPSCI 222. Principles of Data Management. 4 Units.
Covers fundamental principles underlying data management systems. Content includes key techniques including storage management, buffer management, record-oriented file system, access methods, query optimization, and query processing.

Prerequisite: COMPSCI 122A and COMPSCI 143A and COMPSCI 152.

Concurrent with COMPSCI 122C.

COMPSCI 223. Transaction Processing and Distributed Data Management. 4 Units.
Covers fundamental principles underlying transaction processing including database consistency, concurrency control, database recovery, and fault-tolerance. Includes transaction processing in centralized, distributed, parallel, and client-server environments.

Prerequisite: COMPSCI 222 and COMPSCI 131.

COMPSCI 224. Advanced Topics in Data Management. 4 Units.
Selected advanced topics in data management. Content differs in each offering and with instructor's interests. Intended for students interested in data management with focus on reading and critiquing recent research papers, presentations, and substantial research projects.

Prerequisite: COMPSCI 143A and COMPSCI 152 and COMPSCI 161 and COMPSCI 222 and COMPSCI 223.

COMPSCI 225. Next Generation Search Systems. 4 Units.
Discusses concepts and techniques related to all aspects of search systems. After considering basic search technology and the state-of-art systems, rapidly developing techniques for multimedia search, local search, event-search, and video-on-demand are explored.

Prerequisite: I&C SCI 21 or CSE 21 or IN4MATX 41 or I&C SCI 31 or CSE 41.

Restriction: Upper-division or Graduate students only.

Concurrent with COMPSCI 125.

COMPSCI 230. Distributed Computer Systems. 4 Units.
Principles of distributed computing systems. Topics covered include message-passing, remote procedure calls, distributed shared memory synchronization, resource and process/thread management, distributed file systems, naming and security.

COMPSCI 232. Computer and Communication Networks. 4 Units.

Prerequisite: EECS 148 or COMPSCI 132.

Same as EECS 248A, NET SYS 201.

Restriction: Graduate students only.

COMPSCI 233. Networking Laboratory. 4 Units.
A laboratory-based introduction to basic networking concepts such as addressing, sub-netting, bridging, ARP, and routing. Network simulation and design. Structured around weekly readings and laboratory assignments.

Prerequisite: EECS 148 or COMPSCI 132.

Same as NET SYS 202.
COMPSCI 234. Advanced Networks. 4 Units.
Design principles of networked systems, advanced routing and congestion control algorithms, network algorithms, network measurement, management, security, Internet economics, and emerging networks.

Prerequisite: NET SYS 201 or COMPSCI 232 OR EECS 248A.

Same as NET SYS 210.

COMPSCI 236. Wireless and Mobile Networking. 4 Units.
Introduction to wireless networking. The focus is on layers 2 and 3 of the OSI reference model, design, performance analysis, and protocols. Topics covered include: an introduction to wireless networking, digital cellular, next generation cellular, wireless LANs, and mobile IP.

Prerequisite: EECS 148 or COMPSCI 132.

Same as NET SYS 230.

COMPSCI 237. Middleware for Networked and Distributed Systems. 4 Units.
Discusses concepts, techniques, and issues in developing distributed systems middleware that provides high performance and Quality of Service for emerging applications. Also covers existing standards (e.g., CORBA, DCOM, Jini, Espeak) and their relative advantages and shortcomings.

Prerequisite: An undergraduate-level course in operating systems and networks.

Same as NET SYS 260.

COMPSCI 240. Language-Based Security. 4 Units.
Teaches state-of-the-art language-based techniques for increasing the security and reliability of software systems. Covers static (e.g., bytecode verification, proof-carrying code) and dynamic (e.g., reference monitors, stack inspection) techniques. Also discusses information flow and securing legacy code.

Prerequisite: COMPSCI 230 or COMPSCI 242 or COMPSCI 262.

COMPSCI 241. Advanced Compiler Construction. 4 Units.
Advanced study of programming language implementation techniques: optimizations such as common sub-expression elimination, register allocation, and instruction scheduling. Implementation of language features such as type-directed dispatch, garbage collection, dynamic linking, and just-in-time code generation.

Prerequisite: COMPSCI 142A.

COMPSCI 242. Parallel Computing. 4 Units.

COMPSCI 243. High-Performance Architectures and Their Compilers. 4 Units.
Emphasis on the development of automatic tools (i.e., compilers/environments) for the efficient exploitation of parallel machines, and the trade-offs between hardware and software in the design of supercomputing and high-performance machines.

COMPSCI 244. Introduction to Embedded and Ubiquitous Systems. 4 Units.
Embedded and ubiquitous system technologies including processors, DSP, memory, and software. System interfacing basics; communication strategies; sensors and actuators, mobile and wireless technology. Using pre-designed hardware and software components. Design case studies in wireless, multimedia, and/or networking domains.

Prerequisite: I&C SCI 51 and COMPSCI 152 and COMPSCI 161 and (I&C SCI 6N or MATH 3A or MATH 6G or I&C SCI 6D) or B.S. degree in Computer Science.

Same as IN4MATX 244.

COMPSCI 245. Software for Embedded Systems. 4 Units.

Prerequisite: I&C SCI 51 and COMPSCI 152 and COMPSCI 161 and (I&C SCI 6N or MATH 3A or MATH 6G or I&C SCI 6D) or B.S. degree in Computer Science.
COMPSCI 246. Validation and Testing of Embedded Systems. 4 Units.

Prerequisite: B.S. degree in Computer Science or basic courses in algorithms & data structures, calculus, discrete math, linear algebra, symbolic logic.

COMPSCI 247. Design Automation and Prototyping of Embedded Systems. 4 Units.

Prerequisite: I&C SCI 6D and I&C SCI 51 and COMPSCI 152 and COMPSCI 161 and COMPSCI 244 and (I&C SCI 6N or MATH 3A or MATH 6G) or B.S. degree in Computer Science.

COMPSCI 248A. Introduction to Ubiquitous Computing. 4 Units.
The "disappearing computer" paradigm. Differences to the desktop computing model: applications, interaction in augmented environments, security, alternate media, small operating systems, sensors, and embedded systems design. Evaluation by project work and class participation.

Same as IN4MATX 241.

COMPSCI 248B. Ubiquitous Computing and Interaction. 4 Units.
Principles and design techniques for ubiquitous computing applications. Conceptual basis for tangible and embodied interaction. Interaction in virtual and augmented environments. Design methods and techniques. Design case studies. Examination by project work.

Prerequisite: IN4MATX 231 and IN4MATX 241.

Same as IN4MATX 242.

COMPSCI 249S. Seminar in Compilers and Operating Systems. 2 Units.
Current research and research trends in system-level software such as compilers and operating systems. Forum for presentation and criticism by students of new published research and work in progress.

Prerequisite: (COMPSCI 142A and COMPSCI 143A) or B.S. degree in Computer Science.

Repeatability: May be taken for credit 4 times.

COMPSCI 250A. Computer Systems Architecture. 4 Units.
Study of architectural issues and their relation to technology and software: design of processor, interconnections, and memory hierarchies.

Prerequisite: COMPSCI 152.

COMPSCI 250B. Modern Microprocessors. 4 Units.
Fundamental concepts and recent advances in computer architecture necessary to understand and use modern microprocessors. Topics span out-of-order execution, multiple instruction issue, control/data speculation, predication, advanced cache and DRAM organizations, embedded systems, DSP and multi-media instructions.

Prerequisite: COMPSCI 250A.

Overlaps with COMPSCI 243.

COMPSCI 252. Introduction to Computer Design. 4 Units.
The methodology and use of CAD tools for computer design, accomplished by a lab in which students practice design using commercially available silicon compilers and other tools.

Prerequisite: COMPSCI 151 and COMPSCI 152.

COMPSCI 253. Analysis of Programming Languages. 4 Units.
Concepts in modern programming languages, their interaction, and the relationship between programming languages and methods for large-scale, extensible software development. Empirical analysis of programming language usage.

Same as IN4MATX 212.

COMPSCI 259S. Seminar in Design Science. 2 Units.
Current research and research trends in design science. Forum for presentation and criticism by students of research work in progress.

Repeatability: May be taken for credit 18 times.
COMPSCI 260. Fundamentals of the Design and Analysis of Algorithms. 4 Units.
Covers fundamental concepts in the design and analysis of algorithms and is geared toward non-specialists in theoretical computer science. Topics include: deterministic and randomized graph algorithms, fundamental algorithmic techniques like divide-and-conquer strategies and dynamic programming, and NP-completeness.
Prerequisite: COMPSCI 161.

COMPSCI 261. Data Structures. 4 Units.
An in-depth treatment of data structures and their associated management algorithms including resource complexity analysis.
Prerequisite: I&C SCI 46 and COMPSCI 161.

COMPSCI 262. Computational Complexity. 4 Units.
Advanced course in computational models and complexity classes. Covers the fundamentals of Turing Machines, Decidability, and NP-completeness. Includes discussion of more advanced topics including polynomial hierarchy, randomized complexity classes, #P-completeness and hardness of approximation.
Prerequisite: COMPSCI 162.

COMPSCI 263. Analysis of Algorithms. 4 Units.
Analysis of correctness and complexity of various efficient algorithms; discussion of problems for which no efficient solutions are known.
Prerequisite: COMPSCI 161 and COMPSCI 261.

COMPSCI 264. Quantum Computation and Information. 4 Units.
Basic models for quantum computation and their foundations in quantum mechanics. Quantum complexity classes and quantum algorithms including algorithms for factoring and quantum simulation. Introduction to quantum information theory and quantum entanglement.
Prerequisite: Basic courses in linear algebra and algorithms.

COMPSCI 265. Graph Algorithms. 4 Units.
Graph definitions, representation methods, graph problems, algorithms, approximation methods, and applications.
Prerequisite: COMPSCI 161 and COMPSCI 261.

COMPSCI 266. Computational Geometry. 4 Units.
An overview of some of the basic problems in computational geometry and of some algorithmic and data-structuring techniques appropriate to their solution.
Prerequisite: COMPSCI 161 and COMPSCI 261.

COMPSCI 267. Data Compression. 4 Units.
An introduction to the theory and practice of modern data compression techniques. Topics include codes, coding, modeling, text compression, lossless and lossy image compression standards and systems, audio compression.
Prerequisite: (COMPSCI 261 and COMPSCI 260) or COMPSCI 261.

COMPSCI 268. Introduction to Optimization. 4 Units.
Prerequisite: STATS 67 and (I&C SCI 6N or MATH 3A or MATH 6G).
Concurrent with COMPSCI 169.

COMPSCI 269S. Seminar in the Theory of Algorithms and Data Structures. 2 Units.
Current research and research trends in the Theory of algorithms and data structures.
Repeatability: May be taken for credit 18 times.

COMPSCI 271. Introduction to Artificial Intelligence. 4 Units.
The study of theories and computational models for systems which behave and act in an intelligent manner. Fundamental subdisciplines of artificial intelligence including knowledge representation, search, deduction, planning, probabilistic reasoning, natural language parsing and comprehension, knowledge-based systems, and learning.
COMPSCI 273A. Machine Learning. 4 Units.
Computational approaches to learning algorithms for classifications, regression, and clustering. Emphasis is on discriminative classification methods such as decision trees, rules, nearest neighbor, linear models, and naive Bayes.
Prerequisite: COMPSCI 271 and COMPSCI 206.

COMPSCI 274A. Probabilistic Learning: Theory and Algorithms. 4 Units.
An introduction to probabilistic and statistical techniques for learning from data, including parameter estimation, density estimation, regression, classification, and mixture modeling.
Prerequisite: COMPSCI 206.

COMPSCI 274B. Learning in Graphical Models. 4 Units.
Models for data analysis are presented in the unifying framework of graphical models. The emphasis is on learning from data but inference is also covered. Real world examples are used to illustrate the material.
Prerequisite: COMPSCI 274A.

COMPSCI 274C. Neural Networks and Deep Learning. 4 Units.
Neural network and deep learning from multiple perspectives. Theory of parallel distributed processing systems, algorithmic approaches for learning from data in various manners, applications to difficult problems in AI from computer vision, to natural language understanding, to bioinformatics and chemoinformatics.
Prerequisite: STATS 120A and STATS 120B, or MATH 121A or COMPSCI 178 or COMPSCI 273A, or equivalents.
Overlaps with COMPSCI 274A, COMPSCI 279S, COMPSCI 277, COMPSCI 276, COMPSCI 278, COMPSCI 274B.
Concurrent with COMPSCI 172B.

COMPSCI 275. Network-based Reasoning/Constraint Networks. 4 Units.
Study of the theory and techniques of constraint network model. Covers techniques for solving constraint satisfaction problems: backtracking techniques, consistency algorithms, and structure-based techniques. Tractable subclasses. Extensions into applications such as temporal reasoning, diagnosis, and scheduling.
Prerequisite: Basic course in algorithm design and analysis.

COMPSCI 276. Network-based reasoning/Belief Networks. 4 Units.
Focuses on reasoning with uncertainty using "Bayes Networks" that encode knowledge as probabilistic relations between variables, and the main task is, given some observations, to update the degree of belief in each proposition.
Prerequisite: A basic course in probability.

COMPSCI 277. Data Mining. 4 Units.
Introduction to the general principles of inferring useful knowledge from large data sets (commonly known as data mining or knowledge discovery). Relevant concepts from statistics, databases and data structures, optimization, artificial intelligence, and visualization are discussed in an integrated manner.
Prerequisite: COMPSCI 273A or COMPSCI 274A.

COMPSCI 278. Probability Models. 4 Units.
Advanced probability, discrete time Markov chains, Poisson processes, continuous time Markov chains. Queuing or simulation as time permits.
Prerequisite: STATS 120A.
Concurrent with STATS 121.

COMPSCI 279S. Seminar in Artificial Intelligence. 2 Units.
Current research and research trends in artificial intelligence.
Repeatability: May be taken for credit 18 times.
COMPSCI 284A. Representations and Algorithms for Molecular Biology. 4 Units.
Introduction to computational methods in molecular biology, aimed at those interested in learning about this interdisciplinary area. Covers computational approaches to understanding and predicting the structure, function, interactions, and evolution of DNA, RNA, proteins, and related molecules and processes.

Prerequisite: A Basic course in algorithms, or a basic course in molecular biology.
Concurrent with COMPSCI 184A.

COMPSCI 284B. Probabilistic Modeling of Biological Data. 4 Units.
A unified Bayesian probabilistic framework for modeling and mining biological data. Applications range from sequence (DNA, RNA, proteins) to gene expression data. Graphical models, Markov models, stochastic grammars, structure prediction, gene finding, evolution, DNA arrays, single- and multiple-gene analysis.

Prerequisite: COMPSCI 284A.
Concurrent with COMPSCI 184B.

COMPSCI 284C. Computational Systems Biology. 4 Units.

Prerequisite: COMPSCI 284A or COMPSCI 284B or (BIO SCI 99 and MATH 2D and MATH 2J).
Concurrent with COMPSCI 184C.

COMPSCI 285. Mathematical and Computational Biology. 4 Units.

Prerequisite: MATH 227A.
Same as MATH 227C.

COMPSCI 288A. Biological Networks. 4 Units.
Introduces the basics of primarily graph theoretic analysis and modeling of biological networks. Presents the necessary biological background for understanding different types of biological networks as well as mathematical, algorithmic, and computational complexity issues associated with them.

Prerequisite: I&C SCI 6D and (COMPSCI 161 or CSE 161) and BIO SCI M123.

COMPSCI 289S. Seminar for Informatics in Biology and Medicine. 2 Units.
Current research and research trends in bioinformatics and medical informatics. Forum for presentation and criticism by students of recently published research and work in progress.

Prerequisite: COMPSCI 284A or COMPSCI 284B.
Repeatability: May be repeated for credit unlimited times.

COMPSCI 290. Research Seminar. 2 Units.
Forum for presentation and criticism by students of research work in progress. Presentation of problem areas and related work. Specific goals and progress of research.

Grading Option: Satisfactory/unsatisfactory only.
Repeatability: May be repeated for credit unlimited times.
Restriction: Graduate students only.

COMPSCI 295. Special Topics in Information and Computer Science. 4 Units.
Studies in selected areas of Information and Computer Science. Topics addressed vary each quarter.

Repeatability: Unlimited as topics vary.
Restriction: Graduate students only.
COMPSCI 296. Elements of Scientific Writing. 4 Units.
Introduces the concepts and principles of good scientific writing, demonstrates them by examples drawn from the literature, and uses a hands-on approach to apply them to documents being written by the participants.

Grading Option: Satisfactory/unsatisfactory only.

COMPSCI 298. Thesis Supervision. 2-12 Units.
Individual research or investigation conducted in preparation for the M.S. thesis option or the dissertation requirements for the Ph.D. program.

Repeatability: May be repeated for credit unlimited times.

Restriction: Graduate students only. School of Information and Computer Science majors only.

COMPSCI 299. Individual Study. 1-12 Units.
Individual research or investigation with Computer Science faculty.

Repeatability: May be repeated for credit unlimited times.

Restriction: Graduate students only. School of Information and Computer Science majors only.