Donald Bren School of Information and Computer Sciences

On This Page:

- Degrees
- Honors
- Careers

Hal S. Stern, Dean
6210 Donald Bren Hall
Academic Counseling: 949-824-5156
http://www.ics.uci.edu/

Overview

The Donald Bren School of Information and Computer Sciences (ICS) embodies excellence, creativity, and collaborative innovation in computer science and information technology. As the only independent computing school in the University of California system, it is well-positioned to continue its tradition of exploring and advancing the boundaries of a broad, multidisciplinary field on a global scale.

ICS faculty have extensive training in traditional computer science, as well as engineering, mathematics and statistics, and the social sciences. The School’s stand-alone structure, as opposed to being part of an engineering school, enables the faculty to take the broadest possible view of computer science and information technology. This breadth is reflected in the diverse set of academic degree options for undergraduate and graduate students, some of which are interdisciplinary and jointly administered with other academic units.

The School’s three departments — Computer Science, Informatics, and Statistics — fuel a wide range of instructional and research efforts, including: design of algorithms and data structures; computer architecture and embedded computer systems; networked and distributed systems; systems software; social and mobile computing; artificial intelligence, machine learning and data mining; computer games and virtual worlds; databases and information retrieval; computer graphics and visualization; bioinformatics, computational biology and genomics; computer-supported cooperative work, human-centered computing and human-computer interaction; security and privacy; software engineering; managerial and social aspects of computing technology; and statistics. The vibrant Bren School community continues to explore innovative topics ranging from building complete computer systems on chips smaller than a human fingernail to developing user-interface systems that allow workers on opposite sides of the world to collaborate effectively. Bren School research continues to focus on how computing and information technology can be used to solve a broad set of real-world problems, such as improving how first responders communicate during a crisis, optimizing transportation systems, analyzing data to expedite biological research, and improving network security.

Faculty are active participants in and leaders of numerous research institutes spanning computer science, including the Institute for Genomics and Bioinformatics; Institute for Software Research; Institute for Virtual Environments and Computer Games; Center for Embedded Computer Systems; California Institute for Telecommunications and Information Technology (Calit2); Center for Machine Learning and Intelligent Systems; Center for Organizational Research; Center for Digital Transformation; Genetic Epidemiology Research Institute; Center for Pervasive Communications and Computing; Laboratory for Ubiquitous Computing and Interaction; Secure Computing and Networking Center; Institute for Mathematical Behavioral Sciences; Center for Ethnography; Institute for Transportation Studies; and Ada Byron Research Center.

Faculty and student-driven research in the Bren School is supported through a variety of grants, gifts, and contracts from public and private institutions such as the State of California, the U.S. Department of Education, various U.S. defense agencies, the National Science Foundation, the National Institutes of Health, NASA, and various companies, including The Aerospace Corporation, Boeing, Disney, Experian, Google, IBM, Intel, Microsoft, Samsung, and Yahoo!

Faculty and alumni of the Bren School of ICS have contributed some of computing’s most significant advancements, including revolutionizing computer-aided drafting techniques; the creation of the current Hypertext Transfer Protocol (HTTP/1.1); development of the Internet standards for HTTP and Uniform Resource Identifiers (URI); the founding of the Apache HTTP Server Project that produces the software for more than 60 percent of public Internet websites; and the creation of the Domain Name System (DNS) that translates Web and e-mail addresses into the numeric system used to route information along the Internet.

The Bren School is committed to increasing diversity in the computing and information technology fields. The Ada Byron Research Center was created in 2003 to address research and outreach topics aimed at increasing the participation of women and other underrepresented populations in computer science, engineering, digital media, and related information technology areas. The School is an active partner of the National Center for Women & Information Technology, whose overarching goal is parity in the professional information technology workforce.
Degrees

<table>
<thead>
<tr>
<th>Degree</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Information Management</td>
<td>B.S.</td>
</tr>
<tr>
<td>Computer Game Science</td>
<td>B.S.</td>
</tr>
<tr>
<td>Computer Science</td>
<td>B.S., M.S., Ph.D.</td>
</tr>
<tr>
<td>Computer Science and Engineering</td>
<td>B.S.</td>
</tr>
<tr>
<td>Data Science</td>
<td>B.S.</td>
</tr>
<tr>
<td>Informatics</td>
<td>B.S., Ph.D.</td>
</tr>
<tr>
<td>Information and Computer Science</td>
<td>B.S., M.S., Ph.D.</td>
</tr>
<tr>
<td>Networked Systems</td>
<td>M.S., Ph.D.</td>
</tr>
<tr>
<td>Software Engineering</td>
<td>B.S., M.S., Ph.D.</td>
</tr>
<tr>
<td>Statistics</td>
<td>M.S., Ph.D.</td>
</tr>
</tbody>
</table>

1 Offered jointly with The Paul Merage School of Business. See the Interdisciplinary Studies section of the Catalogue for information.
2 Offered jointly with The Henry Samueli School of Engineering. See the Interdisciplinary Studies section of the Catalogue for information.
3 Admission to the Ph.D. program is no longer available.

Honors

Honors at graduation, e.g., *cum laude, magna cum laude, summa cum laude*, are awarded to approximately the top 12 percent of the graduating seniors. A general criterion is that a student must have completed at least 72 units in residence at the University of California. The student’s cumulative record at the end of the final quarter is the basis for consideration of awarding Latin Honors. Other important factors are considered visit at Honors Recognition.

Careers

Graduates of the Donald Bren School of Information and Computer Sciences pursue a variety of careers. Many graduates specify, design, and develop a variety of computer-based systems comprising software and hardware in virtually every application domain, such as aerospace, automotive, biomedical, consumer products, engineering, entertainment, environmental, finance, investment, law, management, manufacturing, and pharmacology. Graduates also find jobs as members of research and development teams, developing advanced technologies, designing software and hardware systems, and specifying, designing, and maintaining computing infrastructures for a variety of institutions. Some work for established or start-up companies, while others work as independent consultants. After a few years in industry, many move into management or advanced technical positions. Many students also use the undergraduate major as preparation for graduate study in computer science or another field (e.g., medicine, law, engineering, management).

On This Page:

- Admissions
- Transfer Student Policy
- Change of Major
- Special Programs and Courses
- Concentration: Engineering and Computer Science in the Global Context
- Undergraduate Major in Business Information Management (BIM)
- Undergraduate Major in Computer Game Science
  - Sample Program of Study — Computer Game Science
- Undergraduate Major in Information and Computer Science
- Undergraduate Major in Software Engineering
  - Sample Program of Study — Software Engineering

Undergraduate Program

A Bren School of ICS undergraduate education is a blend of scholarship, science, technology, and practical application that forms an excellent foundation for professional life.

The basis of the undergraduate program is a set of fundamental courses in mathematics and computer science, supplemented by general education courses from other academic disciplines. A premium is placed on both communication and quantitative skills. Students quickly gain hands-on experience with advanced computing systems, and intense use of computer and network technologies continues throughout the undergraduate program. Students study data organization, algorithm design and analysis, design and organization of hardware and network systems, software engineering, artificial
intelligence, social aspects of system design and use, and management of technology. In the process, students work with state-of-the-art hardware and software technologies, and learn several contemporary programming languages.

The Bren School offers seven majors: Business Information Management (BIM), offered jointly with The Paul Merage School of Business; Computer Game Science (CGS); Computer Science (CompSci); Computer Science and Engineering (CSE), offered jointly with The Henry Samueli School of Engineering; Informatics (IN4MATX); Information and Computer Science (ICS); and Software Engineering (SE). There are also programs of study leading to minors in Bioinformatics, Digital Information Systems, Health Informatics, Informatics, Information and Computer Science, and Statistics.

**B.S. in Business Information Management.** The undergraduate Business Information Management (BIM) major seeks to educate students to understand and apply the theories and concepts of a broad, integrated curriculum covering computing (computer science, informatics, and software); business fundamentals (accounting, finance, marketing, strategy, and operations); and analytical methods (mathematics, statistics, economics, management science, and decision analysis). The fundamentals of information and computer science provide the foundation for understanding and evaluating the technology through which most of the business information is gathered and presented, while the business fundamentals provide a background and context in which information and its analysis will be applied. The major is administered by the Donald Bren School of Information and Computer Sciences and is a collaborative, interdisciplinary degree program between the Bren School and The Paul Merage School of Business.

**B.S. in Computer Game Science.** The Computer Game Science (CGS) major combines a solid foundation in computer science with a focus on designing, building, and understanding computer games and other forms of interactive media. The fundamentals of information and computer science, along with course work in mathematics, statistics, physics, and film and media studies, provide students with the concepts and tools to study a wide scope of computer game technologies. The major emphasizes design, collaboration, and the understanding of computer games and related technologies and media in a social and cultural context.

**B.S. 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: Algorithms, Architecture and Embedded Systems, Bioinformatics, Information, Intelligent Systems, Networked Systems, Systems and Software, and Visual Computing. Additional electives can be used to satisfy a second specialization or obtain a broader understanding of the field.

**B.S. in Computer Science and Engineering.** This program is designed to provide students with the fundamentals of computer science, both hardware and software, and the application of engineering concepts, techniques, and methods to both computer systems engineering and software systems design. The Computer Science and Engineering (CSE) major gives students access to multidisciplinary problems in engineering with a focus on total systems engineering. Students learn the computer science principles that are critical to development of software, hardware, and networking of computer systems. From that background, engineering concepts and methods are added to give students exposure to circuit design, network design, and digital signal processing. Elements of engineering practice include a systems view, manufacturing and economic issues, and multidisciplinary engineering applications. The 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 in The Henry Samueli School of Engineering.

**B.S. in Informatics.** Within the overall discipline of information and computer science, the Informatics (IN4MATX) major is concerned with the relationship between what is inside the computer and what is outside. The Informatics major addresses the broad set of issues surrounding design, ranging from initial requirements-gathering to estimating and measuring the impact of alternative solutions—all from a multidisciplinary perspective that includes computer science, information science, organizational science, social science, and cognitive science. Students pursuing the B.S. in Informatics complete a specialization in one of two areas: Human-Computer Interaction, or Organizations and Information Technology.

**B.S. in Information and Computer Science.** The degree in Information and Computer Science is an individually designed degree. Students must submit a proposal for a four-year plan of study along with a rationale for why the proposed plan is a well-motivated set of courses that does not fit into any of the existing ICS majors. Students submitting proposals are strongly encouraged to follow the lower-division requirements for one of the Bren School majors (or provide a rationale for why this is not appropriate) and should include at least 48 units of upper-division ICS, Computer Science, Informatics, or Statistics courses. Proposals must be approved by the ICS Associate Dean for Student Affairs.

**B.S. in Software Engineering.** The Software Engineering major prepares students to be productive members of software engineering teams in a variety of application domains, with focus on the domains of major importance within each decade. It combines a solid foundation in computer science with knowledge of how to engineer modern software systems, and how to function within teams. Course work in mathematics and statistics provides students the basis for rigorous thinking; course work in the foundations of computer science provides students the basis for computational thinking; course work in topics of software engineering prepares students for the production of software; project courses prepare students for the practice of software development. The major emphasizes the design and implementation of large software systems.

Admissions
To ensure admission consideration for the fall quarter, students should be sure to file their application by November 30 of the prior year. The selection criteria include grades, test scores, and other considerations.

Transfer Student Policy
Transfer requirements vary by major.

Business Information Management
Computer Game Science
Computer Science
Computer Science and Engineering
Data Science
Informatics
Software Engineering

NOTE TO TRANSFER APPLICANTS: These majors require a series of lower-division courses, and prerequisites constrain the order in which they can be taken. Junior-level transfer students who must complete a significant part of this sequence may find that it will take longer than two years at UCI to complete their degree. Python, Java, and C++ are used in the curriculum; therefore, transfer students should plan to learn these languages by studying on their own or by completing related programming courses prior to their first quarter at UCI.

Change of Major
Students interested in changing their major to one offered by the School should contact the ICS Student Affairs Office for more information and assistance. Information is also available at the UCI Change of Major Criteria website (http://www.changeofmajor.uci.edu).

Major and minor restrictions: Click on the "Major/Minor Restrictions" tab at the top of this page.

Special Programs and Courses
The Bren School of ICS Honors Program
The Bren School of ICS Honors Program provides selected upper-division students an opportunity to carry out a research project under the direction of a faculty member in the School. Eligible students participate in the ICS Honors Seminar (I&C SCI H197), which provides an introduction to the range of current faculty research. Each student then affiliates with an ICS faculty advisor who agrees to supervise a minimum of two quarters of research. The participating student prepares a final written research report and submits a copy for review to both the faculty advisor and the Honors Program advisor. Successful completion of the Honors Program earns the student a certificate and medal from the School. Further, a notation of successful completion is added to the student’s transcript. For more information about course requirements, application procedures, and deadlines visit http://honors.ics.uci.edu/, or contact the Student Affairs Office at 949 824-5156.

Other Opportunities
Bren School of ICS undergraduates may complement their educational experience by participating in other programs. Information about the following programs is available elsewhere in the Catalogue and via the program Web sites: Campuswide Honors Program, Undergraduate Research Opportunities Program, Education Road Program, and Student Achievement Guided by Experience (SAGE Scholars).

Concentration: Engineering and Computer Science in the Global Context
The globalization of the marketplace for information technology services and products makes it likely that ICS graduates will work in multicultural settings or be employed by companies with extensive international operations or customer bases. The goal of the concentration is to help students develop and integrate knowledge of the history, language, and culture of a country or geographic region outside the United States, through course work both at UCI and an international host campus, followed by a technology-related internship in the host country.

All Bren School majors in good standing may propose an academic plan that demonstrates the ability to complete the concentration (a minimum of eight courses) and other requirements for graduation in a reasonable time frame. It is expected that a student’s proposal will reflect a high degree of planning that includes the guidance of academic counselors and those at the UCI Study Abroad Center regarding course selection, as well as considerations related to internship opportunities, housing, and financial aid. Each student’s proposed program of study must be approved by the Bren School of ICS Associate Dean for Student Affairs. The Associate Dean will be available to assist qualified students with the development of a satisfactory academic plan, as needed.

The concentration consists of the following components:

1. A minimum of eight courses at UCI or at the international campus with an emphasis on the culture, language (if applicable and necessary), history, literature of the country that corresponds to the international portion of the program, international law, international labor policy, global issues, global institutions, global conflict and negotiation, and global economics;
2. A one- or two-semester sequence of technical courses related to the major and, possibly, culture, history, and literature courses taken at an international university;
3. A two-month or longer technical internship experience in the same country as the international educational experience.

More information about the requirements for the concentration is available in the ICS Student Affairs Office.

**Undergraduate Major in Business Information Management (BIM)**

This program is administered jointly by the Bren School of ICS and The Paul Merage School of Business. For information, see the Interdisciplinary Studies section of the *Catalogue*.

**Requirements for the B.S. Degree in Business Information Management**

All students must meet the University Requirements.

Major Requirements: See the Interdisciplinary Studies section of the *Catalogue*.

**Undergraduate Major in Computer Game Science**

The Computer Game Science major gives students a strong foundation in introductory information and computer science, an extensive education in technologies and design practices associated with computer games, and an opportunity to focus in two areas of particular interest to the student. Students who complete the major will be able to create interactive and human-centered game designs; implement games using skills in modeling, graphics, software engineering, hardware architectures, human interfaces, and aesthetics; and evaluate games and game technology for their use in education, art, and social change.

**Career Paths.** A wide variety of careers and graduate programs are open to Computer Game Science (CGS) graduates. The video game industry is comparable in size to the film and music industries, and job growth projections are strong for people with strong technical backgrounds. Many other fields, including mobile software development, interactive entertainment, and training and education software have demand for similar skill sets and knowledge. CGS graduates are well-trained in computer science, and can thus pursue graduate programs or any career that involves designing, implementing, evaluating, or interacting with computer-based systems.

**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, two semester courses equivalent to other major-related math courses are acceptable.
2. Complete 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.

Transfer applicants to the Computer Game Science major should be aware that several lower-division courses must be taken at UCI; therefore, the minimum time to degree completion will exceed two years.

**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.

**Requirements for the B.S. Degree in Computer Game Science**

All students must meet the University Requirements.

**Major Requirements**

**Lower-division**

A. Select one of the two groups of courses:

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I&amp;C SCI 21</td>
<td>Introduction to Computer Science I</td>
</tr>
<tr>
<td>I&amp;C SCI 22</td>
<td>Introduction to Computer Science II</td>
</tr>
<tr>
<td>I&amp;C SCI 45C</td>
<td>Programming in C/C++ as a Second Language</td>
</tr>
<tr>
<td>I&amp;C SCI 46</td>
<td>Data Structure Implementation and Analysis</td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>I&amp;C SCI 31</td>
<td>Introduction to Programming</td>
</tr>
<tr>
<td>I&amp;C SCI 32</td>
<td>Programming with Software Libraries</td>
</tr>
<tr>
<td>I&amp;C SCI 33</td>
<td>Intermediate Programming</td>
</tr>
<tr>
<td>I&amp;C SCI 45C</td>
<td>Programming in C/C++ as a Second Language</td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Title</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>I&amp;C SCI 46</td>
<td>Data Structure Implementation and Analysis</td>
</tr>
<tr>
<td>B. Complete:</td>
<td></td>
</tr>
<tr>
<td>I&amp;C SCI 51</td>
<td>Introductory Computer Organization</td>
</tr>
<tr>
<td>C. Complete:</td>
<td></td>
</tr>
<tr>
<td>IN4MATX 43</td>
<td>Introduction to Software Engineering</td>
</tr>
<tr>
<td>or I&amp;C SCI 52</td>
<td>Introduction to Software Engineering</td>
</tr>
<tr>
<td>D. Complete:</td>
<td></td>
</tr>
<tr>
<td>MATH 2A</td>
<td>Single-Variable Calculus</td>
</tr>
<tr>
<td>MATH 2B</td>
<td>Single-Variable Calculus</td>
</tr>
<tr>
<td>I&amp;C SCI 6N</td>
<td>Computational Linear Algebra</td>
</tr>
<tr>
<td>or MATH 3A</td>
<td>Introduction to Linear Algebra</td>
</tr>
<tr>
<td>I&amp;C SCI 6B</td>
<td>Boolean Algebra and Logic</td>
</tr>
<tr>
<td>I&amp;C SCI 6D</td>
<td>Discrete Mathematics for Computer Science</td>
</tr>
<tr>
<td>STATS 67</td>
<td>Introduction to Probability and Statistics for Computer Science</td>
</tr>
<tr>
<td>E. Complete:</td>
<td></td>
</tr>
<tr>
<td>I&amp;C SCI 60</td>
<td>Computer Games and Society</td>
</tr>
<tr>
<td>I&amp;C SCI 61</td>
<td>Game Systems and Design</td>
</tr>
<tr>
<td>I&amp;C SCI 62</td>
<td>Game Technologies and Interactive Media</td>
</tr>
<tr>
<td>F. Complete:</td>
<td></td>
</tr>
<tr>
<td>PHYSICS 3A</td>
<td>Basic Physics</td>
</tr>
<tr>
<td>G. Complete:</td>
<td></td>
</tr>
<tr>
<td>FLM&amp;MDA 85A</td>
<td>Introduction to Film and Visual Analysis</td>
</tr>
<tr>
<td>or FLM&amp;MDA 85C</td>
<td>New Media and Digital Technologies</td>
</tr>
</tbody>
</table>

### Upper-division

A. Computer Game Science Core Requirements

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>I&amp;C SCI 161</td>
<td>Game Engine Lab</td>
</tr>
<tr>
<td>I&amp;C SCI 167</td>
<td>Multiplayer Game Systems</td>
</tr>
<tr>
<td>I&amp;C SCI 168</td>
<td>Multiplayer Game Project</td>
</tr>
<tr>
<td>I&amp;C SCI 169A-169B</td>
<td>Capstone Game Project I and Capstone Game Project II</td>
</tr>
</tbody>
</table>

and select two of the following:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<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>

B. Computer Science Core

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 112</td>
<td>Computer Graphics</td>
</tr>
<tr>
<td>COMPSCI 171</td>
<td>Introduction to Artificial Intelligence</td>
</tr>
</tbody>
</table>

C. Select two of the following:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<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 143A</td>
<td>Principles of Operating Systems</td>
</tr>
<tr>
<td>COMPSCI 152</td>
<td>Computer Systems Architecture</td>
</tr>
<tr>
<td>IN4MATX 113</td>
<td>Requirements Analysis and Engineering</td>
</tr>
<tr>
<td>IN4MATX 121</td>
<td>Software Design: Applications</td>
</tr>
<tr>
<td>IN4MATX 131</td>
<td>Human Computer Interaction</td>
</tr>
</tbody>
</table>

D. CGS Elective Courses:

Five additional courses chosen from those listed in E

E. At least three of the 16 upper-division courses satisfying A–D must be in the same Bren ICS track.

### Bren ICS Tracks:

#### Algorithms

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 161</td>
<td>Design and Analysis of Algorithms</td>
</tr>
<tr>
<td>COMPSCI 162</td>
<td>Formal Languages and Automata</td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Title</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------</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 165</td>
<td>Project In Algorithms And Data Structures</td>
</tr>
<tr>
<td><strong>Artificial Intelligence</strong></td>
<td></td>
</tr>
<tr>
<td>COMPSCI 171</td>
<td>Introduction to Artificial Intelligence</td>
</tr>
<tr>
<td>COMPSCI 174</td>
<td>Bioinformatics</td>
</tr>
<tr>
<td>COMPSCI 175</td>
<td>Project in Artificial Intelligence</td>
</tr>
<tr>
<td>COMPSCI 177</td>
<td>Applications of Probability in Computer Science</td>
</tr>
<tr>
<td>COMPSCI 178</td>
<td>Machine Learning and Data-Mining</td>
</tr>
<tr>
<td>COMPSCI 179</td>
<td>Algorithms for Probabilistic and Deterministic Graphical Models</td>
</tr>
<tr>
<td><strong>Computational Biology</strong></td>
<td></td>
</tr>
<tr>
<td>COMPSCI 183</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>COMPSCI 184C</td>
<td>Computational Systems Biology</td>
</tr>
<tr>
<td><strong>Computer Graphics and Vision</strong></td>
<td></td>
</tr>
<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><strong>Computer Networks</strong></td>
<td></td>
</tr>
<tr>
<td>COMPSCI 131</td>
<td>Parallel and Distributed Computing</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 137</td>
<td>Internet Applications Engineering</td>
</tr>
<tr>
<td>IN4MATX 124</td>
<td></td>
</tr>
<tr>
<td><strong>Databases</strong></td>
<td></td>
</tr>
<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 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><strong>Hardware</strong></td>
<td></td>
</tr>
<tr>
<td>COMPSCI 145A</td>
<td>Embedded Software</td>
</tr>
<tr>
<td>COMPSCI 151</td>
<td>Digital Logic Design</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><strong>Human-Computer Interaction</strong></td>
<td></td>
</tr>
<tr>
<td>IN4MATX 131</td>
<td>Human Computer Interaction</td>
</tr>
<tr>
<td>IN4MATX 132</td>
<td>Project in Human-Computer Interaction Requirements and Evaluation</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><strong>Operating Systems</strong></td>
<td></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>
<tr>
<td>COMPSCI 144</td>
<td>High-performance Computers and Program Optimization</td>
</tr>
<tr>
<td>COMPSCI 146</td>
<td>Programming in Multitasking Operating Systems</td>
</tr>
<tr>
<td><strong>Programming Languages and Compilers</strong></td>
<td></td>
</tr>
<tr>
<td>IN4MATX 101</td>
<td>Concepts in Programming Languages I (same as COMPSCI 141)</td>
</tr>
<tr>
<td>IN4MATX 102</td>
<td>Concepts of Programming Language II</td>
</tr>
<tr>
<td>COMPSCI 142A</td>
<td>Compilers and Interpreters</td>
</tr>
<tr>
<td>COMPSCI 142B</td>
<td>Language Processor Construction</td>
</tr>
</tbody>
</table>
### Project Management
- IN4MATX 151: Project Management
- IN4MATX 161: Social Analysis of Computing
- IN4MATX 162W: Organizational Information Systems

### Simulation and Optimization
- COMPSCI 115: Computer Simulation
- COMPSCI 168: Network Optimization
- COMPSCI 169: Introduction to Optimization

### Social Impacts of Computing
- IN4MATX 161: Social Analysis of Computing
- IN4MATX 162W: Organizational Information Systems
- IN4MATX 163: Project in the Social and Organizational Impacts of Computing

### Software Design
- IN4MATX 121: Software Design: Applications
- IN4MATX 122: Software Design: Structure and Implementation
- IN4MATX 123: Software Architecture

### Software Engineering
- IN4MATX 113: Requirements Analysis and Engineering
- IN4MATX 115: Software Testing, Analysis, and Quality Assurance
- IN4MATX 124: Internet Applications Engineering (same as COMPSCI 137)

### Non-Track Courses (some of these courses have prerequisites that are not part of the CGS major):

#### Computer Game Science
- I&C SCI 162: Modeling and World Building
- I&C SCI 163: Mobile and Ubiquitous Games
- I&C SCI 166: Game Design

#### Business Management
- MGMT 101: Management Science
- MGMT 102: Managing Organizational Behavior
- MGMT 105: Introduction to Marketing
- MGMT 154: International Marketing
- MGMT 155: Brand Management

#### Cognitive Science
- PSYCH 130A: Perception and Sensory Processes
- PSYCH 131A: Vision
- PSYCH 131B: Hearing
- PSYCH 135M: The Mind/Body Problem
- PSYCH 140C: Cognitive Science

#### Mathematics
- MATH 112A-112B-112C: Introduction to Partial Differential Equations and Applications
- MATH 115: Mathematical Modeling
- MATH 121A-121B: Linear Algebra and Linear Algebra

#### Film and Media Studies
- FLM&MDA 113: Narrative/Image
- FLM&MDA 114: Film, Media, and the Arts
- FLM&MDA 117A: Introduction to Screenwriting
- FLM&MDA 144: Studies in New Media

With prior approval of the ICS Associate Dean for Student Affairs, a student may design a new track, or an Independent Study, Honors Research, or Special Topics course may be substituted for a course in a track. Computer Game Science (CGS) elective courses may not be counted as part of the Management minor.
## Sample Program of Study — Computer Game Science

<table>
<thead>
<tr>
<th>Freshman</th>
<th>Winter</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall</td>
<td></td>
<td></td>
</tr>
<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>I&amp;C SCI 60&lt;sup&gt;1&lt;/sup&gt;</td>
<td>I&amp;C SCI 61</td>
<td>I&amp;C SCI 62</td>
</tr>
<tr>
<td>MATH 2A</td>
<td>MATH 2B</td>
<td>I&amp;C SCI 6B</td>
</tr>
<tr>
<td>WRITING 39A</td>
<td>WRITING 39B</td>
<td>WRITING 39C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sophomore</th>
<th>Winter</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I&amp;C SCI 45C</td>
<td>I&amp;C SCI 46</td>
<td>IN4MATX 43</td>
</tr>
<tr>
<td>I&amp;C SCI 51</td>
<td>I&amp;C SCI 160</td>
<td>I&amp;C SCI 161</td>
</tr>
<tr>
<td>I&amp;C SCI 6N</td>
<td>COMPSCI 112</td>
<td>STATS 67</td>
</tr>
<tr>
<td>PHYSICS 3A</td>
<td>I&amp;C SCI 6D</td>
<td>GE III/VII</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Junior</th>
<th>Winter</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I&amp;C SCI 162, 163, or 166&lt;sup&gt;3&lt;/sup&gt;</td>
<td>I&amp;C SCI 167</td>
<td>I&amp;C SCI 162, 163, or 166&lt;sup&gt;3&lt;/sup&gt;</td>
</tr>
<tr>
<td>COMPSCI 171</td>
<td>COMPSCI 122A, IN4MATX 113, IN4MATX 121, or IN4MATX 131&lt;sup&gt;3&lt;/sup&gt;</td>
<td>I&amp;C SCI 168</td>
</tr>
<tr>
<td>FLMMMDA 85A&lt;sup&gt;2&lt;/sup&gt;</td>
<td>UD Writing</td>
<td>CGS Elective</td>
</tr>
<tr>
<td>GE IV/VIII</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Senior</th>
<th>Winter</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CGS Elective</td>
<td>I&amp;C SCI 169A</td>
<td>I&amp;C SCI 169B</td>
</tr>
<tr>
<td>CGS Elective</td>
<td>CGS Elective</td>
<td>CGS Elective</td>
</tr>
<tr>
<td>GE III</td>
<td>GE IV</td>
<td>GE VI</td>
</tr>
</tbody>
</table>

1  Fulfills GE III.
2  Fulfills GE IV.
3  Select one of these.

### Undergraduate Major in Information and Computer Science

The Information and Computer Science major is intended for highly motivated students who are currently enrolled at UCI, who find that their academic and career interests are not well served by any of the existing ICS majors, and would be better served by a uniquely designed course of study.

#### Application Process

New students are not admitted directly to the Information and Computer Science major. Continuing students can apply to change their major to Information and Computer Science no earlier than the fall quarter of their sophomore year. Students must submit a proposal for a four-year plan of study, along with rationale explaining why the proposed plan is a well-motivated and coherent set of courses that does not fit into any of the existing ICS majors. Students submitting proposals are strongly encouraged to follow the lower-division requirements for one of the ICS majors (or provide a rationale for why this is not appropriate) and should include at least 48 units of upper-division ICS, Computer Science, Informatics, or Statistics courses. Proposals must be approved by the ICS Associate Dean for Student Affairs. See the ICS Student Affairs Office for more details. Complete information about changing majors to ICS is available at the UCI Change of Major Criteria website (http://www.changeofmajor.uci.edu).

#### Admissions

New students are not admitted directly to the Information and Computer Science major.

#### Transfer Applicants:

Students are strongly encouraged to follow the transfer preparation guidelines for any of the other Bren ICS majors.

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

All students must meet the University Requirements.

Major Requirements: See the ICS Student Affairs Office.

### Undergraduate Major in Software Engineering

The Software Engineering major gives students a strong foundation in software engineering as well as a solid basis in computer science. Students who complete the major will be able to be productive members of software engineering teams in a variety of application domains including, but not restricted
to, Web and mobile applications. The acquired technical knowledge and appreciation for life-long learning, combined with the ability to place software in the social context in which it is developed, empowers students to create novel applications that have the potential to bring social change.

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. Completion of 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; at least one of these should involve 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 Software Engineering
All students must meet the University Requirements.

Major Requirements

Lower-division
A. Select one of the following:
   I&C SCI 21-22  Introduction to Computer Science I
                 and Introduction to Computer Science II
   or
   IN4MATX 41-42  Informatics Core Course I
                 and Informatics Core Course II
   or
   I&C SCI 31-32-33  Introduction to Programming
                      and Programming with Software Libraries
                      and Intermediate Programming

B. Complete:
   I&C SCI 45C  Programming in C/C++ as a Second Language

C. Complete:
   I&C SCI 45J  Programming in Java as a Second Language

D. Complete:
   I&C SCI 46  Data Structure Implementation and Analysis

E. Complete:
   IN4MATX 43  Introduction to Software Engineering
   or I&C SCI 52  Introduction to Software Engineering

F. Complete:
   I&C SCI 51  Introductory Computer Organization

G. Complete:
   MATH 2A-2B  Single-Variable Calculus
               and Single-Variable Calculus
   I&C SCI 6B  Boolean Algebra and Logic
   I&C SCI 6D  Discrete Mathematics for Computer Science
   I&C SCI 6N  Computational Linear Algebra
   or MATH 3A  Introduction to Linear Algebra
   STATS 67  Introduction to Probability and Statistics for Computer Science
### Upper-division

#### A. Core Requirements

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 122A</td>
<td>Introduction to Data Management</td>
</tr>
<tr>
<td>COMPSCI 143A</td>
<td>Principles of Operating Systems</td>
</tr>
<tr>
<td>COMPSCI 132</td>
<td>Computer Networks</td>
</tr>
<tr>
<td>COMPSCI 161</td>
<td>Design and Analysis of Algorithms</td>
</tr>
<tr>
<td>IN4MATX 101/COMPSCI 141</td>
<td>Concepts in Programming Languages I</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 131</td>
<td>Human Computer Interaction</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>or IN4MATX 124</td>
<td>Internet Applications Engineering</td>
</tr>
<tr>
<td>IN4MATX 151</td>
<td>Project Management</td>
</tr>
<tr>
<td>IN4MATX 191A</td>
<td>Senior Design Project</td>
</tr>
<tr>
<td>IN4MATX 191B</td>
<td>Senior Design Project</td>
</tr>
<tr>
<td>I&amp;C SCI 139W</td>
<td>Critical Writing on Information Technology</td>
</tr>
</tbody>
</table>

#### B. Select four 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 125/COMPSCI 113</td>
<td>Computer Game Development</td>
</tr>
<tr>
<td>IN4MATX 132</td>
<td>Project in Human-Computer Interaction Requirements and Evaluation</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>IN4MATX 141/COMPSCI 121</td>
<td>Information Retrieval</td>
</tr>
<tr>
<td>IN4MATX 143</td>
<td>Information Visualization</td>
</tr>
<tr>
<td>IN4MATX 148</td>
<td>Project in Ubiquitous Computing</td>
</tr>
<tr>
<td>IN4MATX 161</td>
<td>Social Analysis of Computing</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 142A</td>
<td>Compilers and Interpreters</td>
</tr>
<tr>
<td>COMPSCI 142B</td>
<td>Language Processor Construction</td>
</tr>
<tr>
<td>COMPSCI 145A-145B</td>
<td>Embedded Software and Embedded Software Laboratory</td>
</tr>
<tr>
<td>COMPSCI 146</td>
<td>Programming in Multitasking Operating Systems</td>
</tr>
<tr>
<td>COMPSCI 165</td>
<td>Project In Algorithms And Data Structures</td>
</tr>
<tr>
<td>I&amp;C SCI 160</td>
<td>Graphics Processors and Game Platforms</td>
</tr>
<tr>
<td>I&amp;C SCI 167</td>
<td>Multiplayer Game Systems</td>
</tr>
<tr>
<td>I&amp;C SCI 168</td>
<td>Multiplayer Game Project</td>
</tr>
</tbody>
</table>

Software Engineering elective courses may not be counted as part of the Management minor.

### Career Paths

A wide variety of careers and graduate programs are open to Software Engineering graduates. The Web and mobile applications industry is a multi-billion dollar industry, and job growth projections are the strongest for people with strong technical backgrounds. Many other application domains, including interactive entertainment, medical information systems, and training and education software have demand for similar skill sets and knowledge. Graduate school in either computer science or software engineering or a related IT field is also a possible career path.

### Sample Program of Study — Software Engineering

#### 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></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 45C</td>
<td>I&amp;C SCI 46</td>
<td>COMPSCI 122A</td>
</tr>
<tr>
<td>I&amp;C SCI 51</td>
<td>IN4MATX 113</td>
<td>COMPSCI 143A</td>
</tr>
<tr>
<td>I&amp;C SCI 6D</td>
<td>I&amp;C SCI 6N</td>
<td>STATS 67</td>
</tr>
<tr>
<td>GE III/VII</td>
<td>IN4MATX 131</td>
<td>GE IV</td>
</tr>
</tbody>
</table>

Junior

<table>
<thead>
<tr>
<th>Fall</th>
<th>Winter</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>I&amp;C SCI 45J</td>
<td>IN4MATX 101</td>
<td>IN4MATX 123</td>
</tr>
<tr>
<td>IN4MATX 115</td>
<td>IN4MATX 122</td>
<td>IN4MATX 191A</td>
</tr>
<tr>
<td>IN4MATX 121</td>
<td>IN4MATX 151</td>
<td>COMPSCI 132</td>
</tr>
<tr>
<td>GE III</td>
<td>GE IV/VIII</td>
<td>GE IV</td>
</tr>
</tbody>
</table>

Senior

<table>
<thead>
<tr>
<th>Fall</th>
<th>Winter</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 161</td>
<td>I&amp;C SCI 139W</td>
<td>SE Elective</td>
</tr>
<tr>
<td>IN4MATX 191B</td>
<td>SE Elective</td>
<td>SE Elective</td>
</tr>
<tr>
<td>SE Elective</td>
<td></td>
<td>GE VI</td>
</tr>
</tbody>
</table>

Important Notes:

Students who have completed both I&C SCI 21 and I&C SCI 22 with grades of C or better and who wish to change majors to Informatics may use I&C SCI 21 and I&C SCI 22 in satisfaction of the requirements of IN4MATX 41 and IN4MATX 42.

Students who have completed both IN4MATX 41 and IN4MATX 42 with grades of C or better and who wish to change majors to Computer Game Science, Computer Science, or Information and Computer Science may use IN4MATX 41 and IN4MATX 42 in satisfaction of the requirement of I&C SCI 21 and I&C SCI 22.

Students enrolled in other degree programs who are interested in the field of computer science may pursue the Bren School introductory course sequences (I&C SCI 31/CSE 41, I&C SCI 32/CSE 42, and I&C SCI 33/CSE 43) followed by other courses for which they have met the prerequisites as far as their interests require and their programs permit. The introductory courses, along with other lower-division ICS courses, may be used to fulfill General Education requirements. Nonmajors may also take other Bren ICS courses for which they have met the prerequisites.

The ICS Student Affairs Office is staffed by professional academic counselors and peer advisors. These individuals are available to assist students with program planning, questions on University and School policies and procedures, progress toward graduation, and other issues that arise in the course of a student’s education. Faculty also are available for advising, generally for suggestions of additional course work in the student’s academic, research, and career interest areas and on preparation for graduate school.

On This Page:

- Minor in Bioinformatics
- Minor in Digital Information Systems
- Minor in Health Informatics
- Minor in Information and Computer Science
- NOTES:
  b. A maximum of two courses may be taken for the Pass/Not Pass grade option toward any minor
  c. There are no applications for a Bren ICS minor. Students must have a peer advisor or academic counselor of their major add the minor to their Degree Audit once they begin pursuing the minor.
  d. To ensure you are certified for the minor at graduation, the minor must be on your Degree Audit and Application for Graduation. Your Student Affairs Office will certify your minor at time of graduation, so it is important to keep them updated on your academic progress.

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:

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I&amp;C SCI 21-22</td>
<td>Introduction to Computer Science I and Introduction to Computer Science II</td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>I&amp;C SCI 31/CSE 41</td>
<td>Introduction to Programming</td>
</tr>
</tbody>
</table>

Complete:

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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.

**Minor in Digital Information Systems**

Students outside the Bren School of ICS may pursue a minor in Digital Information Systems (DIS). The minor is designed for students who want to learn about information systems, computation, and digital communication without preparing to be computer programmers. Students completing the DIS minor will be able to understand the role of digital information systems in society, and will learn about the technological underpinnings of these systems and constraints on their design and use.

**Requirements for the Minor**

Select two of the following:

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I&amp;C SCI 3</td>
<td>Internet Technologies and their Social Impact</td>
</tr>
<tr>
<td>I&amp;C SCI 4</td>
<td>Human Factors for the Web</td>
</tr>
<tr>
<td>I&amp;C SCI 5</td>
<td>Global Disruption and Information Technology</td>
</tr>
<tr>
<td>I&amp;C SCI 7</td>
<td>Introducing Modern Computational Tools</td>
</tr>
<tr>
<td>I&amp;C SCI 8</td>
<td>Practical Computer Security</td>
</tr>
<tr>
<td>I&amp;C SCI 11</td>
<td>The Internet and Public Policy</td>
</tr>
<tr>
<td>I&amp;C SCI 22</td>
<td>Introduction to Computer Science II</td>
</tr>
<tr>
<td>or CSE 22</td>
<td>Introduction to Computer Science II</td>
</tr>
<tr>
<td>I&amp;C SCI H22</td>
<td>Honors Introduction to Computer Science II</td>
</tr>
<tr>
<td>IN4MATX 42</td>
<td>Informatics Core Course II</td>
</tr>
<tr>
<td>or I&amp;C SCI 32</td>
<td>Programming with Software Libraries</td>
</tr>
<tr>
<td>IN4MATX 43</td>
<td>Introduction to Software Engineering</td>
</tr>
<tr>
<td>I&amp;C SCI 61</td>
<td>Game Systems and Design</td>
</tr>
</tbody>
</table>

Select one of the following:

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I&amp;C SCI 10</td>
<td>How Computers Work</td>
</tr>
<tr>
<td>I&amp;C SCI 21</td>
<td>Introduction to Computer Science I</td>
</tr>
<tr>
<td>or CSE 21</td>
<td>Introduction to Computer Science I</td>
</tr>
<tr>
<td>I&amp;C SCI H21</td>
<td>Honors Introduction to Computer Science I</td>
</tr>
<tr>
<td>IN4MATX 41</td>
<td>Informatics Core Course I</td>
</tr>
<tr>
<td>or I&amp;C SCI 31</td>
<td>Introduction to Programming</td>
</tr>
</tbody>
</table>

Select four of the following:

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I&amp;C SCI 105</td>
<td>Digital Information Systems</td>
</tr>
<tr>
<td>IN4MATX 131</td>
<td>Human Computer Interaction</td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Title</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>IN4MATX 132</td>
<td>Project in Human-Computer Interaction Requirements and Evaluation</td>
</tr>
<tr>
<td>IN4MATX 143</td>
<td>Information Visualization</td>
</tr>
<tr>
<td>IN4MATX 148</td>
<td>Project in Ubiquitous Computing</td>
</tr>
<tr>
<td>IN4MATX 151</td>
<td>Project Management</td>
</tr>
<tr>
<td>IN4MATX 161</td>
<td>Social Analysis of Computing</td>
</tr>
<tr>
<td>IN4MATX 162W</td>
<td>Organizational Information Systems</td>
</tr>
<tr>
<td>IN4MATX 163</td>
<td>Project in the Social and Organizational Impacts of Computing</td>
</tr>
<tr>
<td>IN4MATX 171</td>
<td>Introduction to Medical Informatics</td>
</tr>
<tr>
<td>IN4MATX 172</td>
<td>Project in Health Informatics</td>
</tr>
</tbody>
</table>

1 Students cannot take both IN4MATX 43 and I&C SCI 105.

NOTE: Bren School of ICS majors may not minor in Digital Information Systems. Courses used to complete the minor in Digital Information Systems may not also count toward the requirements for the Information and Computer Science minor or the Informatics minor.

**Minor in Health Informatics**

The minor in Health Informatics prepares students to understand the expanding role of information technology (IT) in health care and to participate in creating IT solutions to health care issues. It includes course work and fieldwork addressing a variety of health care IT settings. Students completing this minor will gain practical experience applying IT to serve the health care needs of communities and individuals.

**Requirements for the Minor**

Complete:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN4MATX 171</td>
<td>Introduction to Medical Informatics</td>
</tr>
<tr>
<td>IN4MATX 172</td>
<td>Project in Health Informatics</td>
</tr>
</tbody>
</table>

Select two of the following:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>I&amp;C SCI 4</td>
<td>Human Factors for the Web ¹</td>
</tr>
<tr>
<td>I&amp;C SCI 7</td>
<td>Introducing Modern Computational Tools ¹</td>
</tr>
<tr>
<td>I&amp;C SCI 10</td>
<td>How Computers Work ¹</td>
</tr>
<tr>
<td>I&amp;C SCI 31</td>
<td>Introduction to Programming ¹</td>
</tr>
<tr>
<td>I&amp;C SCI 32</td>
<td>Programming with Software Libraries ¹</td>
</tr>
<tr>
<td>IN4MATX 121</td>
<td>Software Design: Applications</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 143</td>
<td>Information Visualization</td>
</tr>
<tr>
<td>COMPSCI 111</td>
<td>Digital Image Processing</td>
</tr>
<tr>
<td>COMPSCI 121/IN4MATX 141</td>
<td>Information Retrieval</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 134</td>
<td>Computer and Network Security</td>
</tr>
<tr>
<td>COMPSCI/CSE 145A</td>
<td>Embedded Software</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>
</tbody>
</table>

Select two of the following:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUR SCI 110W</td>
<td>Frameworks for Professional Nursing Practice</td>
</tr>
<tr>
<td>PUBH/LTH 101</td>
<td>Introduction to Epidemiology</td>
</tr>
<tr>
<td>PUBH/LTH 104</td>
<td>Analytic and Applied Epidemiology</td>
</tr>
<tr>
<td>PUBH/LTH 122</td>
<td>Health Policy</td>
</tr>
<tr>
<td>PUBH/LTH 124</td>
<td>Environmental and Public Health Policy</td>
</tr>
</tbody>
</table>

Select one of the following:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN4MATX 151</td>
<td>Project Management</td>
</tr>
<tr>
<td>IN4MATX 161</td>
<td>Social Analysis of Computing</td>
</tr>
<tr>
<td>IN4MATX 162W</td>
<td>Organizational Information Systems</td>
</tr>
</tbody>
</table>
STATS 7  Basic Statistics
STATS 8  Introduction to Biological Statistics
STATS 67  Introduction to Probability and Statistics for Computer Science

1  This course may only be counted by majors outside of the Bren School of ICS.

NOTE: No more than one of these courses may be used to satisfy both the requirements of this minor and the requirements of the student’s major. A student must earn a grade of C or better in all courses used to satisfy the requirements of this minor.

Minor in Information and Computer Science

Students outside the School may also pursue a minor in Information and Computer Science. The minor provides a focused study of Information and Computer Science to supplement a student’s major program of study and prepares students for a profession, career, or academic pursuit in which computer science is an integral part but is not the primary focus. The ICS minor contributes to students’ competence in computing technology and proficiency in programming as well as exposing them to the fundamentals of computer science. The minor allows students sufficient flexibility to pursue courses that complement their major field or address specific interests.

Requirements for the Minor

Select one of the following groups:

I&C SCI 21- 22- 46  Introduction to Computer Science I
                     and Introduction to Computer Science II
                     and Data Structure Implementation and Analysis

or

I&C SCI 31- 32- 33- 45C- 46  Introduction to Programming
                               and Programming with Software Libraries
                               and Intermediate Programming
                               and Programming in C/C++ as a Second Language
                               and Data Structure Implementation and Analysis

Complete:

I&C SCI 6D  Discrete Mathematics for Computer Science

Complete one of:

I&C SCI 51  Introductory Computer Organization
or I&C SCI 52
or IN4MATX 43  Introduction to Software Engineering

Select two upper-division from the following: 1

CS 111-144
CS 151-177
IN4MATX 101-102
IN4MATX 111-119
IN4MATX 123  Software Architecture
IN4MATX 125  Computer Game Development
IN4MATX 131  Human Computer Interaction
IN4MATX 132-134
IN4MATX 141  Information Retrieval
IN4MATX 148  Project in Ubiquitous Computing
IN4MATX 153  Computer Supported Cooperative Work
IN4MATX 161-163
IN4MATX 171  Introduction to Medical Informatics

1  COMPSCI 190–199 and IN4MATX 190–199 may not be applied to the minor.

On This Page:

• Admission
• Financial Assistance
• Students with a Previously Earned Master's Degree
• Course Substitutions
Graduate Programs in Information and Computer Sciences


ICS M.S. students must complete one of the following concentrations: Embedded Systems or Informatics (INF).

For additional information about the following graduate programs and requirements, click on these links: Computer Science; Informatics; Statistics; Software Engineering; Networked Systems, which is supervised by an interdepartmental faculty group from the Department of Computer Science in the Bren School and the Department of Electrical Engineering and Computer Science in The Henry Samueli School of Engineering. Information is available on the Interdisciplinary Studies section (http://catalogue.uci.edu/interdisciplinarystudies) of the Catalogue.

Admission

Applicants will be evaluated on the basis of their prior academic record. Applicants for the M.S. degree are expected to have a bachelor’s degree in computer science or a related field. Those who do not have an undergraduate degree in computer science may take the Computer Science Subject GRE test to demonstrate sufficient background in the field. Scores are reviewed on a case-by-case basis. Ph.D. applicants will additionally be evaluated in their potential for creative research and teaching in Information and Computer Sciences.

Applicants are expected to have (1) skills in computer programming at least equivalent to those obtained in college-level courses in programming and language development; (2) skills in mathematics equivalent to those obtained in complete college-level courses in logic and set theory, analysis, linear algebra and modern algebra, or probability and statistics; (3) data structures, analysis of algorithms, automata theory, or formal languages; and (4) computer architectures.

All applicants are evaluated on the materials submitted: letters of recommendation, official GRE test scores, official college transcripts, and personal statement. For more information, contact the ICS graduate counselor at 949-824-5156 or send e-mail to gcounsel@ics.uci.edu.

Financial Assistance

Financial assistance is available to Ph.D. students in the form of fellowships, teaching assistantships, and research assistantships. Although assistance varies, it is the School’s goal to support all entering Ph.D. students, subject to availability of funds. International students who are not citizens of countries where English is either the primary or dominant language, as approved by Graduate Council, and who apply for teaching assistantships must take one of the approved English proficiency examinations. More information is available in the Graduate Division section of the Catalogue.

Students with a Previously Earned Master's Degree

Credit for one or all required courses may be given at the time of admission to those students who have completed a master’s degree in computer science or a closely related field. Course equivalency will be determined by the Bren School Associate Dean for Student Affairs following a written recommendation from a sponsoring research advisor. Research advisors can require that a student take additional courses when this is appropriate.

An additional M.S. degree will not be awarded if the student currently holds an M.S. degree in computer science or a related field from another university.

Course Substitutions

A student who has taken relevant graduate courses at UCI or another university may petition to have a specific course certified as equivalent to one which satisfies Bren School of ICS requirements. The petition should describe the course and should be approved by either the student’s advisor or the instructor teaching the class, and by the Associate Dean for Student Affairs. Only two courses can be substituted.

Master of Science Program


Students pursuing the M.S. in Information and Computer Science must complete a concentration in Embedded Systems or Informatics (INF).

For additional information about the following graduate programs and requirements, click on these links: Computer Science; Software Engineering; Statistics; Networked Systems.

M.S. students may select one of two options, the thesis plan or the comprehensive examination plan, as described below. The normative time for completion of the M.S. degree is two years. All study must be completed within four calendar years from the date of admission.
Plan I: Thesis Plan. The thesis option is available for graduate students who may wish to continue on to a Ph.D. program or those who wish to concentrate on a specific problem. To qualify for this option, students must be in good academic standing with their Department. The student must enroll in at least two quarters of Thesis Supervision (COMPSCI 298 or IN4MATX 298) that will substitute for two required courses as specified under the concentration area or specialization of choice. All required courses must be completed with a grade of B or better, and the student must write a research or thesis project. A committee of three faculty members (voting members of the Academic Senate) will guide the student and give final approval of the thesis. The committee will consist of an advisor (faculty member from the student’s department) who is willing to supervise the thesis project, and two other faculty members (one of which must be from the student’s department) who are willing to serve on the committee as readers of the thesis. An oral presentation of the thesis to the committee will be required. Seminar courses that have an “S” suffix (e.g., 209S) do not count toward degree requirements.

Plan II: Comprehensive Examination Plan. The student completes the required units as specified under the concentration area. Each course must be completed with a grade of B or better. Seminar courses that have an “S” suffix (e.g., 209S) do not count toward degree requirements. The student must take a comprehensive examination given by ICS faculty. The examination covers the core requirements.

ICS Concentration in Embedded Systems—M.S.

The goal of this program is to prepare students for challenges in developing future embedded systems. These future systems will further integrate communications, multimedia, and advanced processors with complex embedded and real-time software for automotive, medical, telecommunications, and many other application domains. Furthermore, embedded systems are becoming parallel, deploying multiprocessor systems-on-a-chip and parallel application software. An in-depth knowledge of the underlying scientific and engineering principles is required to understand these advances and to contribute productively to development of such systems. This program helps students master embedded system fundamentals, advanced computer architecture and compilers, networking, security, embedded, parallel and distributed software, and computer graphics in a sequence of courses and labs. Students also complete a large embedded systems project and may choose to write a Master’s thesis.

Required Courses

The following courses must be completed with a grade of B or better.

Select six of the following:

<table>
<thead>
<tr>
<th>List A</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 244</td>
</tr>
<tr>
<td>COMPSCI 247</td>
</tr>
<tr>
<td>COMPSCI 250A</td>
</tr>
<tr>
<td>COMPSCI 232</td>
</tr>
<tr>
<td>COMPSCI 203</td>
</tr>
<tr>
<td>COMPSCI 242</td>
</tr>
<tr>
<td>COMPSCI 250B</td>
</tr>
<tr>
<td>COMPSCI 230</td>
</tr>
<tr>
<td>COMPSCI 243</td>
</tr>
</tbody>
</table>

Select six additional courses in one of the following two ways:

1. For students pursuing the M.S. thesis option, two four-unit courses in Thesis Supervision (COMPSCI 298) plus four graduate courses taken from List A or the following List B
2. For all other students, six graduate courses taken from List A or the following List B

Select six additional courses in one of the following two ways:

<table>
<thead>
<tr>
<th>List B</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPSCI 241</td>
</tr>
<tr>
<td>COMPSCI 245</td>
</tr>
<tr>
<td>COMPSCI 246</td>
</tr>
<tr>
<td>COMPSCI 252</td>
</tr>
<tr>
<td>EECS 211</td>
</tr>
<tr>
<td>COMPSCI 211A</td>
</tr>
<tr>
<td>COMPSCI 248A/IN4MATX 241</td>
</tr>
<tr>
<td>IN4MATX 211</td>
</tr>
<tr>
<td>IN4MATX 235</td>
</tr>
<tr>
<td>COMPSCI 236</td>
</tr>
<tr>
<td>COMPSCI 267</td>
</tr>
<tr>
<td>COMPSCI 265</td>
</tr>
<tr>
<td>EECS 223</td>
</tr>
</tbody>
</table>

M.S. Students who do not have an undergraduate degree in Computer Science or equivalent must also take COMPSCI 260.
Comprehensive Examination or Thesis
Each student must either (1) pass a comprehensive examination administered by the Embedded Systems faculty; or (2) submit a thesis for approval by a three-person committee consisting of an advisor (who is an ICS Embedded Systems full-time faculty member) and two other full-time faculty members (one of which must be from ICS).

ICS Concentration in Informatics (INF)—M.S.
Informatics is the interdisciplinary study of the design, application, use, and impact of information technology. It goes beyond technical design to focus on the relationship between information system design and use in real-world settings. These investigations lead to new forms of system architecture, new approaches to system design and development, new means of information system implementation and deployment, and new models of interaction between technology and social, cultural, and organizational settings.

In the Donald Bren School of Information and Computer Sciences, Informatics is concerned with software architecture, software development, design and analysis, programming languages, ubiquitous computing, information retrieval and management, human-computer interaction, computer-supported cooperative work, and other topics that lie at the relationship between information technology design and use in social and organizational settings. Effective design requires an ability to analyze things from many different perspectives, including computer science, information science, organizational science, social science, and cognitive science. Relevant courses in those disciplines are therefore an integral part of the program and give this concentration a unique interdisciplinary flavor—which is imperative as the computing and information technology fields play such a pervasive role in our daily lives.

This degree program requires 48 units of coursework, including 24 units of core requirements, and 24 units of electives (of which up to 12 units may be used as independent study).

A. Complete the following required core courses:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN4MATX 261</td>
<td>Social Analysis of Computing</td>
</tr>
<tr>
<td>IN4MATX 231</td>
<td>User Interface Design and Evaluation</td>
</tr>
<tr>
<td>or IN4MATX 232</td>
<td>Research in Human-Centered Computing</td>
</tr>
<tr>
<td>IN4MATX 209S</td>
<td>Seminar in Informatics (twice, usually in first year)</td>
</tr>
</tbody>
</table>

B. Complete the following Research Methods core courses:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN4MATX 201</td>
<td>Research Methodology for Informatics</td>
</tr>
<tr>
<td>IN4MATX 203</td>
<td>Qualitative Research Methods in Information Systems</td>
</tr>
<tr>
<td>IN4MATX 205</td>
<td>Quantitative Research Methods in Information Systems</td>
</tr>
</tbody>
</table>

C. Select six electives in Informatics: ¹

¹ A set of six elective courses at the graduate level. The selection of courses should form a coherent educational plan to be approved by the student's faculty advisor or by the program director in the case that the student is not working with a faculty advisor. Although the courses may be chosen from any graduate level courses on campus, it is recommended that at least three be chosen from within the school of ICS. Students may use up to 12 units of independent study (IN4MATX 298/IN4MATX 299) as electives.

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)

Brigitte Baldi, Ph.D. Massachusetts Institute of Technology, Lecturer of Statistics

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)

Scott Bartell, Ph.D. University of California, Davis, Associate Professor of Program in Public Health; Environmental Health Sciences; Epidemiology; Social Ecology; Statistics

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)

Geoffrey C. Bowker, Ph.D. University of Melbourne, Professor of Informatics; Anthropology; Visual Studies (values in design, social studies of databases, science and technology studies)
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)

Carter Butts, Ph.D. Carnegie Mellon University, Professor of Sociology; Electrical Engineering and Computer Science; Statistics (mathematical sociology, social networks, quantitative methodology, human judgment and decision making, economic sociology)

Michael Carey, Ph.D. University of California, Berkeley, Donald Bren Professor of Information & Computer Sciences and Professor of Computer Science

Yunan Chen, Ph.D. Drexel University, Associate Professor of Informatics; Program in Public Health (medical informatics, human-computer interaction)

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

John Christopher Dobrian, Ph.D. University of California, San Diego, Professor of Music; Informatics

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)

Daniel L. Gillen, Ph.D. University of Washington, Professor of Statistics; Epidemiology; Program in Public Health

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

Judith Gregory, Ph.D. University of California, San Diego, Associate Adjunct Professor of Informatics (values in design, translational biomedical informatics, participatory design, design and emotion)

Vijay Gurbaxani, Ph.D. University of Rochester, Taco Bell Chair in Information Technology Management and Professor of Paul Merage School of Business; Informatics (economics of information systems management, impact of information technology on organization and market structure)

Stacey A. Hancock, Ph.D. Colorado State University, Lecturer with Potential Security of Employment of Statistics

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)

Gillian Hayes, Ph.D. Georgia Institute of Technology, Associate Professor of Informatics; Education (interactive and collaborative technology, human-computer interaction, computer-supported cooperative work, educational technology, ubiquitous computing)

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

Mizuko Ito, Ph.D. Stanford University, John D. and Catherine T. MacArthur Foundation Chair in Digital Media and Learning and Professor in Residence of Anthropology; Education; Informatics (ethnography, game studies, youth culture, learning sciences, online communities)

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

Ivan G. Jeliazkov, Ph.D. Washington University, Associate Professor of Economics; Statistics

Wesley O. Johnson, Ph.D. University of Minnesota, Professor of Statistics

James Jones, Ph.D. Georgia Institute of Technology, Associate Professor of Informatics (software engineering, software testing and analysis, debugging and fault localization, static and dynamic analysis, software visualization)

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

Cory P. Knobel, Ph.D. University of Michigan, Assistant Adjunct Professor of Informatics (interactive and collaborative technology, values in design, modes of knowledge representation, philosophy of science and technology)

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

Peter O. Krapp, Ph.D. University of California, Santa Barbara, Department Chair and Professor of Film and Media Studies; English; Informatics; Visual Studies (digital culture, media history, cultural memory)

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)

Gloria J. Mark, Ph.D. Columbia University, Professor of Informatics (computer-supported cooperative work, human-computer interaction)
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)

Melissa Mazmanian, Ph.D. Massachusetts Institute of Technology, Associate Professor of Informatics (computer-mediated communication, organization studies, information and communication technologies in practice, social response to emerging technologies, work/non-work negotiations in the information age)

Gopi Meenakshisundaram, 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)

Bonnie A. Nardi, Ph.D. University of California, Irvine, Professor of Informatics (computer-supported collaborative work, human-computer interaction, computer-mediated communication, user studies methods, activity theory, cultural responses to technology development)

Emily Navarro, Ph.D. University of California, Irvine, Lecturer of Informatics

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)

Gary Olson, Ph.D. Stanford University, Donald Bren Professor of Information & Computer Sciences and Professor of Informatics (interactive and collaborative technology, human-computer interaction, computer-supported cooperative work)

Judith Olson, Ph.D. University of Michigan, Donald Bren Professor of Information & Computer Sciences and Professor of Informatics; Paul Merage School of Business; Planning, Policy, and Design (interactive and collaborative technology, human-computer interaction, computer-supported cooperative work)

Hernando C. Ombao, Ph.D. University of Michigan, Professor of Statistics

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)

Simon G. Penny, M.F.A. Hong Kong University of Science and Technology, Professor of Art; Informatics (informatics, robotic sculpture, interactive environments, electronic media)

Kavita S. Philip, Ph.D. Cornell University, Associate Professor of History; Comparative Literature; Informatics (history of modern South Asia, science and technology, political ecology, critical theoretical studies of race, gender, colonialism, new media, and globalization)

Dale J. Poirier, Ph.D. University of Wisconsin-Madison, Professor of Economics; Statistics

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

David F. Redmiles, Ph.D. University of Colorado Boulder, Professor of Informatics (computer-supported cooperative work, human computer interaction, software engineering, globally distributed development teams, user interfaces, software tools)

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

Debra J. Richardson, Ph.D. University of Massachusetts, Professor of Informatics (software engineering, program testing, life-cycle validation, software environments)

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**

Thomas A. Standish, Ph.D. Carnegie Institute of Technology, **Professor Emeritus of Information and Computer Sciences** (software testing and analysis, software semantics and epistemology, programming and cognition, software comprehension)

Hal S. Stern, Ph.D. Stanford University, **Dean of the Donald Bren School of Information and Computer Sciences, Ted and Janice Smith Family Foundation Endowed Chair in Information and Computer Science, and Professor of Statistics; Cognitive Sciences**

Joshua Tanenbaum, M.A. Simon Fraser University, **Acting Assistant Professor of Informatics** (digital games and narrative, tangible and wearable interaction, maker and DIY culture, nonverbal communication and virtual worlds)

Richard N. Taylor, Ph.D. University of Colorado Boulder, **Professor Emeritus of Computer Science** (software engineering, user interfaces, environments, team support)

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

William M. Tomlinson, Ph.D. Massachusetts Institute of Technology, **Professor of Informatics; Education** (environmental informatics, educational technology, computer graphics/visualization/digital arts)

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

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

Jessica Utts, Ph.D. Pennsylvania State University, **Department Chair and Professor of Statistics**

André W. van der Hoek, Ph.D. University of Colorado Boulder, **Professor of Informatics** (software engineering)

Alexander Veidenbaum, Ph.D. Simon Fraser University, **Associate Professor of Informatics** (digital games and narrative, tangible and wearable interaction, maker and DIY culture, nonverbal communication and virtual worlds)

Richard N. Taylor, Ph.D. University of Colorado Boulder, **Professor Emeritus of Computer Science** (software engineering, user interfaces, environments, team support)

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

William M. Tomlinson, Ph.D. Massachusetts Institute of Technology, **Professor of Informatics; Education** (environmental informatics, educational technology, computer graphics/visualization/digital arts)

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

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

Jessica Utts, Ph.D. Pennsylvania State University, **Department Chair and Professor of Statistics**

André W. van der Hoek, Ph.D. University of Colorado Boulder, **Professor of Informatics** (software engineering)

Alexander Veidenbaum, Ph.D. Simon Fraser University, **Associate Professor of Informatics** (digital games and narrative, tangible and wearable interaction, maker and DIY culture, nonverbal communication and virtual worlds)

Richard N. Taylor, Ph.D. University of Colorado Boulder, **Professor Emeritus of Computer Science** (software engineering, user interfaces, environments, team support)

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

William M. Tomlinson, Ph.D. Massachusetts Institute of Technology, **Professor of Informatics; Education** (environmental informatics, educational technology, computer graphics/visualization/digital arts)

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

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

Jessica Utts, Ph.D. Pennsylvania State University, **Department Chair and Professor of Statistics**

André W. van der Hoek, Ph.D. University of Colorado Boulder, **Professor of Informatics** (software engineering)

Alexander Veidenbaum, Ph.D. Simon Fraser University, **Associate Professor of Informatics** (digital games and narrative, tangible and wearable interaction, maker and DIY culture, nonverbal communication and virtual worlds)

Richard N. Taylor, Ph.D. University of Colorado Boulder, **Professor Emeritus of Computer Science** (software engineering, user interfaces, environments, team support)

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

William M. Tomlinson, Ph.D. Massachusetts Institute of Technology, **Professor of Informatics; Education** (environmental informatics, educational technology, computer graphics/visualization/digital arts)

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

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

Jessica Utts, Ph.D. Pennsylvania State University, **Department Chair and Professor of Statistics**

André W. van der Hoek, Ph.D. University of Colorado Boulder, **Professor of Informatics** (software engineering)

Alexander Veidenbaum, Ph.D. Simon Fraser University, **Associate Professor of Informatics** (digital games and narrative, tangible and wearable interaction, maker and DIY culture, nonverbal communication and virtual worlds)

Richard N. Taylor, Ph.D. University of Colorado Boulder, **Professor Emeritus of Computer Science** (software engineering, user interfaces, environments, team support)

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

William M. Tomlinson, Ph.D. Massachusetts Institute of Technology, **Professor of Informatics; Education** (environmental informatics, educational technology, computer graphics/visualization/digital arts)

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

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

Jessica Utts, Ph.D. Pennsylvania State University, **Department Chair and Professor of Statistics**

André W. van der Hoek, Ph.D. University of Colorado Boulder, **Professor of Informatics** (software engineering)

Alexander Veidenbaum, Ph.D. Simon Fraser University, **Associate Professor of Informatics** (digital games and narrative, tangible and wearable interaction, maker and DIY culture, nonverbal communication and virtual worlds)

Richard N. Taylor, Ph.D. University of Colorado Boulder, **Professor Emeritus of Computer Science** (software engineering, user interfaces, environments, team support)

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

William M. Tomlinson, Ph.D. Massachusetts Institute of Technology, **Professor of Informatics; Education** (environmental informatics, educational technology, computer graphics/visualization/digital arts)

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

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

Jessica Utts, Ph.D. Pennsylvania State University, **Department Chair and Professor of Statistics**

André W. van der Hoek, Ph.D. University of Colorado Boulder, **Professor of Informatics** (software engineering)

Alexander Veidenbaum, Ph.D. Simon Fraser University, **Associate Professor of Informatics** (digital games and narrative, tangible and wearable interaction, maker and DIY culture, nonverbal communication and virtual worlds)

Richard N. Taylor, Ph.D. University of Colorado Boulder, **Professor Emeritus of Computer Science** (software engineering, user interfaces, environments, team support)

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

William M. Tomlinson, Ph.D. Massachusetts Institute of Technology, **Professor of Informatics; Education** (environmental informatics, educational technology, computer graphics/visualization/digital arts)

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

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

Jessica Utts, Ph.D. Pennsylvania State University, **Department Chair and Professor of Statistics**

André W. van der Hoek, Ph.D. University of Colorado Boulder, **Professor of Informatics** (software engineering)

Alexander Veidenbaum, Ph.D. Simon Fraser University, **Associate Professor of Informatics** (digital games and narrative, tangible and wearable interaction, maker and DIY culture, nonverbal communication and virtual worlds)

Richard N. Taylor, Ph.D. University of Colorado Boulder, **Professor Emeritus of Computer Science** (software engineering, user interfaces, environments, team support)

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

William M. Tomlinson, Ph.D. Massachusetts Institute of Technology, **Professor of Informatics; Education** (environmental informatics, educational technology, computer graphics/visualization/digital arts)

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

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

Jessica Utts, Ph.D. Pennsylvania State University, **Department Chair and Professor of Statistics**

André W. van der Hoek, Ph.D. University of Colorado Boulder, **Professor of Informatics** (software engineering)

Alexander Veidenbaum, Ph.D. Simon Fraser University, **Associate Professor of Informatics** (digital games and narrative, tangible and wearable interaction, maker and DIY culture, nonverbal communication and virtual worlds)

Richard N. Taylor, Ph.D. University of Colorado Boulder, **Professor Emeritus of Computer Science** (software engineering, user interfaces, environments, team support)

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

William M. Tomlinson, Ph.D. Massachusetts Institute of Technology, **Professor of Informatics; Education** (environmental informatics, educational technology, computer graphics/visualization/digital arts)

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

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

Jessica Utts, Ph.D. Pennsylvania State University, **Department Chair and Professor of Statistics**

André W. van der Hoek, Ph.D. University of Colorado Boulder, **Professor of Informatics** (software engineering)

Alexander Veidenbaum, Ph.D. Simon Fraser University, **Associate Professor of Informatics** (digital games and narrative, tangible and wearable interaction, maker and DIY culture, nonverbal communication and virtual worlds)

Richard N. Taylor, Ph.D. University of Colorado Boulder, **Professor Emeritus of Computer Science** (software engineering, user interfaces, environments, team support)
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 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 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 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 33 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 algebra, 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. IN4MATX 45 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 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 6B 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 161. 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.

Same as CSE 161.
COMPSCI 162. 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 163. 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 164. 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 165. 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 167. 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 168. 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.
Informatics Courses

IN4MATX 12. Barter to Bitcoin: Society, Technology and the Future of Money. 4 Units.
Digital money has captured the broad imagination of speculators, coders, regulators, criminals and the mass media. Course puts this change in context: how do we understand money as a social, political and technological phenomenon? Course may be offered online.

Same as SOC SCI 11A.

(II, III)

IN4MATX 41. Informatics Core Course I. 6 Units.
Fundamental concepts of computer software design and construction. Data, algorithms, functions, and abstractions. Overview of computer systems: data representation, architectural components, operating systems, networks. Introduction to information systems: parties involved, architectural alternatives, usability, organizational and social concerns.

Restriction: IN4MATX 41 may not be taken for credit if taken after I&C SCI 22 or CSE 22.

(II, Vb)

IN4MATX 42. Informatics Core Course II. 6 Units.
Alternative data structure implementations; analysis of time and space efficiency. Object-oriented programming concepts and techniques: classes, objects, inheritance, interfaces. Formal languages and automata. Problem modeling and design tradeoffs.

Prerequisite: IN4MATX 41. IN4MATX 41 with a grade of C or better.

Overlaps with I&C SCI 32, CSE 42, I&C SCI 33, CSE 43, I&C SCI 22, CSE 22.

(II, Vb)

IN4MATX 43. Introduction to Software Engineering. 4 Units.
Concepts, methods, and current practice of software engineering. Large-scale software production, software life cycle models, principles and techniques for each stage of development.

Overlaps with I&C SCI 52, I&C SCI 105.

IN4MATX 44. Seminar in Informatics Research Topics. 2 Units.
Introduction to current research topics in Informatics. Various faculty members present current research and relate it to the course content of the Informatics degree program.

Grading Option: Pass/no pass only.

IN4MATX 45. Patterns of Software Construction. 4 Units.
Building software applications; reusing and integrating components; designing for reuse. Effective use of libraries and APIs, file and network I/O, creation of user interfaces.

Prerequisite: IN4MATX 42 or I&C SCI 22 or CSE 22 or I&C SCI H22. I&C SCI H22 with a grade of C or better.

(Vb)

IN4MATX 101. 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 CSE 141, COMPSCI 141.

IN4MATX 102. Concepts of Programming Language II. 4 Units.
In-depth study of major programming paradigms: imperative, functional, declarative, object-oriented, and aspect-oriented. Understanding the role of programming languages in software development and the suitability of languages in context. Domain-specific languages. Designing new languages for better software development support.

Prerequisite: IN4MATX 101 or COMPSCI 141 or CSE 141. CSE 141 with a grade of C or better.
IN4MATX 113. Requirements Analysis and Engineering. 4 Units.
Aims to equip students to develop techniques of software-intensive systems through successful requirements analysis techniques and requirements engineering. Students learn systematic process of developing requirements through cooperative problem analysis, representation, and validation.
Prerequisite: IN4MATX 42 or I&C SCI 22 or CSE 22 or I&C SCI 33 or CSE 43 and (IN4MATX 43 or I&C SCI 52). IN4MATX 42 with a grade of C or better. I&C SCI 22 with a grade of C or better. CSE 22 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. IN4MATX 43 with a grade of C or better. I&C SCI 52 with a grade of C or better.

IN4MATX 115. Software Testing, Analysis, and Quality Assurance. 4 Units.
Preparation for developing high-quality software through successful verification and validation techniques. Fundamental principles of software testing, implementing software testing practices, ensuring the thoroughness of testing to gain confidence in the correctness of the software.
Prerequisite: (I&C SCI 45J or I&C SCI 45C or I&C SCI 65 or I&C SCI 23 or CSE 23 or I&C SCI 46 or CSE 46 or IN4MATX 45) and (IN4MATX 43 or I&C SCI 52). I&C SCI 45J with a grade of C or better. I&C SCI 45C with a grade of C or better. I&C SCI 65 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. IN4MATX 45 with a grade of C or better. IN4MATX 43 with a grade of C or better. I&C SCI 52 with a grade of C or better.

IN4MATX 117. Project in Software System Design. 4 Units.
Specification, design, construction, testing, and documentation of a complete software system. Special emphasis on the need for and use of teamwork, careful planning, and other techniques for working with large systems.
Prerequisite: (IN4MATX 43 or I&C SCI 52) and (I&C SCI 33 or CSE 43 or I&C SCI 22 or CSE 22 or IN4MATX 42) I&C SCI 52 with a grade of a C or better. IN4MATX 42 with a grade of C or better.
Restriction: Upper-division students only.

IN4MATX 121. Software Design: Applications. 4 Units.
Introduction to application design: designing the overall functionality of a software application. Topics include general design theory, software design theory, and software architecture. Includes practice in designing and case studies of existing designs.
Prerequisite: IN4MATX 45 or I&C SCI 23 or CSE 23 or I&C SCI 33 or CSE 43, IN4MATX 45 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 33 with a grade of C or better. CSE 43 with a grade of C or better.
Restriction: Upper-division students only.

IN4MATX 122. Software Design: Structure and Implementation. 4 Units.
Introduction to implementation design: designing the internals of a software application. Topics include design aesthetics, design implementation, design recovery, design patterns, and component reuse. Includes practice in designing and case studies of existing designs.
Prerequisite: (I&C SCI 45J or I&C SCI 46 or IN4MATX 45) and (IN4MATX 101 or COMPSCI 141 or CSE 141).

IN4MATX 123. Software Architecture. 4 Units.
Prepares students to engineer well-structured software systems. Students learn a wide range of software architectural styles, architectural platforms that provide standard services to applications, and formal architecture description languages.
Prerequisite: IN4MATX 122 or ((IN4MATX 101 or COMPSCI 141 or CSE 141) and IN4MATX 113).

IN4MATX 124. 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 COMPSCI 137.
Restriction: Upper-division students only.

IN4MATX 125. 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 COMPSCI 113.
IN4MATX 131. Human Computer Interaction. 4 Units.
Basic principles of human-computer interaction (HCI). Introduces students to user interface design techniques, design guidelines, and usability testing. Students gain the ability to design and evaluate user interfaces and become familiar with some of the outstanding research problems in HCI.

Prerequisite: IN4MATX 41 or I&C SCI 10 or I&C SCI 21 or CSE 21 or I&C SCI H21 or I&C SCI 31 or CS 41 or ENGR 10 or ENGRMAE 10 or EECS 10. IN4MATX 41 with a grade of C or better. I&C SCI 10 with a grade of C or better. I&C SCI 21 with a grade of C or better. CSE 21 with a grade of C or better. I&C SCI H21 with a grade of C or better. I&C SCI 31 with a grade of C or better. CS 41 with a grade of C or better. ENGR 10 with a grade of C or better. ENGRMAE 10 with a grade of C or better. EECS 10 with a grade of C or better.

Restriction: Upper-division students only.

IN4MATX 132. Project in Human-Computer Interaction Requirements and Evaluation. 4 Units.
Students undertake significant projects in the elicitation and specification of HCI requirements and the thorough evaluation of user interfaces.

Prerequisite: IN4MATX 131.

IN4MATX 133. User Interaction Software. 4 Units.
Introduction to human-computer interaction programming. Emphasis on current tools, standards, methodologies for implementing effective interaction designs. Widget toolkits, Web interface programming, geo-spatial and map interfaces, mobile phone interfaces.

Prerequisite: I&C SCI 45J. I&C SCI 45J with a grade of C or better.

IN4MATX 134. Project in User Interaction Software. 4 Units.
Students complete an end-to-end user interface programming project based on an iterative design paradigm. Topics may include requirements brainstorming, paper prototyping, iterative development, cognitive walk-through, quantitative evaluation, and acceptance testing. Materials fee.

Prerequisite: IN4MATX 131 and IN4MATX 133.

IN4MATX 141. 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 COMPSCI 121.

IN4MATX 143. Information Visualization. 4 Units.
Introduction to interactive visual interfaces for large datasets, and to principles of human visual perception and human computer interaction that inform their design. Various applications for data analysis and monitoring are discussed.

Prerequisite: IN4MATX 131 or I&C SCI 52 or (IN4MATX 43 and (I&C SCI 31 or CSE 41 or I&C SCI 21 or CSE 21 or IN4MATX 41)). IN4MATX 131 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. I&C SCI 31 with a grade of C or better. CSE 41 with a grade of C or better. I&C SCI 21 with a grade of C or better. CSE 21 with a grade of C or better. IN4MATX 41 with a grade of C or better.

IN4MATX 148. Project in Ubiquitous Computing. 4 Units.
Introduction to ubiquitous computing research methods, tools, and techniques. Prototyping, design, and evaluation of physical computing applications, smart environments, embedded systems, and future computing scenarios. Includes hands-on in-class laboratory exercises. Materials fee.

Prerequisite: I&C SCI 10 or I&C SCI 21 or CSE 21 or I&C SCI 31 or CSE 41 or IN4MATX 41. I&C SCI 10 with a grade of C or better. I&C SCI 21 with a grade of C or better. CSE 21 with a grade of C or better. I&C SCI 31 with a grade of C or better. CSE 41 with a grade of C or better. IN4MATX 41 with a grade of C or better.

Restriction: Upper-division students only.

IN4MATX 151. Project Management. 4 Units.
Introduces theoretical and practical aspects of project management. Topics include organizational theory, group behavior, project management skills, case studies, personal and group productivity tools, management of distributed work, stakeholders, consultants, and knowledge management. Students do a project exercise.

Prerequisite: IN4MATX 43 or I&C SCI 52. I&C SCI 52 with a grade of C or better.

Restriction: Upper-division students only.
IN4MATX 153. Computer Supported Cooperative Work. 4 Units.
Introduces concepts and principles of collaborative systems. Topics may include shared workspaces, group interaction, workflow, architectures, interaction between social and technical features of group work, and examples of collaborative systems used in real-world settings. Students develop a simple collaborative application.

Prerequisite: (IN4MATX 161 or I&C SCI 52 or IN4MATX 43) and (I&C SCI 31 or CSE 41 or I&C SCI 21 or CSE 21 or IN4MATX 41). I&C SCI 52 with a grade of C or better. I&C SCI 31 with a grade of C or better. CSE 41 with a grade of C or better. I&C SCI 21 with a grade of C or better. CSE 21 with a grade of C or better. IN4MATX 41 with a grade of C or better.

IN4MATX 161. Social Analysis of Computing. 4 Units.
Introduction of computing as a social process. Examines the social opportunities and problems raised by new information technologies, and the consequences of different ways of organizing. Topics include computing and work life, privacy, virtual communities, productivity paradox, systems risks.

Prerequisite: IN4MATX 41 or I&C SCI 10 or I&C SCI 21 or CSE 21 or I&C SCI 31 or CSE 41 or ENGR 10 or EECS 10 or ENGRMAE 10. IN4MATX 41 with a grade of C or better. I&C SCI 10 with a grade of C or better. I&C SCI 21 with a grade of C or better. CSE 21 with a grade of C or better. I&C SCI 31 with a grade of C or better. CSE 41 with a grade of C or better. ENGR 10 with a grade of C or better. EECS 10 with a grade of C or better. ENGRMAE 10 with a grade of C or better. Satisfactory completion of the Lower-Division Writing requirement.

IN4MATX 162W. Organizational Information Systems. 4 Units.
Introduction to role of information systems in organizations, components and structure of organizational information systems, and techniques used in information systems analysis, design, and implementation.

Prerequisite: IN4MATX 161. Satisfactory completion of the Lower-Division Writing requirement.

(INB)

IN4MATX 163. Project in the Social and Organizational Impacts of Computing . 4 Units.
Students undertake projects intended to gather and analyze data from situations in which computers are used, organize and conduct experiments intended to test hypotheses about impacts, and explore the application of concepts learned in previous courses.

Prerequisite: IN4MATX 162.

IN4MATX 171. Introduction to Medical Informatics. 4 Units.
Broad overview of medical informatics for students with varied backgrounds. Electronic medical records, online resources, mobile technologies, patient safety, and computational design. Legal, ethical, and public policy issues. Health systems management. Evaluation and fieldwork for health systems.
Same as PUBHLTH 105.
Restriction: Upper-division students only.

IN4MATX 172. Project in Health Informatics. 4 Units.
Students undertake significant quarter-long projects related to health informatics. Topics may include field evaluations of health care technologies, prototypes, iterative design, and system implementations.

Prerequisite: PUBHLTH 105 or IN4MATX 171.
Same as PUBHLTH 106.

IN4MATX 190. Special Topics in Informatics. 4 Units.
Studies in selected areas of informatics. Topics addressed vary each quarter.

Prerequisite: Prerequisites vary.
Repeatability: Unlimited as topics vary.

IN4MATX 191A. Senior Design Project. 4 Units.
Group supervised project in which students analyze, specify, design, construct, evaluate, and adapt a significant information processing system. Topics include team management, professional ethics, and systems analysis.

Prerequisite: IN4MATX 121 and IN4MATX 131 and IN4MATX 151 and IN4MATX 161.
Grading Option: In progress only.
Restriction: Seniors only.
IN4MATX 191B. Senior Design Project. 4 Units.
Group supervised project in which students analyze, specify, design, construct, evaluate, and adapt a significant information processing system. Topics include team management, professional ethics, and systems analysis.
Prerequisite: IN4MATX 191A.
Restriction: Seniors only.

IN4MATX H198. Honors Research. 4 Units.
Directed independent research in Informatics for honors students.
Prerequisite: Satisfactory completion of the Lower-Division Writing requirement.
Repeatability: May be repeated for credit unlimited times.
Restriction: Bren School of ICS Honors Program or the Campuswide Honors Program students only.

IN4MATX 199. Individual Study. 2-5 Units.
Individual research or investigation under the direction of an individual faculty member.
Repeatability: May be repeated for credit unlimited times.

IN4MATX 201. Research Methodology for Informatics. 4 Units.
Introduction to strategies and idioms of research in Informatics. Includes examination of issues in scientific inquiry, qualitative and quantitative methods, and research design. Both classic texts and contemporary research literature are read and analyzed.

IN4MATX 203. Qualitative Research Methods in Information Systems. 4 Units.
Introduction to qualitative research methods used to study computerization and information systems, such as open-ended interviewing, participant observation, and ethnography. Studies of the methods in practice through examination of research literature.
Prerequisite: IN4MATX 261 or IN4MATX 251.

IN4MATX 205. Quantitative Research Methods in Information Systems. 4 Units.
Quantitative research methods used to study computerization and information systems. Design of instruments, sampling, sample sizes, and data analysis. Validity and reliability. Longitudinal versus cross-sectional designs. Analysis of secondary data. Studies of the methods through examination of research literature.
Prerequisite: (IN4MATX 251 or IN4MATX 261) and basic knowledge of elementary statistics.

IN4MATX 207S. Doctoral Seminar on Research and Writing. 2 Units.
Doctoral seminar centered on original research and writing. Provides a chance for doctoral students at all levels to present original work, brainstorm ongoing issues, and learn to provide and receive critical feedback from peers.
Grading Option: Satisfactory/unsatisfactory only.
Repeatability: May be repeated for credit unlimited times.

IN4MATX 209S. Seminar in Informatics. 2 Units.
Current research and research trends in informatics. Forum for presentation and criticism by students of research work in progress.
Repeatability: Unlimited as topics vary.

IN4MATX 211. Software Engineering. 4 Units.
Study of the concepts, methods, and tools for the analysis, design, construction, and measurement of complex software-intensive systems. Underlying principles emphasized. State-of-the-art software engineering and promising research areas covered, including project management.

IN4MATX 212. 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 COMPSCI 253.
IN4MATX 213. Requirements Engineering and Specification. 4 Units.
Study of rigorous techniques in requirements engineering - requirements definition phase of software development - with focus on modeling and specification. Topics include notations and models for requirements specification; and methods, tools and processes for software requirements elicitation, representation, analysis.

Restriction: Graduate students only.

IN4MATX 215. Software Analysis and Testing. 4 Units.
Studies techniques for developing confidence in software from traditional testing schemes to integrated, multitechnique analytic approaches. Considers strengths and weaknesses and explores opportunities for synergistic technique application. Emphasis is on approaches integrated into the software process.

IN4MATX 217. Software Processes. 4 Units.
Explores vehicles for modeling, coding, and analyzing software processes. Considers integration of software process programming approaches with traditional management issues. Explores the use of software process execution as a vehicle for effective integration of tools into environments.

IN4MATX 219. Software Environments. 4 Units.
Study of the requirements, concepts, and architectures of comprehensive, integrated, software development and maintenance environments. Major topics include process support, object management, communication, interoperability, measurement, analysis, and user interfaces in the environment context.

IN4MATX 221. Software Architecture. 4 Units.
Study of the concepts, representation techniques, development methods, and tools for architecture-centric software engineering. Topics include domain-specific software architectures, architectural styles, architecture description languages, software connectors, and dynamism in architectures.

IN4MATX 223. Applied Software Design Techniques. 4 Units.
Study of concepts, representations, techniques, and case studies in structuring software systems, with an emphasis on design considerations. Topics include static and dynamic system structure, data models, abstractions, naming, protocols and application programmer interfaces.

IN4MATX 225. 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 COMPSCI 221.

Restriction: Graduate students only.

IN4MATX 231. User Interface Design and Evaluation. 4 Units.
Introduction to the design and evaluation of user interfaces, with an emphasis on methodology. Cognitive principles, design life cycle, on-line and off-line prototyping techniques. Toolkits and architectures for interactive systems. Evaluation techniques, including heuristic and laboratory methods.

IN4MATX 232. Research in Human-Centered Computing. 4 Units.
Introduction to contemporary topics in human-centered computing, including methods, technologies, design, and evaluation. Emerging application domains and their challenges to traditional research methods. Advanced architectures and technologies. Critical issues.

Prerequisite: Some familiarity with HCI principles.

IN4MATX 233. Intelligent User Interfaces. 4 Units.
Explores example software systems and their underlying concepts that leverage computing to empower and augment human individuals in their activities. Topics span the fields of user interface design, human-computer interaction, software engineering, and cognitive computing.

Prerequisite: COMPSCI 171.

IN4MATX 235. Advanced User Interface Architecture. 4 Units.
Architectural concerns in advanced interactive systems. The design of current and emerging platforms for novel interactive systems. Paradigms such as constraint-based programming, multimodal interaction, and perceptual user interfaces for individual, distributed, and ubiquitous applications.

IN4MATX 241. 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 COMPSCI 248A.
IN4MATX 242. 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 COMPSCI 248B.

IN4MATX 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 COMPSCI 244.

IN4MATX 251. Computer-Supported Cooperative Work. 4 Units.
The role of information systems in supporting work in groups and organizations. Examines various technologies designed to support communication, information sharing, and coordination. Focuses on behavioral and social aspects of designing and using group support technologies.

IN4MATX 261. Social Analysis of Computing. 4 Units.
The social and economic impacts of computing and information technologies on groups, organizations, and society. Topics include computerization and changes in the character of work, social control and privacy, electronic communities, and risks of safety-critical systems to people.

IN4MATX 263. Computerization, Work, and Organizations. 4 Units.
Selected topics in the influence of computerization and information systems in transforming work and organizations. Theories of organization and organizational change. Processes by which diverse information technologies influence changes in work and organizations over short and long time periods.
Prerequisite: IN4MATX 251 or IN4MATX 261.

IN4MATX 265. Theories of Computerization and Information Systems. 4 Units.
Social and economic conceptions of information technology. Macrosocial and economic conditions that foster changes in information technologies. Social construction of information and computer technology in professional worlds. Theories of information technology and large-scale social change.
Prerequisite: IN4MATX 251 or IN4MATX 261.

IN4MATX 267. Digital Media and Society. 4 Units.
Selected topics in the technological and social aspects of online interactions, and policy including online games, social media, electronic activism, e-commerce, and digital libraries. Media-theoretic approaches to digital technology. Architectures, infrastructure considerations, and their consequences.
Prerequisite: IN4MATX 251 or IN4MATX 261.

IN4MATX 269. Computer Law. 4 Units.
Restriction: Graduate students only.

IN4MATX 273. Information Technology in Global Sustainability. 4 Units.
Explores the relationship between recent developments in information technology and the global transition to sustainability. Topics include the role of IT systems in the provision of human needs and wants (e.g., smart grids, food systems, and other IT-enabled infrastructure).
Restriction: Graduate students only.

IN4MATX 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.
IN4MATX 291S. Literature Survey in Software Engineering. 2 Units.
Reading and analysis of relevant literature in Software Engineering under the direction of a faculty member.
Repeatability: May be repeated for credit unlimited times.

IN4MATX 295. Special Topics in Informatics. 4 Units.
Studies in selected areas of informatics. Topics addressed vary each quarter.
Repeatability: Unlimited as topics vary.
Restriction: Graduate students only.

IN4MATX 296. 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.

IN4MATX 299. Individual Study. 2-12 Units.
Individual research or investigation under the direction of an individual faculty member.
Repeatability: May be repeated for credit unlimited times.
Restriction: Graduate students only.

Information and Computer Science Courses

I&C SCI 3. Internet Technologies and their Social Impact. 4 Units.
Examines current Internet technologies and social implications at the individual, group, and societal level. Blogs, wikis, sharing of video, photos, and music, e-commerce, social networking, gaming, and virtual environments. Issues include privacy, trust, identity, reputation, governance, copyright, and malicious behavior.

(Ill)

I&C SCI 4. Human Factors for the Web. 4 Units.
Restriction: May not be taken for credit after IN4MATX 131.

(Ill)

I&C SCI 5. Global Disruption and Information Technology. 4 Units.
Explores how new forms of information technology may support transition to a sustainable civilization. Topics include design and implementation of IT systems, science of global change, online community building, and “green IT”. Activities involve reading, writing, discussion, and final project. Course may be offered online.

(Ill)

I&C SCI 6B. Boolean Algebra and Logic. 4 Units.
Relations and their properties; Boolean algebras, formal languages; finite automata.
Prerequisite: High school mathematics through trigonometry.

(Vb)

I&C SCI 6D. Discrete Mathematics for Computer Science. 4 Units.
Covers essential tools from discrete mathematics used in computer science with an emphasis on the process of abstracting computational problems and analyzing them mathematically. Topics include mathematical induction, combinatorics, and recurrence relations.
Prerequisite: I&C SCI 6B.

(Vb)
I&C SCI 6N. Computational Linear Algebra. 4 Units.
Matrices and linear transformations, systems of linear equations, determinants, linear vector spaces, eigenvalues and eigenvectors, orthogonal matrices, diagonalization, and least squares. Topics will be taught primarily from an algorithmic perspective, including computational solutions, applications, and numerical error analysis.

Overlaps with MATH 6G, MATH 3A.
(Ii, Vb)

I&C SCI 7. Introducing Modern Computational Tools. 4 Units.
A unified look at a spectrum of modern tools for building, solving, and analyzing simple computational models (deterministic and random) in diverse subject areas. Tools include those for numeric/symbolic computation, and those for acquiring, organizing, translating, processing, and displaying information.

(Va)

I&C SCI 8. Practical Computer Security. 4 Units.
Principles of practical computer security to enable students to defend themselves against malicious threats. Firewalls, anti-virus, secure setup of a wireless access point. Cryptography basics and its application. Embedded devices and related security issues.

(Ii)

I&C SCI 10. How Computers Work. 4 Units.
Introduction to digital computer and communication systems. Capabilities and limitations of information technology. Representing information in digital form. Overview of computer organization, Internet, operating systems, software. Human-computer interaction and social impact.

Restriction: May not be taken for credit after I&C SCI 51, I&C SCI 52, I&C SCI 105, or IN4MATX 43.
(Ii)

I&C SCI 11. The Internet and Public Policy. 4 Units.
How the Internet works. Current public policy issues concerning the Internet. Introductory economics. Communications law. Interactions between information technology, economics, and law. Case studies about Internet and communications policy.

Same as ECON 11.
(Ii or III )

I&C SCI 21. Introduction to Computer Science I . 6 Units.
Introduces fundamental concepts related to computer software design and construction. Develops initial design and programming skills using a high-level language. Fundamental concepts of control structures, data structures, and object-oriented programming.

Same as CSE 21.
Overlaps with I&C SCI H21, I&C SCI 31, EECS 10, EECS 12, ENGRMAE 10.

Restriction: CSE 21 or I&C SCI 21 may not be taken for credit if taken after IN4MATX 42.
(Ii, Vb)

I&C SCI H21. Honors Introduction to Computer Science I. 6 Units.
Introduces fundamental concepts of computer software design and construction. Develops initial design and programming skills using a high-level language. Fundamental concepts of control structures, data structures, functional and object-oriented programming. Introduces topics in computer organization and social impact of technology.

Overlaps with I&C SCI 21, I&C SCI 31, CSE 21, EECS 10, EECS 12, ENGR 10.

Restriction: Information and Computer Science, Computer Science Engineering, and Computer Science majors only. Campuswide Honors Program students only. May not be taken for credit after IN4MATX 42.
(Ii, Vb)
I&C SCI 22. Introduction to Computer Science II. 6 Units.
Abstract behavior of classic data structures (stacks, queues, sorted and unsorted maps), alternative implementations, analysis of time, and space efficiency.

Prerequisite: CSE 21 or I&C SCI 21 or I&C SCI H21. CSE 21 with a grade of C or better. I&C SCI 21 with a grade of C or better. I&C SCI H21 with a grade of C or better.

Same as CSE 22.
Overlaps with I&C SCI H22, CSE 22, CSE 42, I&C SCI 32, CSE 43.

(II, Vb)

I&C SCI H22. Honors Introduction to Computer Science II. 6 Units.
Abstract behavior of classic data structures (stacks, queues, sorted and unsorted maps), alternative implementations. Recursion. Mathematical analysis of time and space efficiency, program analysis and correctness, system design techniques, programming paradigms.

Prerequisite: I&C SCI H21 or I&C SCI 21 or CSE 21. I&C SCI H21 with a grade of B- or better. I&C SCI 21 with a grade of A or better. CSE 21 with a grade of A or better.

Overlaps with IN4MATX 42, CSE 22, CSE 42, I&C SCI 22, I&C SCI 33, CSE 43.

(II, Vb)

I&C SCI H23. Honors Introduction to Computer Science III. 4 Units.
Builds on ICS H22 with respect to mathematical tools and analysis. Focuses on fundamental algorithms in computer science, basic data structures for primary and secondary memory, storage allocation and management techniques, data description, and design techniques.

Prerequisite: I&C SCI H22 or I&C SCI 22 or IN4MATX 42. I&C SCI H22 with a grade of B- or better. I&C SCI 22 with a grade of A or better CSE 22 with a grade of A or better. IN4MATX 42 with a grade of A or better.

Overlaps with I&C SCI 46, CSE 46.

(Vb)

I&C SCI 31. Introduction to Programming. 4 Units.
Introduction to fundamental concepts and techniques for writing software in a high-level programming language. Covers the syntax and semantics of data types, expressions, exceptions, control structures, input/output, methods, classes, and pragmatics of programming.

Same as CSE 41.
Overlaps with I&C SCI 21, CSE 21, I&C SCI H21, EECS 10, EECS 12.

(II, Vb)

I&C SCI 32. Programming with Software Libraries. 4 Units.
Construction of programs for problems and computing environments more varied than in CSE41. Using library modules for applications such as graphics, sound, GUI, database, Web, and network programming. Language features beyond those in CSE41 are introduced as needed.

Prerequisite: I&C SCI 31 or CSE 41. I&C SCI 31 with a grade of C or better. CSE 41 with a grade of C or better.

Same as CSE 42.
Overlaps with I&C SCI 22, CSE 22, I&C SCI H22, IN4MATX 42.

(II and (VA or VB) )

I&C SCI 33. Intermediate Programming. 4 Units.
Intermediate-level language features and programming concepts for larger, more complex, higher-quality software. Functional programming, name spaces, modules, class protocols, inheritance, iterators, generators, operator overloading, reflection. Analysis of time and space efficiency.

Prerequisite: I&C SCI 32 or CSE 42. I&C SCI 32 with a grade of C or better. CSE 42 with a grade of C or better.

Same as CSE 43.
Overlaps with I&C SCI 33, I&C SCI 22, CSE 22, I&C SCI H22, IN4MATX 42.

(II, Vb)
I&C SCI 45C. Programming in C/C++ as a Second Language. 4 Units.

Prerequisite: I&C SCI 22 or CSE 22 or IN4MATX 42 or I&C SCI 33 or CSE 43. I&C SCI 22 with a grade of C or better. CSE 22 with a grade of C or better. IN4MATX 42 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 CSE 45C.

I&C SCI 45J. Programming in Java as a Second Language. 4 Units.
An introduction to the lexical, syntactic, semantic, and pragmatic characteristics of the Java language for experienced programmers. Emphasis on object-oriented programming, using standard libraries, and programming with automatic garbage collection.

Prerequisite: I&C SCI 33 or CSE 43. I&C SCI 33 with a grade of C or better. CSE 43 with a grade of C or better.

Overlaps with I&C SCI 22, CSE 22, I&C SCI 23, CSE 23, IN4MATX 45.

Restriction: I&C SCI 45J may not be taken for credit after I&C SCI 22, CSE 22, I&C SCI 23, CSE 23, or IN4MATX 45.

I&C SCI 46. Data Structure Implementation and Analysis. 4 Units.
Focuses on implementation and mathematical analysis of fundamental data structures and algorithms. Covers storage allocation and memory management techniques.

Prerequisite: CSE 45C or I&C SCI 45C. CSE 45C with a grade of C or better. I&C SCI 45C with a grade of C or better.

Same as CSE 46.

Overlaps with I&C SCI H23.

(Vb)

I&C SCI 51. Introductory Computer Organization. 6 Units.
Multilevel view of system hardware and software. Operation and interconnection of hardware elements. Instruction sets and addressing modes. Virtual memory and operating systems. Laboratory work using low-level programming languages. Course may be offered online.

Prerequisite: (I&C SCI 21 or CSE 21 or I&C SCI 31 or CSE 41 or IN4MATX 42) and I&C SCI 6B. IN4MATX 42 with a grade of C or better.

(II)

I&C SCI 52. Introduction to Software Engineering. 6 Units.
Introduction to concepts, methods, and current practice of software engineering. Study of large-scale software production; software life cycle models as an organizing structure; principles and techniques appropriate for each stage of production. Laboratory work involves a project illustrating these elements.

Prerequisite: I&C SCI 23. I&C SCI 23 with a grade of C or better.

Overlaps with I&C SCI 105, IN4MATX 43.

I&C SCI 53. Principles in System Design. 4 Units.
Principles and practice of engineering of computer software and hardware systems. Topics include techniques for controlling complexity; strong modularity using client-server design, virtual memory, and threads; networks; coordination of parallel activities; security and encryption; and performance optimizations.

Corequisite: I&C SCI 53L.

Prerequisite: I&C SCI 51.

I&C SCI 53L. Principles in System Design Library. 2 Units.
Required laboratory section and co-requisite for I&C SCI 53.

Corequisite: I&C SCI 53.

Prerequisite: I&C SCI 51.
I&C SCI 60. Computer Games and Society. 4 Units.
The study and critical analysis of computer games as art objects, cultural artifacts, gateways to virtual worlds, educational aids, and tools for persuasion and social change. Emphasis on understanding games in their historical and cultural context.

Prerequisite: Satisfaction of the UC Entry Level Writing Requirement.

Overlaps with UNI STU 12A, UNI STU 12B, UNI STU 12C.

(Ill)

I&C SCI 61. Game Systems and Design. 4 Units.
Principles and usage of game design elements. Introduction to technologies that support modern computer games. Students design, implement, and critique several small games.

(Ii)

I&C SCI 62. Game Technologies and Interactive Media. 4 Units.
Technologies for interactive media and game design. Web-based software systems, virtual world platforms, and game engines. Emphasis on conceptual and architectural aspects of these technologies.

Prerequisite: I&C SCI 21 or CSE 21 or I&C SCI 31 or CSE 41 or IN4MATX 42. IN4MATX 42 with a grade of C or better.

I&C SCI 80. Special Topics in Information and Computer Science. 2-4 Units.
Studies in selected areas of information and computer sciences. Topics addressed vary each quarter.

Prerequisite: Prerequisites vary.

Repeatability: Unlimited as topics vary.

I&C SCI 90. New Students Seminar. 1 Unit.
Introduces students to the Donald Bren School of Information and Computer Sciences. Focuses on advising students making the transition to UCI, community building, and mostly surveying the technical areas within departments in ICS, via talks by faculty on their research.

Grading Option: Pass/no pass only.

I&C SCI 105. Digital Information Systems. 4 Units.
Design and analysis of digital information systems. Covers underlying database and network technology, and software engineering principles used to build these systems. Evaluating digital information systems, and recognizing common flaws and vulnerabilities.

Prerequisite: I&C SCI 10 or I&C SCI 21 or CSE 21 or I&C SCI H21 or IN4MATX 41. I&C SCI 10 with a grade of C or better. I&C SCI 21 with a grade of C or better. CSE 21 with a grade of C or better. I&C SCI H21 with a grade of C or better. IN4MATX 41 with a grade of C or better.

Overlaps with I&C SCI 52, IN4MATX 43.

I&C SCI 139W. Critical Writing on Information Technology. 4 Units.
Study and practice of critical writing and oral communication as it applies to information technology. Each student writes assignments of varying lengths, totaling at least 4,000 words.

Prerequisite: Satisfactory completion of the Lower-Division Writing requirement.

Restriction: Upper-division students only.

(Ib)

I&C SCI 160. Graphics Processors and Game Platforms. 4 Units.
Principles of computer architecture emphasizing hardware used with general purpose processor to support high-performance computer games and graphics engines.

Prerequisite: I&C SCI 51.

Overlaps with COMPSCI 152.

I&C SCI 161. Game Engine Lab. 4 Units.
The use of an open source game or graphics engine in the design and implementation of a computer game. Principles of game engine design. Students work on teams to design, implement, and evaluate new computer games based on an engine.

Prerequisite: I&C SCI 45C or I&C SCI 65.
I&C SCI 162. Modeling and World Building. 4 Units.
Use of 3D modeling software and related tools to design and create animated, textured models and expansive virtual worlds incorporating objects, scenes, and venues for activity within game worlds and online environments.
Prerequisite: COMPSCI 112.

I&C SCI 163. Mobile and Ubiquitous Games. 4 Units.
Design and technology of mobile games, including mixed reality gaming, urban games, and locative media. Case studies of significant systems. Uses and limitations of location-based technologies. Infrastructures and their relationships to gameplay and design.
Prerequisite: I&C SCI 61 and (I&C SCI 10 or I&C SCI 21 or I&C SCI 31 or IN4MATX 41).

I&C SCI 166. Game Design. 4 Units.
Game design takes into consideration psychology, narrative, platform features and limitations, marketing, computer science capabilities, human-computer interface principles, industry trends, aesthetic judgment, and other factors. Students focus on video game design through lectures, readings, presentations, implementation, and play testing.
Prerequisite: I&C SCI 61 and (IN4MATX 43 or I&C SCI 52).

I&C SCI 167. Multiplayer Game Systems. 4 Units.
Foundations and technologies that enable multiplayer, networked, and persistent virtual environments. Emphasis on database design and management, network protocols, and concurrency control to accommodate large numbers of simultaneous users.
Prerequisite: I&C SCI 51.

I&C SCI 168. Multiplayer Game Project. 4 Units.
Designing and implementing a multiplayer, networked, and persistent virtual environment or game. Emphasis on cultural aspects, community building, user interface issues and design, security, privacy, and economics.
Prerequisite: (I&C SCI 52 or IN4MATX 43) and I&C SCI 167. IN4MATX 43 with a grade of C or better.

I&C SCI 169A. Capstone Game Project I. 4 Units.
Students work in teams to design and implement a new computer game or virtual world. Emphasis on sound, art, and level design, building a community, cut scenes, production values, full utilization of hardware and software platform, and current industry trends.
Prerequisite: I&C SCI 168.
Grading Option: In progress only.

I&C SCI 169B. Capstone Game Project II. 4 Units.
Students work in teams to design and implement a new computer game or virtual world. Emphasis on sound, art, and level design, building a community, cut scenes, production values, full utilization of hardware and software platform, and current industry trends.
Prerequisite: I&C SCI 169A.

I&C SCI 192. Industrial or Public Sector Field Study. 2 Units.
Students participate in an off-campus, supervised internship for a minimum of 60 hours. Students apply classroom knowledge through internship projects in the private sector or nonprofit agencies.
Grading Option: Pass/no pass only.
Repeatability: May be taken for credit 2 times.

I&C SCI 193. Tutoring in ICS. 2 Units.
Principles and practice of providing technical assistance to novice learners in information and computer sciences.
Repeatability: May be taken for credit for 18 units.
Restriction: ICS Peer Tutoring Program students only.

I&C SCI H197. Honors Seminar. 2 Units.
An overview of computer science and selected recent trends in research. Students attend talks on current faculty research, with opportunities for discussion.
Grading Option: Pass/no pass only.
Restriction: Bren School of ICS Honors Program or Campuswide Honors Program students only.
I&C SCI 398A. Teaching Assistant Training Seminar. 2 Units.
Theories, methods, and resources for teaching computer science at the university level, particularly by teaching assistants. Classroom presentations, working with individuals, grading, motivating students. Participants will give and critique presentations and may be videotaped while teaching.
Grading Option: Satisfactory/unsatisfactory only.

I&C SCI 398B. Advanced Teaching Assistant Seminar. 4 Units.
Teaching computer science at the university level, emphasizing issues in teaching an entire course. Course organization, designing examinations and projects, grading, motivating students. Participants will begin to assemble teaching portfolios.
Prerequisite: I&C SCI 398A.
Grading Option: Satisfactory/unsatisfactory only.

I&C SCI 399. University Teaching. 4 Units.
Involves on-the-job experience for Teaching Assistants.
Repeatability: May be repeated for credit unlimited times.
Restriction: Teaching assistants only.

Statistics Courses

STATS 5. Seminar in Data Science. 1 Unit.
An introduction to the field of Data Science; intended for entering freshman and transfers.
Grading Option: Pass/no pass only.
Restriction: Information and Computer Science majors only.

STATS 7. Basic Statistics. 4 Units.
Introduces basic inferential statistics including confidence intervals and hypothesis testing on means and proportions, t-distribution, Chi Square, regression and correlation. F-distribution and nonparametric statistics included if time permits. Course may be offered online.
Overlaps with STATS 8, MGMT 7.
Restriction: STATS 7 may not be taken for credit if taken after STATS 67.
(Va)

STATS 8. Introduction to Biological Statistics . 4 Units.
Introductory statistical techniques used to collect and analyze experimental and observational data from health sciences and biology. Includes exploration of data, probability and sampling distributions, basic statistical inference for means and proportions, linear regression, and analysis of variance. Course may be offered online.
Overlaps with SOCECOL 13, MGMT 7.
(Va)

STATS 67. Introduction to Probability and Statistics for Computer Science. 4 Units.
Introduction to the basic concepts of probability and statistics with discussion of applications to computer science.
Prerequisite: MATH 2B.
Overlaps with STATS 7, MGMT 7.
Restriction: STATS 7 and MGMT 7 may not be taken for credit if taken after STATS 67.
(Va)

STATS 68. Exploratory Data Analysis. 4 Units.
Introduces key concepts in statistical computing. Techniques such as exploratory data analysis, data visualization, simulation, and optimization methods, will be presented in the context of data analysis within a statistical computing environment.
Prerequisite: STATS 7 and I&C SCI 31.
STATS 110. Statistical Methods for Data Analysis I. 4 Units.
Introduction to statistical methods for analyzing data from experiments and surveys. Methods covered include two-sample procedures, analysis of variance, simple and multiple linear regression.
Prerequisite: STATS 7 or (STATS 120A and STATS 120B and STATS 120C).
Concurrent with STATS 201.

STATS 111. Statistical Methods for Data Analysis II. 4 Units.
Introduction to statistical methods for analyzing data from surveys or experiments. Emphasizes application and understanding of methods for categorical data including contingency tables, logistic and Poisson regression, loglinear models.
Prerequisite: STATS 110.
Concurrent with STATS 202.

STATS 112. Statistical Methods for Data Analysis III. 4 Units.
Introduction to statistical methods for analyzing longitudinal data from experiments and cohort studies. Topics covered include survival methods for censored time-to-event data, linear mixed models, non-linear mixed effects models, and generalized estimating equations.
Prerequisite: STATS 111.
Concurrent with STATS 203.

STATS 115. Introduction to Bayesian Data Analysis. 4 Units.
Basic Bayesian concepts and methods with emphasis on data analysis. Special emphasis on specification of prior distributions. Development for one-two samples and on to binary, Poisson, and linear regression. Analyses performed using free OpenBugs software.
Prerequisite: STATS 120C. Recommended: STATS 110 or STATS 202 or STATS 210 or STATS 211.
Concurrent with STATS 205.

STATS 120A. Introduction to Probability and Statistics. 4 Units.
Introduction to basic principles of probability and statistical inference. Axiomatic definition of probability, random variables, probability distributions, expectation.
Prerequisite: MATH 2A and MATH 2B and (MATH 2D or MATH 4).
Overlaps with MATH 130A.

STATS 120B. Introduction to Probability and Statistics. 4 Units.
Introduction to basic principles of probability and statistical inference. Point estimation, interval estimating, and testing hypotheses, Bayesian approaches to inference.
Prerequisite: STATS 120A.

STATS 120C. Introduction to Probability and Statistics. 4 Units.
Introduction to basic principles of probability and statistical inference. Linear regression, analysis or variance, model checking.
Prerequisite: STATS 120B and (Math 3A or Math 6G or I&C SCI 6N).

STATS 121. 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 COMPSCI 278.

STATS 140. Multivariate Statistical Methods. 4 Units.
Theory and application of multivariate statistical methods. Topics include: statistical inference for the multivariate normal model and its extensions to multiple samples and regression, use of statistical packages for data visualization and reduction, discriminant analysis, cluster analysis, and factor analysis.
Prerequisite: STATS 120C or MATH 121A.
Concurrent with STATS 240.
STATS 170A. Project in Data Science. 4 Units.
Problem definition and analysis, data representation, algorithm selection, solution validation, and results presentation. Students do team projects and lectures cover analysis alternatives, project planning, and data analysis issues. First quarter emphasizes approach selection, project planning, and experimental design.

Prerequisite:

Grading Option: In progress only.

Restriction: Seniors only. Data Science majors have first consideration for enrollment.

STATS 170B. Project in Data Science. 4 Units.
Problem definition and analysis, data representation, algorithm selection, solution validation, and results presentation. Students do team projects and lectures cover analysis alternatives, project planning, and data analysis issues. Second quarter emphasizes project execution and analysis, and presentation of results.

Prerequisite: STATS 170A.

Restriction: Seniors only. Data Science majors have first consideration for enrollment.

STATS 199. Individual Study. 2-5 Units.
Individual research or investigations under the direction of an individual faculty member.

Repeatability: May be repeated for credit unlimited times.

STATS 200A. Intermediate Probability and Statistical Theory. 4 Units.
Basics of probability theory, random variables and basic transformations, univariate distributions—discrete and continuous, multivariate distributions.

Prerequisite: STATS 120C.

STATS 200B. Intermediate Probability and Statistical Theory. 4 Units.
Random samples, transformations, limit laws, normal distribution theory, introduction to stochastic processes, data reduction, point estimation (maximum likelihood).

Prerequisite: STATS 200A.

STATS 200C. Intermediate Probability and Statistical Theory. 4 Units.
Interval estimation, hypothesis testing, decision theory and Bayesian inference, basic linear model theory.

Prerequisite: STATS 200B.

STATS 201. Statistical Methods for Data Analysis I. 4 Units.
Introduction to statistical methods for analyzing data from experiments and surveys. Methods covered include two-sample procedures, analysis of variance, simple and multiple linear regression.

Prerequisite: Knowledge of basic statistics.

Concurrent with STATS 110.

STATS 202. Statistical Methods for Data Analysis II. 4 Units.
Introduction to statistical methods for analyzing data from surveys or experiments. Emphasizes application and understanding of methods for categorical data including contingency tables, logistic and Poisson regression, loglinear models.

Prerequisite: STATS 201 or STATS 210.

Concurrent with STATS 111.

STATS 203. Statistical Methods for Data Analysis III. 4 Units.
Introduction to statistical methods for analyzing longitudinal data from experiments and cohort studies. Topics covered include survival methods for censored time-to-event data, linear mixed models, non-linear mixed effects models, and generalized estimating equations.

Prerequisite: STATS 202.

Concurrent with STATS 112.
STATS 205. Introduction to Bayesian Data Analysis. 4 Units.
Basic Bayesian concepts and methods with emphasis on data analysis. Special emphasis on specification of prior distributions. Development for one-
two samples and on to binary, Poisson and linear regression. Analyses performed using free OpenBugs software.

Prerequisite: STATS 120C. Recommended: STATS 110 or STATS 202 or STATS 210 or STATS 211.

Concurrent with STATS 115.

STATS 210. Statistical Methods I: Linear Models. 4 Units.
Statistical methods for analyzing data from surveys and experiments. Topics include randomization and model-based inference, two-sample methods,
analysis of variance, linear regression and model diagnostics.

Prerequisite: Knowledge of basic statistics, calculus, linear algebra.

STATS 211. Statistical Methods II: Generalized Linear Models. 4 Units.
Development of the theory and application of generalized linear models. Topics include likelihood estimation and asymptotic distributional theory
for exponential families, quasi-likelihood and mixed model development. Emphasizes methodological development and application to real scientific
problems.

Corequisite: STATS 200B.
Prerequisite: STATS 210.

STATS 212. Statistical Methods III: Methods for Correlated Data. 4 Units.
Development and application of statistical methods for analyzing correlated data. Topics covered include repeated measures ANOVA, linear mixed
models, non-linear mixed effects models, and generalized estimating equations. Emphasizes both theoretical development and application of the
presented methodology.

Prerequisite: STATS 211.

STATS 220A. Advanced Probability and Statistics Topics. 4 Units.
Advanced topics in probability and statistical inference including measure theoretic probability, large sample theory, decision theory, resampling and
Monte Carlo methods, nonparametric methods.

Prerequisite: STATS 200C.

STATS 220B. Advanced Probability and Statistics Topics. 4 Units.
Advanced topics in probability and statistical inference including measure theoretic probability, large sample theory, decision theory, resampling and
Monte Carlo methods, nonparametric methods.

Prerequisite: STATS 220A.

STATS 225. Bayesian Statistical Analysis. 4 Units.
Introduction to the Bayesian approach to statistical inference. Topics include univariate and multivariate models, choice of prior distributions, hierarchical
models, computation including Markov chain Monte Carlo, model checking, and model selection.

Prerequisite: Two quarters of upper-division or graduate training in probability and statistics.

STATS 226. Advanced Topics in Modern Bayesian Statistical Inference. 4 Units.
Modern Bayesian Statistics: selected topics from theory of Markov chains, application of theory to modern methods of Markov chain Monte Carlo
sampling; Bayesian non-parametric and semiparametric modeling, including Dirichlet Process Mixtures; Mixtures of Polya Trees.

Prerequisite: STATS 200C.

STATS 230. Statistical Computing Methods. 4 Units.
Numerical computations and algorithms with applications in statistics. Topics include optimization methods including the EM algorithm, random number
generation and simulation, Markov chain simulation tools, and numerical integration.

Prerequisite: Two quarters of upper-division or graduate training in probability and statistics.
Overlaps with COMPSCI 206.
STATS 235. Modern Data Analysis Methods . 4 Units.
Introduces selected modern tools for data analysis. Emphasizes use of computational and resampling techniques for data analyses when the data do not conform to standard toolbox of regression models and/or complexity of modeling problem threatens validity of standard methods.
Prerequisite: STATS 120C.
Restriction: Graduate students only.

STATS 240. Multivariate Statistical Methods. 4 Units.
Theory and application of multivariate statistical methods. Topics include: statistical inference for the multivariate normal model and its extensions to multiple samples and regression, use of statistical packages for data visualization and reduction, discriminant analysis, cluster analysis, and factor analysis.
Prerequisite: STATS 120C or MATH 121A.

STATS 245. Time Series Analysis. 4 Units.
Statistical models for analysis of time series from time and frequency domain perspectives. Emphasizes theory and application of time series data analysis methods. Topics include ARMA/ARIMA models, model identification/estimation, linear operators, Fourier analysis, spectral estimation, state space models, Kalman filter.
Prerequisite: STATS 200C.

STATS 246. Spectral Analysis. 4 Units.
Spectral methods that are most commonly utilized for analyzing univariate and multivariate time series and signals. These methods include spectral and coherence estimation, transfer function modeling, classification and discrimination of time series, non-stationary time series, time-frequency analysis, and wavelets analysis.
Prerequisite: STATS 200A and STATS 200B and STATS 210.

STATS 250. Biostatistics. 4 Units.
Statistical methods commonly used to analyze data arising from clinical studies. Topics include analysis of observational studies and randomized clinical trials, techniques in the analysis of survival and longitudinal data, approaches to handling missing data, meta-analysis, nonparametric methods.
Prerequisite: STATS 210.

STATS 255. Statistical Methods for Survival Data. 4 Units.
Statistical methods for analyzing survival data from cohort studies. Topics include parametric and nonparametric methods, the Kaplan-Meier estimator, log-rank tests, regression models, the Cox proportional hazards model and accelerated failure time models, efficient sampling designs, discrete survival models.
Prerequisite: STATS 211.

STATS 257. Introduction to Statistical Genetics. 4 Units.
Provides students with knowledge of the basic principles, concepts, and methods used in statistical genetic research. Topics include principles of population genetics, and statistical methods for family- and population-based studies.
Prerequisite: Two quarters of upper-division or graduate training in statistical methods.
Same as EPIDEM 215.

STATS 260. Inference with Missing Data. 4 Units.
Statistical methods and theory useful for analysis of multivariate data with partially observed variables. Bayesian and likelihood-based methods developed. Topics include EM-type algorithms, MCMC samplers, multiple imputation, and general location model. Applications from economics, education, and medicine are discussed.
Prerequisite: STATS 200C and STATS 210.

STATS 262. Theory and Practice of Sample Surveys. 4 Units.
Basic techniques and statistical methods used in designing surveys and analyzing collected survey data. Topics include simple random sampling, ratio and regression estimates, stratified sampling, cluster sampling, sampling with unequal probabilities, multistage sampling, and methods to handle nonresponse.
Prerequisite: STATS 120C.
STATS 265. Causal Inference. 4 Units.
Various approaches to causal inference focusing on the Rubin causal model and propensity-score methods. Topics include randomized experiments, observational studies, non-compliance, ignorable and non-ignorable treatment assignment, instrumental variables, and sensitivity analysis. Applications from economics, politics, education, and medicine.
Prerequisite: STATS 200C and STATS 210.

STATS 270. Stochastic Processes. 4 Units.
Introduction to the theory and application of stochastic processes. Topics include Markov chains, continuous-time Markov processes, Poisson processes, and Brownian motion. Applications include Markov chain Monte Carlo methods and financial modeling (for example, option pricing).
Prerequisite: STATS 120C.
Overlaps with MATH 271A, MATH 271B, MATH 271C.

STATS 275. Statistical Consulting. 4 Units.
Training in collaborative research and practical application of statistics. Emphasis on effective communication as it relates to identifying scientific objectives, formulating a statistical analysis plan, choice of statistical methods, and interpretation of results and their limitations to non-statisticians.
Prerequisite: STATS 203 or STATS 212.
Repeatability: May be taken for credit 2 times.

STATS 280. Seminar in Statistics. 0.5 Units.
Periodic seminar series covering topics of current research in statistics and its application.
Grading Option: Satisfactory/unsatisfactory only.
Repeatability: May be repeated for credit unlimited times.
Restriction: Graduate students only.

STATS 281. Topics in Astrostatistics. 1-4 Units.
Topics in statistical methods for astronomy, astrophysics, particle physics, and solar physics, typically including spectral analysis, image processing and analysis, time series, classification, clustering, massive data, etc. Emphasizes computationally intensive methods, Bayesian and frequentist methods, machine learning, and signal processing.
Repeatability: May be repeated for credit unlimited times.
Restriction: Graduate students only.

STATS 285. Special Topics in Statistics. 4 Units.
Studies in selected areas of statistics. Topics addressed vary each quarter.
Repeatability: Unlimited as topics vary.

STATS 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.

STATS 299. Individual Study. 2-12 Units.
Individual research or investigation under the direction of an individual faculty member.
Repeatability: May be repeated for credit unlimited times.