DEPARTMENT OF COMPUTER AND DATA SCIENCES

Glennan Building (7071)
http://engineering.case.edu/cds/
Phone: 216.368.2800; Fax: 216.368.6888

Vipin Chaudhary (vipin@case.edu), Kevin J. Kranzusch Professor and Chair

While the BS degree in Computer Science at Case Western Reserve University was approved in 1987, the Department of Computer and Data Sciences (CDS) was recently established in 2019. Computer Science is the study of the theory, practice, and application of computer systems. Data Science is an interdisciplinary field that utilizes computer systems, computational algorithms, and statistical methods to manage, analyze, and visualize data from different domains in order to extract information and knowledge from data. Computer Science and Data Science are at the heart of modern technology with applications in many disciplines. They both have a profound impact on our society and drive job creation. Starting salaries in our fields are consistently ranked at the top of all college majors. Our graduates work in cutting-edge companies—from giants to start-ups, in a variety of technology sectors, including computer and internet, business and finance, healthcare and medical devices, energy, and consulting.

CDS offers a BS degree in Computer Science, a Computer Science major toward the BA degree, a BS degree in Data Science & Analytics, a combined BS/MS program in Computer Science, and MS and PhD degrees in Computer Science. CDS also offers minors in Computer Science, Computer Gaming, and Artificial Intelligence. The minor in Applied Data Science is administered by the Department of Materials Science and Engineering https://engineering.case.edu/materials-science-and-engineering/academics/.

Educational Philosophy

The CDS department is dedicated to developing high-quality graduates who will take positions of leadership as their careers advance. We recognize that the increasing role of technology in virtually every facet of our society, life, and culture makes it vital that our students have access to progressive and cutting-edge higher education programs. The program values for all of the degree programs in the department are:

- mastery of fundamentals
- creativity
- social awareness
- leadership skills
- professionalism

Stressing excellence in these core values helps to ensure that our graduates are valued and contributing members of our global society and that they will carry on the tradition of industrial and academic leadership established by our alumni.

Our goal is to graduate students who have fundamental technical knowledge of their profession and the requisite technical breadth and communications skills to become leaders in creating the new techniques and technologies which will advance their fields. To achieve this goal, the department offers a wide range of technical specialties consistent with the breadth of computer science and data science, including recent developments in the fields. Because of the rapid pace of advancement in these fields, our degree programs emphasize a broad and foundational science and technology background that equips students for future developments. Our programs include a wide range of electives and our students can also develop individualized programs that can combine computer and data sciences with other disciplines.

Research

At Case Western, we thrive to provide outstanding educational experiences for both our undergraduate and graduate students, while performing cutting edge research in:

1. Algorithms, Theory, and Quantum Computing
2. Artificial Intelligence and Machine Learning
3. Bioinformatics and Systems Biology
4. Data Science and Analytics
5. Databases & Data Mining
7. Security and Privacy
8. Software Engineering

Primary Faculty Appointments

Erman Ayday, PhD
(Georgia Institute of Technology)
Assistant Professor
Cybersecurity, network security, trust and reputation management, big data analytics

Vipin Chaudhary, PhD
(The University of Texas at Austin)
Kevin J. Kranzusch Professor and Chair
High performance computing, machine learning, computational and data science, computer aided diagnosis and interventions, and quantum computing

Harold S. Connamacher, PhD
(University of Toronto)
Robert J. Herbold Associate Professor in Transformative Teaching
Constraint satisfaction problems, graph theory, random structures, and algorithms

Mehmet Koyuturk, PhD
(Purdue University)
Professor
Bioinformatics and computational biology, computational modeling and algorithm development for systems biology, integration, mining and analysis of biological data, algorithms for distributed systems

Michael Lewicki, PhD
(California Institute of Technology)
Professor
Computational perception and scene analysis, visual representation and processing, auditory representation and analysis

Jing Li, PhD
(University of California, Riverside)
Leonard Case Jr. Professor
Computational biology and bioinformatics, data mining and machine learning, data science and analytics, algorithms
Vincenzo Liberatore, PhD
(Rutgers University)
Associate Professor
Distributed systems, Internet computing, randomized algorithms

Orhan Ozguner, PhD
(Case Western Reserve University)
Assistant Professor
Algorithms, data science, data structure, programming

H. Andy Podgurski, PhD
(University of Massachusetts, Amherst)
Professor
Software engineering methodology and tools, especially use of data mining, machine learning, and program analysis techniques in software testing, fault detection and localization, reliable engineering and software security, electronic medical records, privacy

Michael Rabinovich, PhD
(University of Washington)
Professor
Computer networks, distributed systems, Internet security and performance

Soumya Ray, PhD
(University of Wisconsin, Madison)
Associate Professor
Artificial intelligence, machine learning, reinforcement learning, automated planning, applications to interdisciplinary problems including medicine and bioinformatics

An Wang, PhD
(George Mason University)
Assistant Professor
Systems and network security

Yinghui Wu, PhD
(University of Edinburgh (UK))
Assistant Professor
Data science

Xusheng Xiao, PhD
(North Carolina State University)
Assistant Professor
Software engineering, computer security

Shuai Xu, PhD
(Florida International University)
Assistant Professor
Algorithms and theory

Yanfang (Fanny) Ye, PhD
(Xiamen University)
Theodore L. and Dana J. Schroeder Associate Professor
Cybersecurity, data mining, machine learning, health intelligence

Advisory Faculty Appointments

Secondary Faculty Appointments

Gurkan Bebek, PhD
Assistant Professor
SOM-Center for Proteomics

Marc Buchner, PhD
Associate Professor
CSE-ECSE

M. Cenk Cavusoglu, PhD
(University of California, Berkeley)
Nord Professor of Engineering

Vira Chankong, PhD
Associate Professor
CSE-ECSE

Roger French, PhD
Professor
CSE-EMSE

Michael Fu, PhD
Timothy E. and Allison L. Schroeder Assistant Professor
CSE-ECSE

Mark Griswold, PhD
Professor
SOM-Radiology
Department of Computer and Data Sciences

Evren Gurkan-Cavusoglu, PhD  
Associate Professor  
CSE-ECSE

Fulai Jin, PhD  
Assistant Professor  
SOM-Genetics & Genome Sciences

Thomas LaFramboise, PhD  
Associate Professor, Genetics  
SOM-Genetics

Xiao Li, PhD  
Assistant Professor  
SOM-RNA Center

Anant Madabhushi, PhD  
Professor  
CSE-EBME

Aziz Nazha, PhD  
Assistant Professor  
SOM, CCF-Ctr of Clinical Artificial Intelligence

Christos Papachristou, PhD  
Professor  
CSE-ECSE

Satya Sahoo, PhD  
Associate Professor  
SOM-Dept. of Population & Quantitative Health Sciences

Peter Thomas, PhD  
Associate Professor  
CAS-MATH

Xiong (Bill) Yu, PhD  
Professor  
CSE-ECIV

Emeritus Faculty
Gultekin Ozsoyoglu, PhD  
(University of Alberta, Canada)  
Emeritus Professor
Graph databases and data mining problems in metabolic networks, metabolomics, and systems biology, bioinformatics, web data mining

Z. Meral Ozsoyoglu, PhD  
(University of Alberta, Canada)  
Emeritus Professor
Database systems, database query languages and optimization, data models, index structures, bioinformatics, medical informatics

Undergraduate Programs
The Computer and Data Science (CDS) department offers programs leading to degrees in:

1. Data Science and Analytics, Bachelor of Science  
2. Computer Science, Bachelor of Science  
3. Computer Science, Bachelor of Arts

These programs provide students with a strong background in the fundamentals of mathematics and science. Students can use their technical and open electives to pursue concentrations in software engineering, algorithms, artificial intelligence, databases, data mining, bioinformatics, security, computer systems, and computer networks. In addition to an excellent technical education, all students in the department are exposed to societal issues, ethics, professionalism, and have the opportunity to develop leadership and creativity skills.

Data Science and Analytics BS
The Data Science and Analytics BS program provides our students with a broad foundation in the field and with the instruction, skills, and experience needed to understand and handle large amounts of data to derive actionable information. The degree program has a unique focus on real-world data and real-world applications.

This major is one of the first undergraduate programs nationwide with a curriculum that includes mathematical modeling, computation, data analytics, visual analytics and project-based applications – all elements of the future emerging field of data science.

Program Educational Objectives
Graduates from the Data Science and Analytics Bachelor of Science program will be prepared to:
1. Analyze real-world problems and create data-driven solutions based on the fundamentals of data science and computing.
2. Work effectively, professionally, and ethically.
3. Assume positions of leadership in industry, academia, public service, and entrepreneurship.
4. Successfully progress in advanced degree programs in data science, computing, and related fields.

Student Outcomes
As preparation for achieving the above educational objectives, the Bachelor of Science degree program in Data Science and Analytics is designed so that students attain the ability to:
1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
3. Communicate effectively in a variety of professional contexts.
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
6. Apply theory, techniques, and tools throughout the data analysis life cycle and employ the resulting knowledge to satisfy stakeholders' needs.

Major Requirements
The major requires completion of the university general education requirements (http://bulletin.case.edu/undergraduatetudies/degreeprograms/) and the engineering general education requirements (http://bulletin.case.edu/undergraduatetudies/csedegree/#bachelorofscienceindatascienceandanalytics) as modified for the Bachelor of Science in Data Science and Analytics degree, and the following courses:

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 133</td>
<td>Introduction to Data Science and Engineering for Majors</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 233</td>
<td>Introduction to Data Structures</td>
<td>4</td>
</tr>
<tr>
<td>CSDS 234</td>
<td>Structured and Unstructured Data</td>
<td>3</td>
</tr>
</tbody>
</table>
Two of STAT 243, STAT 244, STAT 312, and STAT 325 3 units each. Core courses provide our students with a strong background in foundations and analytics.

**Foundations**
Each student must supplement their competence in foundational technical areas by taking at least three additional courses, totaling at least nine credit hours from the following list. Other courses, beyond those that are listed, may be approved by the student’s academic advisor. The following list is organized in topical areas for informational purposes only; foundation courses may come from the same or from different areas.

**Systems**
- CSDS 293: Software Craftsmanship
- CSDS 305: Files, Indexes and Access Structures for Big Data
- CSDS 315: Scalable Parallel Data Analysis
- CSDS 338: Intro to Operating Systems and Concurrent Programming
- CSDS 344: Computer Security
- CSDS 356: Data Privacy
- CSDS 393: Software Engineering

**Statistics**
- STAT 243: Statistical Theory with Application I
- STAT 244: Statistical Theory with Application II
- Any STAT course number 300 or above

**Analytics: Artificial Intelligence**
- CSDS 390: Advanced Game Development Project
- CSDS 391: Introduction to Artificial Intelligence
- CSDS 442: Causal Learning from Data
- CSDS 491: Artificial Intelligence: Probabilistic Graphical Models

**Analytics: Data Mining**
- CSDS 339: Web Data Mining
- CSDS 391: Introduction to Artificial Intelligence
  or CSDS 435: Data Mining

**Theory**
- CSDS 477: Advanced Algorithms
- MATH 201: Introduction to Linear Algebra for Applications

**Applications**
Data science graduates are expected to be knowledgeable in a wide range of areas of applications of the data science profession. The breadth requirement is satisfied by choosing at least two courses (totaling at least six credit hours) from the following list. Additional courses, beyond those that are listed, may be approved by the student’s academic advisor.

**Engineering: Signals**
- ECSE 246: Signals and Systems
- ECSE 313: Signal Processing

**Engineering: Optimization**
- ECSE 346: Engineering Optimization
- ECSE 416: Convex Optimization for Engineering

**Technical Electives**
Two more courses from the core, foundations, and applications lists for at least six credit hours. The combination of core, foundations, and application courses with technical and open electives makes it possible to achieve a minor in fields as different as Economics and Biology. Interested students should contact their advisors.

**Suggested Program of Study: Data Science and Analytics BS**
The following is a suggested program of study. Current students should always consult their advisors and their individual graduation requirement plans as tracked in SIS (http://sis.case.edu).

### First Year

<table>
<thead>
<tr>
<th>Course</th>
<th>Units</th>
<th>Fall</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Principles of Chemistry for Engineers (CHEM 111)</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Introduction to Programming in Java (CSDS 132)</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Calculus for Science and Engineering I (MATH 121)</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>PHED (2 half semester courses)</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SAGES First Year Seminar</td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>General Physics I - Mechanics (PHYS 121)</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

*Units listed for First Year are minimums and may require additional credits depending on the student’s academic advisors.*
### Calculus for Science and Engineering II (MATH 122) or Calculus II (MATH 124)
- **Units:** 4
  - Fall: 4
  - Spring: 0

### Introduction to Data Science and Engineering for Majors (CSDS 133)
- **Units:** 3
  - Fall: 3
  - Spring: 0

### Introduction to Data Structures (CSDS 233)
- **Units:** 4
  - Fall: 4
  - Spring: 0

### PHED (2 half semester courses)
- **Units:** 0
  - Fall: 0
  - Spring: 0

### SAGES University Seminar
- **Units:** 3
  - Fall: 3
  - Spring: 0

**Year Total:** 15

### Structured and Unstructured Data (CSDS 234)
- **Units:** 3
  - Fall: 3

### SAGES University Seminar
- **Units:** 3
  - Fall: 3
  - Spring: 0

### Discrete Mathematics (CSDS 302)
- **Units:** 3
  - Fall: 3

### General Physics II - Electricity and Magnetism (PHYS 122)
- **Units:** 2
  - Fall: 2
  - Spring: 4

### Calculus for Science and Engineering III (MATH 223) or Calculus III (MATH 227)
- **Units:** 2
  - Fall: 2

### Algorithms (CSDS 310)
- **Units:** 3
  - Fall: 3

### Introduction to Database Systems (CSDS 341)
- **Units:** 3
  - Fall: 3

### Elementary Differential Equations (MATH 224) or Differential Equations (MATH 228)
- **Units:** 2
  - Fall: 2

### Breadth Elective
- **Units:** 2
  - Fall: 2

### Probability/Statistics Elective
- **Units:** 3
  - Fall: 3

**Year Total:** 16

### Introduction to Data Analysis (CSDS 313)
- **Units:** 3
  - Fall: 3

### Computer Security (CSDS 344) or Foundations
- **Units:** 4
  - Fall: 4

### Open Elective
- **Units:** 3
  - Fall: 3

### Breadth Elective
- **Units:** 2
  - Fall: 2

### Probability or Statistics
- **Units:** 3
  - Fall: 3

**Third Year Total:** 15

### Professional Communication for Engineers (ENGL 398)
- **Units:** 2
  - Fall: 2

### Professional Communication for Engineers (ENGR 398)
- **Units:** 1
  - Fall: 1

### Breadth Elective
- **Units:** 2
  - Fall: 2

### Open Elective
- **Units:** 3
  - Fall: 3

### Probability or Statistics
- **Units:** 3
  - Fall: 3

**Fourth Year Total:** 15

### Breadth Elective
- **Units:** 2
  - Fall: 2

### Foundations
- **Units:** 3
  - Fall: 3

### Foundations
- **Units:** 3
  - Fall: 3

### Applications
- **Units:** 3
  - Fall: 3

### Open Elective
- **Units:** 3
  - Fall: 3

### Senior Project in Computer Science (CSDS 395)
- **Units:** 4
  - Fall: 4

### Technical Elective
- **Units:** 3

### Open elective
- **Units:** 2

**Total Units in Sequence:** 127

1. University General Education Requirement
2. Engineering General Education Requirement
3. Probability (MATH 380) or Statistics (Two of STAT 243, STAT 244, STAT 312, and STAT 325)
4. Three courses and nine credit hours required from the Foundation list
5. Two courses and six credit hours required from the Applications list

### Applied Data Science Minor
An undergraduate minor in applied data science (http://bulletin.case.edu/schoolengineering/materialsscienceengineering/#undergraduateetext) is administered in the Materials Science and Engineering Department.

### Computer Science BS
The Bachelor of Science degree program in computer science is designed to give a student a strong background in the fundamentals of mathematics and computer science. The curriculum is designed according to the latest ACM/IEEE computer science curriculum guidelines. A graduate of this program should be able to use these fundamentals to analyze and evaluate software systems and the underlying abstractions upon which they are based. A graduate should also be able to design and implement software systems that are state-of-the-art solutions to a variety of computing problems; this includes problems that are sufficiently complex to require the evaluation of design alternatives and engineering trade-offs. In addition to these program-specific objectives, all students in the Case School of Engineering are exposed to societal issues, professionalism, and are provided opportunities to develop leadership skills.

The Bachelor of Science degree program in computer science is accredited by the Computing Accreditation Commission of ABET, http://www.abet.org/.

### Mission
The mission of the Bachelor of Science degree program in computer science is to graduate students who have fundamental technical knowledge of their profession and the requisite technical breadth and communications skills to become leaders in creating the new techniques and technologies which will advance the field of computer science and its application to other disciplines.

### Program Educational Objectives
1. To educate and train students in the fundamentals of computer science and mathematics
2. To educate students with an understanding of real-world computing needs
3. To train students to work effectively, professionally and ethically in computing-related professions
Student Outcomes
As preparation for achieving the above educational objectives, the Bachelor of Science degree program in computer science is designed so that students attain the ability to:

- Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
- Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.
- Communicate effectively in a variety of professional contexts.
- Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
- Function effectively as a member or leader of a team engaged in activities appropriate to the program’s discipline.
- Apply computer science theory and software development fundamentals to produce computing-based solutions.

Core and breadth courses provide our students with the flexibility to work across many disciplines and prepare them for a variety of professions. Our curriculum is designed to teach fundamental skills and knowledge needed by all CS graduates while providing the greatest flexibility in selecting topics. Students are also required to develop depth in at least one of the following technical areas: software engineering; algorithms and theory; computer systems, networks, and security; databases and data mining; bioinformatics; or artificial intelligence.

Major Requirements
Each student is required to complete a total of 20 computer science and computer science related courses, totaling at least 63 credits.

The 20 courses must include:

1. all 6 core courses;
2. at least 5 computer science breadth courses;
3. at least 4 courses in one of the listed computer science depth areas, including all starred courses in that area;
4. and a course from the secure computing requirement list.

The remaining courses needed to fulfill the 20 course requirement may come from the computer science breadth courses, courses of any computer science depth area, and up to 6 of the 20 courses may come from the list of approved technical electives with at most two group 2 courses.

Other computer science related courses not listed here may be used with prior permission from the student’s academic advisor. Some courses appear in more than one list. The same course may be used to satisfy multiple requirements of the core, computer science breadth and depth requirements, but courses may not be double counted for the purpose of achieving 20 separate computer science courses and 63 credits.

In addition to engineering general education requirements and (http://bulletin.case.edu/undergraduatestudies/csedegree/#bachelorofscienceincomputersciencetext) and university general education requirements (http://bulletin.case.edu/undergraduatestudies/degreeprograms/), the major requires the following courses:

Computer Science Core Requirement
All computer science majors are required to complete the following 6 courses.

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 132</td>
<td>Introduction to Programming in Java</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 233</td>
<td>Introduction to Data Structures</td>
<td>4</td>
</tr>
<tr>
<td>CSDS 281</td>
<td>Logic Design and Computer Organization</td>
<td>4</td>
</tr>
<tr>
<td>CSDS 302</td>
<td>Discrete Mathematics</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 310</td>
<td>Algorithms</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 395</td>
<td>Senior Project in Computer Science</td>
<td>4</td>
</tr>
</tbody>
</table>

Computer Science Breadth Requirement
BS students are required to complete at least 5 of the 7 following computer science breadth courses.

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 314</td>
<td>Computer Architecture</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 325</td>
<td>Computer Networks I</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 338</td>
<td>Intro to Operating Systems and Concurrent Programming</td>
<td>4</td>
</tr>
<tr>
<td>CSDS 341</td>
<td>Introduction to Database Systems</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 345</td>
<td>Programming Language Concepts</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 391</td>
<td>Introduction to Artificial Intelligence</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 393</td>
<td>Software Engineering</td>
<td>3</td>
</tr>
</tbody>
</table>

Statistics Requirement
Computer science BS students are required to complete a statistics elective.

One Statistics elective may be chosen from:

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 380</td>
<td>Introduction to Probability</td>
<td>3</td>
</tr>
<tr>
<td>STAT 312</td>
<td>Basic Statistics for Engineering and Science</td>
<td>3</td>
</tr>
<tr>
<td>STAT 313</td>
<td>Statistics for Experimenters</td>
<td>3</td>
</tr>
<tr>
<td>STAT 332</td>
<td>Statistics for Signal Processing</td>
<td>3</td>
</tr>
<tr>
<td>STAT 333</td>
<td>Uncertainty in Engineering and Science</td>
<td>3</td>
</tr>
</tbody>
</table>

Computer Science Secure Computing Requirement
Students pursuing the BS degree must demonstrate competence in the principles and practices of secure computing by completing one of the following courses as part of their 20 computer science or computer science related courses.

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 344</td>
<td>Computer Security</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 356</td>
<td>Data Privacy</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 427</td>
<td>Internet Security and Privacy</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 444</td>
<td>Computer Security</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 448</td>
<td>Smartphone Security</td>
<td>3</td>
</tr>
<tr>
<td>MATH 408</td>
<td>Introduction to Cryptology</td>
<td>3</td>
</tr>
</tbody>
</table>

This course may be double counted as a computer science depth course, as appropriate. There is no secure computing requirement for students pursuing the BA degree.

Computer Science Depth Requirement
Students pursuing the BS degree must demonstrate a depth of competence in one of the technical areas listed below. To complete the depth requirement, students must complete at least four courses in one of the depth areas, including all starred courses. Recommended general background courses are listed following each area where applicable.
Area 1: Software Engineering

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 293</td>
<td>Software Craftsmanship</td>
<td>4</td>
</tr>
<tr>
<td>CSDS 337</td>
<td>Compiler Design</td>
<td>4</td>
</tr>
<tr>
<td>CSDS 344</td>
<td>Computer Security</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 345</td>
<td>Programming Language Concepts *</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 392</td>
<td>App Development for iOS</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 393</td>
<td>Software Engineering *</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 427</td>
<td>Internet Security and Privacy</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 438</td>
<td>High Performance Computing</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 448</td>
<td>Smartphone Security</td>
<td>3</td>
</tr>
</tbody>
</table>

Area 2: Algorithms and Theory

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 310</td>
<td>Algorithms *</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 343</td>
<td>Theoretical Computer Science *</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 394</td>
<td>Introduction to Information Theory</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 440</td>
<td>Machine Learning</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 455</td>
<td>Applied Graph Theory</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 477</td>
<td>Advanced Algorithms</td>
<td>3</td>
</tr>
<tr>
<td>MATH 406</td>
<td>Mathematical Logic and Model Theory</td>
<td>3</td>
</tr>
<tr>
<td>MATH 408</td>
<td>Introduction to Cryptology</td>
<td>3</td>
</tr>
<tr>
<td>PHIL 306</td>
<td>Mathematical Logic and Model Theory</td>
<td>3</td>
</tr>
</tbody>
</table>

Recommended preparation: MATH 380 Introduction to Probability

Area 3: Computer Systems, Networks and Security

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 312</td>
<td>Introduction to Data Science Systems</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 325</td>
<td>Computer Networks I *</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 337</td>
<td>Compiler Design</td>
<td>4</td>
</tr>
<tr>
<td>CSDS 338</td>
<td>Intro to Operating Systems and Concurrent Programming</td>
<td>4</td>
</tr>
<tr>
<td>CSDS 344</td>
<td>Computer Security</td>
<td>3</td>
</tr>
<tr>
<td>or CSDS 444</td>
<td>Computer Security</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 356</td>
<td>Data Privacy</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 427</td>
<td>Internet Security and Privacy</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 428</td>
<td>Computer Communications Networks II</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 438</td>
<td>High Performance Computing</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 448</td>
<td>Smartphone Security</td>
<td>3</td>
</tr>
<tr>
<td>MATH 408</td>
<td>Introduction to Cryptology</td>
<td>3</td>
</tr>
</tbody>
</table>

Area 4: Databases and Data Mining

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 234</td>
<td>Structured and Unstructured Data</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 313</td>
<td>Introduction to Data Analysis</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 339</td>
<td>Web Data Mining</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 341</td>
<td>Introduction to Database Systems *</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 405</td>
<td>Data Structures and File Management</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 433</td>
<td>Database Systems</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 435</td>
<td>Data Mining</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 440</td>
<td>Machine Learning</td>
<td>3</td>
</tr>
<tr>
<td>MATH 382</td>
<td>High Dimensional Probability</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 444</td>
<td>Mathematics of Data Mining and Pattern Recognition</td>
<td>3</td>
</tr>
</tbody>
</table>

Area 5: Bioinformatics

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 310</td>
<td>Algorithms *</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 341</td>
<td>Introduction to Database Systems</td>
<td>3</td>
</tr>
<tr>
<td>or CSDS 341D</td>
<td>Introduction to Databases: DS Major</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 435</td>
<td>Data Mining</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 440</td>
<td>Machine Learning</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 458</td>
<td>Introduction to Bioinformatics *</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 459</td>
<td>Bioinformatics for Systems Biology</td>
<td>3</td>
</tr>
</tbody>
</table>


Area 6: Artificial Intelligence

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 391</td>
<td>Introduction to Artificial Intelligence *</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 394</td>
<td>Introduction to Information Theory</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 440</td>
<td>Machine Learning</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 442</td>
<td>Causal Learning from Data</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 465</td>
<td>Computer Vision</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 491</td>
<td>Artificial Intelligence: Probabilistic Graphical Models</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 496</td>
<td>Artificial Intelligence: Sequential Decision Making</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 497</td>
<td>Artificial Intelligence: Statistical Natural Language Processing</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 499</td>
<td>Algorithmic Robotics</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 484</td>
<td>Computational Intelligence I: Basic Principles</td>
<td>3</td>
</tr>
<tr>
<td>MATH 382</td>
<td>High Dimensional Probability</td>
<td>3</td>
</tr>
</tbody>
</table>

Recommended breadth and preparation: MATH 380 Introduction to Probability, and either ECSE 416 Convex Optimization for Engineering or CSDS 477 Advanced Algorithms.

List of Approved Technical Electives

This list of approved technical electives is divided into groups according to how closely a course is related to the core knowledge areas as defined in the ACM/IEEE computer science curriculum guidelines. For Computer Science BS students, up to 6 of the 20 computer science and computer science related courses may come from this list with up to two courses from group 2. Computer science related courses not listed below may be used as a technical elective but require prior permission from the student's academic advisor.

Group 1

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECSE 301</td>
<td>Digital Logic Laboratory</td>
<td>2</td>
</tr>
<tr>
<td>ECSE 303</td>
<td>Embedded Systems Design and Laboratory</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 315</td>
<td>Digital Systems Design</td>
<td>4</td>
</tr>
<tr>
<td>ECSE 317</td>
<td>Computer Design - FPGAs</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 419</td>
<td>Computer System Architecture</td>
<td>3</td>
</tr>
</tbody>
</table>

Any CSDS course.
<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECSE 485</td>
<td>VLSI Systems</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 488</td>
<td>Embedded Systems Design</td>
<td>3</td>
</tr>
<tr>
<td>MATH 330</td>
<td>Introduction to Scientific Computing</td>
<td>3</td>
</tr>
<tr>
<td>MATH 431</td>
<td>Introduction to Numerical Analysis I</td>
<td>3</td>
</tr>
</tbody>
</table>

**Group 2**

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSCI 351</td>
<td>Exploratory Data Science</td>
<td>3</td>
</tr>
<tr>
<td>DSCI 352</td>
<td>Applied Data Science Research</td>
<td>3</td>
</tr>
<tr>
<td>DSCI 353</td>
<td>Data Science: Statistical Learning, Modeling and Prediction</td>
<td>3</td>
</tr>
<tr>
<td>ECON 380</td>
<td>Computational Economics</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 245</td>
<td>Electronic Circuits</td>
<td>4</td>
</tr>
<tr>
<td>ECSE 246</td>
<td>Signals and Systems</td>
<td>4</td>
</tr>
<tr>
<td>ECSE 304</td>
<td>Control Engineering I with Laboratory</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 305</td>
<td>Control Engineering I Laboratory</td>
<td>1</td>
</tr>
<tr>
<td>ECSE 309</td>
<td>Electromagnetic Fields I</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 313</td>
<td>Signal Processing</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 318</td>
<td>VLSI/CAD</td>
<td>4</td>
</tr>
<tr>
<td>ECSE 319</td>
<td>Applied Probability and Stochastic Processes for Biology</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 324</td>
<td>Modeling and Simulation of Continuous Dynamical Systems</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 346</td>
<td>Engineering Optimization</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 350</td>
<td>Operations and Systems Design</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 354</td>
<td>Digital Communications</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 375</td>
<td>Applied Control</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 408</td>
<td>Introduction to Linear Systems</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 413</td>
<td>Nonlinear Systems I</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 414</td>
<td>Wireless Communications</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 416</td>
<td>Convex Optimization for Engineering</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 489</td>
<td>Robotics I</td>
<td>3</td>
</tr>
<tr>
<td>ENGR 210</td>
<td>Introduction to Circuits and Instrumentation</td>
<td>4</td>
</tr>
<tr>
<td>MATH 224</td>
<td>Elementary Differential Equations</td>
<td>3</td>
</tr>
<tr>
<td>MATH 228</td>
<td>Differential Equations</td>
<td>3</td>
</tr>
<tr>
<td>MATH 303</td>
<td>Elementary Number Theory</td>
<td>3</td>
</tr>
<tr>
<td>MATH 308</td>
<td>Introduction to Abstract Algebra</td>
<td>3</td>
</tr>
<tr>
<td>MATH 327</td>
<td>Convexity and Optimization</td>
<td>3</td>
</tr>
<tr>
<td>MATH 413</td>
<td>Graph Theory</td>
<td>3</td>
</tr>
<tr>
<td>MATH 439</td>
<td>Bayesian Scientific Computing</td>
<td>3</td>
</tr>
<tr>
<td>MATH 497</td>
<td>Stochastic Models: Time Series and Markov Chains</td>
<td>3</td>
</tr>
<tr>
<td>PHIL 201</td>
<td>Introduction to Logic</td>
<td>3</td>
</tr>
<tr>
<td>PHYS 221</td>
<td>Introduction to Modern Physics</td>
<td>3</td>
</tr>
<tr>
<td>PHYS 250</td>
<td>Computational Methods in Physics</td>
<td>3</td>
</tr>
<tr>
<td>STAT 345</td>
<td>Theoretical Statistics I</td>
<td>3</td>
</tr>
<tr>
<td>STAT 346</td>
<td>Theoretical Statistics II</td>
<td>3</td>
</tr>
</tbody>
</table>

**First Year**

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAGES First Year Seminar*</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Principles of Chemistry for Engineers (CHEM 111)</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Calculus for Science and Engineering I (MATH 121)</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Introduction to Programming in Java (CSDS 132)</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>PHED (2 half semester courses)*</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Open elective</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>SAGES University Seminar*</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>General Physics I - Mechanics (PHYS 121)</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Calculus for Science and Engineering II (MATH 122) or Calculus II (MATH 124)</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Introduction to Data Structures (CSDS 233)</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>PHED (2 half semester courses)*</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Year Total:</td>
<td></td>
<td>18 15</td>
</tr>
</tbody>
</table>

**Second Year**

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAGES University Seminar*</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>General Physics II - Electricity and Magnetism (PHYS 122)</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Calculus for Science and Engineering III (MATH 223) or Calculus III (MATH 227)</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Discrete Mathematics (CSDS 302)</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Logic Design and Computer Organization (CSDS 281)</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Introduction to Linear Algebra for Applications (MATH 201) or Linear Algebra (MATH 307)</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Statistics elective*a</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Computer science breadth course*b</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Breadth elective**</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Technical elective*d</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Year Total:</td>
<td></td>
<td>17 15</td>
</tr>
</tbody>
</table>

**Third Year**

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms (CSDS 310)</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Computer science breadth course*b</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Computer science breadth course*b</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Technical elective*d</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Breadth elective**</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Professional Communication for Engineers (ENGL 398)</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Professional Communication for Engineers (ENGR 398)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Computer science breadth course*b</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Computer science breadth course*b</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Computer science depth course*c</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Breadth elective**</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Year Total:</td>
<td></td>
<td>15 15</td>
</tr>
</tbody>
</table>

**Suggested Program of Study: Computer Science BS**

The following is a suggested program of study. Current students should always consult their advisors and their individual graduation requirement plans as tracked in SIS (http://sis.case.edu/).
and routinely complete all of the requirements for the double major in computer science. Students can major in another discipline in addition to computer science. This degree is particularly useful for computer professionals, but it is less technical than the Bachelor of Science degree program in computer science. This degree is designed so that it allows students to major in computer science and take all of the pre-med courses in a four-year period.

**Mission**

The mission of the Bachelor of Arts degree program in computer science is to graduate students who have fundamental technical knowledge of their profession and the requisite technical breadth and communications skills to become leaders in creating the new techniques and technologies which will advance the field of computer science and its application to other disciplines.

**Program Educational Objectives**

1. To educate and train students in the fundamentals of computer science and mathematics
2. To educate students with an understanding of real-world computing needs
3. To train students to work effectively, professionally and ethically in computing-related professions

**Student Outcomes**

As preparation for achieving the above educational objectives, the Bachelor of Arts degree program in computer science is designed so that students attain the ability to:

- Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
- Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
- Communicate effectively in a variety of professional contexts.
- Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
- Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
- Apply computer science theory and software development fundamentals to produce computing-based solutions.

Core and breadth courses provide our students with the flexibility to work across many disciplines and prepare them for a variety of professions. Our curriculum is designed to teach fundamental skills and knowledge needed by all CS graduates while providing the greatest flexibility in selecting topics.

**Major Requirements**

Students are required to complete a total of 13 computer science and computer science related courses, totaling at least 42 credits. The 13 courses for a total of 42 credits must include all 6 core courses and at least 3 computer science breadth courses. The remaining 4 courses may be any CS Course, defined below, plus at most 3 courses from either the group 1 or the group 3 list of approved technical electives. There is no depth requirement for the Computer Science BA degree.

In addition to arts & sciences general education requirements (http://bulletin.case.edu/undergraduatestudies/casdegree/) and university general education requirements (http://bulletin.case.edu/)

---

### Fourth Year

<table>
<thead>
<tr>
<th>Course</th>
<th>Fall</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breadth elective**</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Computer science depth course**</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Technical elective^d</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Technical elective^d</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Senior Project in Computer Science (CSDS 395)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Computer science depth course^c</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Technical elective^d</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Technical elective^d</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td><strong>Total Units in Sequence:</strong></td>
<td>15</td>
<td>16</td>
</tr>
</tbody>
</table>

**Hours Required for Graduation: 126**

- * University general education requirement
- ** Engineering general education requirement (http://bulletin.case.edu/undergraduatestudies/csedegree/#bachelorofscienceincomputersciencetext)
- c Each student must complete 4 courses in one of the computer science depth areas listed above, including all starred courses.
- d Chosen from additional computer science breadth courses, depth courses, or the list of approved technical electives. Any other course used as a technical elective must be approved by the student's advisor.

---

**Computer Science BA**

The Bachelor of Arts degree program in computer science is a combination of a liberal arts program and a computing major. It is a professional program in the sense that graduates can be employed as computer professionals, but it is less technical than the Bachelor of Science degree program in computer science. This degree is particularly suitable for students with a wide range of interests. For example, students can major in another discipline in addition to computer science and routinely complete all of the requirements for the double major in order to achieve a 4 year period. This is possible because over a third of the courses in the program are open electives. Furthermore, if a student is majoring in computer science and a second technical field such as mathematics or physics many of the technical electives will be accepted for both majors. Another example of the utility of this program is that it routinely allows students to major in computer science and take all of the pre-med courses in a four-year period.
undergraduatestudies/degreeprograms/), the major requires the following courses:

**Computer Science BA Mathematics Requirement**

BA students must complete two calculus courses.

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 121</td>
<td>Calculus for Science and Engineering I</td>
<td>4</td>
</tr>
<tr>
<td>or MATH 123</td>
<td>Calculus I</td>
<td></td>
</tr>
<tr>
<td>or MATH 125</td>
<td>Math and Calculus Applications for Life, Managerial, and Social Sci I</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 122</td>
<td>Calculus for Science and Engineering II (or)</td>
<td>4</td>
</tr>
<tr>
<td>or MATH 124</td>
<td>Calculus II</td>
<td></td>
</tr>
<tr>
<td>or MATH 126</td>
<td>Math and Calculus Applications for Life, Managerial, and Social Sci II</td>
<td></td>
</tr>
</tbody>
</table>

**Computer Science Core Requirement**

All computer science majors are required to complete the following 6 courses.

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 132</td>
<td>Introduction to Programming in Java</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 233</td>
<td>Introduction to Data Structures</td>
<td>4</td>
</tr>
<tr>
<td>CSDS 281</td>
<td>Logic Design and Computer Organization</td>
<td>4</td>
</tr>
<tr>
<td>CSDS 302</td>
<td>Discrete Mathematics</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 310</td>
<td>Algorithms</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 395</td>
<td>Senior Project in Computer Science</td>
<td>4</td>
</tr>
</tbody>
</table>

**Computer Science Breadth Requirement**

Computer science BA students are required to complete at least 3 of the 7 following computer science breadth courses.

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 314</td>
<td>Computer Architecture</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 325</td>
<td>Computer Networks I</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 338</td>
<td>Intro to Operating Systems and Concurrent Programming</td>
<td>4</td>
</tr>
<tr>
<td>CSDS 341</td>
<td>Introduction to Database Systems</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 345</td>
<td>Programming Language Concepts</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 391</td>
<td>Introduction to Artificial Intelligence</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 393</td>
<td>Software Engineering</td>
<td>3</td>
</tr>
</tbody>
</table>

**CS Courses**

In addition to the required core and breadth courses, computer science BA students must complete an additional CS Course so that the total number of CS Courses completed is at least 10. The remaining 3 courses to complete the major may come from this list of CS Courses or from the list of approved technical electives below.

The CS Courses are:

Any computer science core course.

Any computer science breadth course.

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 234</td>
<td>Structured and Unstructured Data</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 293</td>
<td>Software Craftsmanship</td>
<td>4</td>
</tr>
<tr>
<td>CSDS 312</td>
<td>Introduction to Data Science Systems</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 313</td>
<td>Introduction to Data Analysis</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 337</td>
<td>Compiler Design</td>
<td>4</td>
</tr>
<tr>
<td>CSDS 339</td>
<td>Web Data Mining</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 343</td>
<td>Theoretical Computer Science</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 344</td>
<td>Computer Security</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 356</td>
<td>Data Privacy</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 392</td>
<td>App Development for iOS</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 394</td>
<td>Introduction to Information Theory</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 405</td>
<td>Data Structures and File Management</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 427</td>
<td>Internet Security and Privacy</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 428</td>
<td>Computer Communications Networks II</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 433</td>
<td>Database Systems</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 435</td>
<td>Data Mining</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 438</td>
<td>High Performance Computing</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 440</td>
<td>Machine Learning</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 442</td>
<td>Causal Learning from Data</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 448</td>
<td>Smartphone Security</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 455</td>
<td>Applied Graph Theory</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 458</td>
<td>Introduction to Bioinformatics</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 459</td>
<td>Bioinformatics for Systems Biology</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 465</td>
<td>Computer Vision</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 477</td>
<td>Advanced Algorithms</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 491</td>
<td>Artificial Intelligence: Probabilistic Graphical Models</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 496</td>
<td>Artificial Intelligence: Sequential Decision Making</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 497</td>
<td>Artificial Intelligence: Statistical Natural Language Processing</td>
<td>3</td>
</tr>
<tr>
<td>CSDS 499</td>
<td>Algorithmic Robotics</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 484</td>
<td>Computational Intelligence I: Basic Principles</td>
<td>3</td>
</tr>
</tbody>
</table>

**List of Approved Technical Electives**

For Computer Science BA students, up to 3 of the 13 computer science and computer science related courses may come from the group 1 and group 3 lists. Computer science related courses not listed below may be used as a technical elective but require prior permission from the student's academic advisor.

**Group 1**

Any CSDS course.

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECSE 301</td>
<td>Digital Logic Laboratory</td>
<td>2</td>
</tr>
<tr>
<td>ECSE 303</td>
<td>Embedded Systems Design and Laboratory</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 317</td>
<td>Computer Design - FPGAs</td>
<td>3</td>
</tr>
<tr>
<td>ECSE 318</td>
<td>VLSI/CAD</td>
<td>4</td>
</tr>
<tr>
<td>ECSE 419</td>
<td>Computer System Architecture</td>
<td>3</td>
</tr>
</tbody>
</table>
**Suggested Program of Study: Computer Science BA**

The following is a suggested program of study. Current students should always consult their advisors and their individual graduation requirement plans as tracked in SIS (http://sis.case.edu/).

### First Year

<table>
<thead>
<tr>
<th>Units</th>
<th>Fall</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to Programming in Java (CSDS 132)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Breadth elective**</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Math and Calculus Applications for Life, Managerial, and Social Sci I (MATH 125)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>or Calculus I (MATH 123)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>or Calculus for Science and Engineering I (MATH 121)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>PHED (2 half semester courses)*</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SAGES First Year Seminar*</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Math and Calculus Applications for Life, Managerial, and Social Sci II (MATH 126)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>or Calculus II (MATH 124)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>or Calculus for Science and Engineering II (MATH 122)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Breadth elective**</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>PHED (2 half semester courses)*</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SAGES University Seminar*</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Year Total:</td>
<td>17</td>
<td>16</td>
</tr>
</tbody>
</table>

### Second Year

<table>
<thead>
<tr>
<th>Units</th>
<th>Fall</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Design and Computer Organization (CSDS 281)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Breadth elective**</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>SAGES University Seminar*</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Introduction to Data Structures (CSDS 233)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Discrete Mathematics (CSDS 302)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Breadth elective**</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Year Total:</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

### Third Year

<table>
<thead>
<tr>
<th>Units</th>
<th>Fall</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer science breadth coursea</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Computer science breadth coursea</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>SAGES Department Seminar</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Computer science breadth coursea</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Technical electiveb</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Technical electiveb</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Year Total:</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

### Fourth Year

<table>
<thead>
<tr>
<th>Units</th>
<th>Fall</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms (CSDS 310)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Technical electiveb</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Senior Project in Computer Science (CSDS 395)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Open elective</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Technical electiveb</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Year Total:</td>
<td>15</td>
<td>16</td>
</tr>
</tbody>
</table>

**Total Units in Sequence:** 120

**Hours Required for Graduation:** 120

- **University general education requirement**
- **Arts and Sciences general education requirement** (http://bulletin.case.edu/undergraduatestudies/casdegree/)


  b Chosen from the list of CS Courses or the list of approved technical electives. Any other course used as a technical elective must be approved by the student’s advisor.

**Cooperative Education Program in Computer Science**

Opportunities are available for students to alternate studies with work in industry or government as a co-op student, which involves paid full-
time employment over seven months (one semester and one summer). Students may work in one or two co-ops, beginning in the third year of study. Co-ops provide students the opportunity to gain valuable hands-on experience in their field by completing a significant engineering project while receiving professional mentoring. During a co-op placement, students do not pay tuition but maintain their full-time student status while earning a salary. Learn more at http://engineering.case.edu/coop/. Alternatively or additionally, students may obtain employment as summer interns.

**BS/MS Program in Computer Science**

Students with a grade point average of 3.2 or higher are encouraged to apply to the BS/MS Program which will allow them to get both degrees in five years. The BS can be in Computer Science or a related discipline, such as mathematics or electrical engineering. Integrating graduate study in computer science with the undergraduate program allows a student to satisfy all requirements for both degrees in five years. Review the Office of Undergraduate Studies BS/MS program requirements here (http://bulletin.case.edu/undergraduatetext/accelerationtowardgraduatedegreetext).

<table>
<thead>
<tr>
<th>Minor in Computer Science (BS or BSE)</th>
<th>For students pursuing a BS or BSE degree, the following four courses are required for a minor in computer science:</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 132</td>
<td>Introduction to Programming in Java</td>
</tr>
<tr>
<td>CSDS 233</td>
<td>Introduction to Data Structures</td>
</tr>
<tr>
<td>CSDS 302</td>
<td>Discrete Mathematics</td>
</tr>
<tr>
<td>CSDS 310</td>
<td>Algorithms</td>
</tr>
</tbody>
</table>

A student must take an additional 3 credit hours of CS Courses (see BA Major Requirements).

<table>
<thead>
<tr>
<th>Minor in Computer Science (BA)</th>
<th>For students pursuing BA degrees, the following courses are required for a minor in computer science:</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 132</td>
<td>Introduction to Programming in Java</td>
</tr>
<tr>
<td>CSDS 233</td>
<td>Introduction to Data Structures</td>
</tr>
<tr>
<td>MATH 125</td>
<td>Math and Calculus Applications for Life, Managerial, and Social Sci I</td>
</tr>
</tbody>
</table>

Two additional CS Courses (see BA Major Requirements) are required for this minor.

<table>
<thead>
<tr>
<th>Minor in Artificial Intelligence</th>
<th>The minor consists of five courses. Every student who takes the minor in artificial intelligence must take the two courses, ENGR 131 Elementary Computer Programming and CSDS 391 Introduction to Artificial Intelligence. Students who take the Artificial Intelligence minor must also take an additional three courses from one of two minor tracks.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Technology Track (requires 3 of the following courses):</th>
<th>BIOL 373</th>
<th>Introduction to Neurobiology</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDS 465</td>
<td>Computer Vision</td>
<td></td>
</tr>
<tr>
<td>CSDS 477</td>
<td>Advanced Algorithms</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cognitive Science Track (requires 3 of the following courses):</th>
<th>BIOL 373</th>
<th>Introduction to Neurobiology</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIOL 374</td>
<td>Neurobiology of Behavior</td>
<td></td>
</tr>
<tr>
<td>ENGL 301</td>
<td>Linguistic Analysis</td>
<td></td>
</tr>
<tr>
<td>PHIL 201</td>
<td>Introduction to Logic</td>
<td></td>
</tr>
<tr>
<td>PHIL 306</td>
<td>Mathematical Logic and Model Theory</td>
<td></td>
</tr>
<tr>
<td>PSCL 101</td>
<td>General Psychology I</td>
<td></td>
</tr>
<tr>
<td>PSCL 352</td>
<td>Physiological Psychology</td>
<td></td>
</tr>
<tr>
<td>PSCL 353</td>
<td>Psychology of Learning</td>
<td></td>
</tr>
<tr>
<td>PSCL 357</td>
<td>Cognitive Psychology</td>
<td></td>
</tr>
<tr>
<td>PSCL 402</td>
<td>Cognition and Information Processing</td>
<td></td>
</tr>
</tbody>
</table>

|-------------------------------|--------------------------------------------------------------------------------------------------|

**MS Degree Program (Computer Science)**

*The Master of Science in Computer Science has replaced the Master of Science in Computing and Information Science.*

MS in CS has three tracks: a Course-Focused track, a Project-Focused track, and a Thesis-Focused track. Although all of the three options require 30 semester hours of credit, they are structured differently to achieve different objectives. The Course-Focused track prepares students for advanced industry employment and should be treated as a terminal MS degree in CS. The Project-Focused track if for students who seek opportunities for completing an applied project, for example within the context of an established collaboration with industry. The Thesis-Focused track is mainly for students who have interests in research. Therefore, the three tracks have different requirements in admission, advising, and course work.

**Admission**

Graduate students shall be admitted to the MS degree program upon recommendation of the faculty of the CS program. Requirements
for admission include a strong record of scholarship in a completed bachelor's degree program in computer science and related areas, and fluency in written and spoken English. The University requires all foreign applicants to show English proficiency by achieving a TOEFL score of at least 90 on the internet-based exam for the thesis-focused or the project-focused track. For the course-focused track, a minimum TOEFL score of 80 is required. For students who are expected to have any professional student to student interaction, e.g., as a teaching assistant, a lab instructor, or a tutor, a minimum TOEFL score of 90 is required. It is required that all students submit original copies of GRE scores, with the exception of CWRU students applying to the BS/MS program.

The MS program requires students to have substantial knowledge of undergraduate computer science material. Applications from students with a bachelor’s degree in fields other than computer science may be granted admission on a provisional basis. Students should have knowledge equivalent to that in the courses:

- CSDS 233 Introduction to Data Structures
- CSDS 310 Algorithms
- Any one course listed as an undergraduate Computer Science Breadth Requirement

Students deficient in one or more of these areas (admission with provision) may be required to satisfy this requirement by taking the corresponding courses listed above. These courses cannot be counted towards their MS requirement. However, a student taking and passing a course that subsumes one of the requirements automatically demonstrates knowledge of the material in the required course; e.g. taking CSDS 410 Analysis of Algorithms demonstrates knowledge of the material in CSDS 310 Algorithms. Such graduate level courses will be used to satisfy their MS requirement.

Applicants lacking the required background are encouraged to explore the Computer Science pathway options. Additional information on the Computer Science pathway can be obtained by contacting the department.

**Registration**
Course registration can be performed through the SIS system. Each semester before registration, students should update any personal information that may have changed by logging onto the SIS and editing the appropriate information. All registration holds must be lifted in order to successfully complete the registration process.

**Advising**
Each MS student will be assigned an academic advisor, who will assist the student in formulating an academic program. A student in the thesis-focused or project-focused track is expected to pick a research advisor in the program by the end of their first semester. The research advisor will supervise the student's thesis or project and also serves as the academic advisor. A student whose research advisor is a faculty member not in the CDS department must maintain an academic advisor in the CS program. Each student, in consultation with their advisor(s), must submit a Planed Program of Study preferably before completing 9 credit hours of coursework. This should specify all courses and thesis/project work that will be counted toward the 30 credit hour requirement.

**Requirements of different tracks**
The Course-Focused MS degree program requirements consist of the completion of 30 hours of approved coursework, satisfactory completion of a comprehensive exam, i.e., passing the course ENGR 600 with a grade of “P”. ENGR 600 consists of Comprehensive Exam questions that are administered in CSDS 410 Analysis of Algorithms, CSDS 425 Computer Networks I, CSDS 440 Machine Learning, CSDS 444 Computer Security, and CSDS 493 Software Engineering. Students must take and pass questions in at least two of these classes. Students who fail one exam in a course may retake that exam one more time but are not required to retake the associated course.

The Project-Focused track requires 24 semester hours of coursework and 6 semester hours of project (CSDS 695 Project M.S.).

The Thesis-Focused track requires 18 semester hours of coursework credit and 12 hours of thesis (CSDS 651 Thesis M.S.). A BS/MS student is required to choose the thesis-focused track initially. Both the Thesis-Focused and the Project-Focused track require a formal written report, as well as a final oral examination by a committee of at least three faculty members, two of whom must be primarily affiliated with the CS program. The academic advisor is normally one of the committee members. For Project-Focused track students, the oral examination fulfills the Comprehensive Examination requirement of the School of Graduate Studies.

If a student wishes to switch from one track to another, the following requirements apply:

- Deadline. In each semester, students must request to switch track one week before the date at which Drop/Add ends, as stated in the academic calendar.
- Course-only or Project to Thesis. A course-only student may request to switch to the thesis track only if she (1) has already taken at least 9 credit hours of letter graded CSDS courses and (2) has a GPA of 3.5 or higher and (3) has a TOEFL score of 90 or higher and (4) has the recommendation of a CDS advisor or (co)advisor.
- Course-only to Project. A course-only student may request to switch to the thesis track only if she (1) has a TOEFL score of 90 or higher and (2) has the recommendation of a CDS advisor or (co)advisor.
- Thesis to Project, or Thesis or Project to Course-only. Such a transfer needs approval from the student's advisor and the department chair.
- Petition. If a student fails to satisfy the transfer requirements, a petition may be submitted by a CDS advisor or (co)advisor to the department chair. In no case, petitions may be submitted by non-CDS faculty members or by students.

Students should consult with their academic advisor and/or department to determine the detailed requirements within this framework.

**Course Requirements**
For all three tracks, at least 18 hours of coursework must be at the 400 level or above. All students are required to have specialized knowledge in at least one of the following depth areas, by taking at least three graduate-level classes from that area. The list of acceptable classes is shown below. For research or project-focused tracks, the chosen area should correspond to the student's thesis research area or project in general. CSDS 600 Special Topics classes will also qualify in this category with approval from the student's advisor. The remaining classes can be (i) any other class from the classes listed below, or (ii) any letter graded CSDS class (see note below), or (iii) at most two graduate-level classes other than those in category (i) and (ii) (such as non-letter-graded graduate CSDS classes or graduate classes in other departments).
(Note: The Graduate School and the School of Engineering limit the number of undergraduate courses that can be taken for credit by Master students.)

Students should discuss their courses with their advisor every semester prior to registration. Students must achieve a grade point average of 3.0 or higher; it is computed for all of the letter-graded courses on the student's academic program.

**List of depth areas and corresponding courses**

1. **Algorithms & Theory:**
   a. CSDS 410 Analysis of Algorithms
   b. CSDS 440 Machine Learning
   c. CSDS 455 Applied Graph Theory
   d. CSDS 456 Data Privacy
   e. CSDS 477 Advanced Algorithms
   f. MATH 408 Introduction to Cryptology

2. **Artificial Intelligence:**
   a. CSDS 440 Machine Learning
   b. CSDS 442 Causal Learning from Data
   c. CSDS 455 Applied Graph Theory
   d. ECSE 484 Computational Intelligence I: Basic Principles
   e. CSDS 491 Artificial Intelligence: Probabilistic Graphical Models
   f. CSDS 496 Artificial Intelligence: Sequential Decision Making
   g. CSDS 497 Artificial Intelligence: Statistical Natural Language Processing
   h. CSDS 499 Algorithmic Robotics
   i. CSDS 465 Computer Vision

3. **Bioinformatics:**
   a. CSDS 410 Analysis of Algorithms
   b. CSDS 435 Data Mining
   c. CSDS 440 Machine Learning
   d. CSDS 456 Data Privacy
   e. EECS 458 Introduction to Bioinformatics
   f. EECS 459 Bioinformatics for Systems Biology
   g. SYBB 412 Survey of Bioinformatics: Programming for Bioinformatics

4. **Computer Networks and Systems:**
   a. CSDS 427 Internet Security and Privacy
   b. ECSE 414 Wireless Communications
   c. CSDS 425 Computer Networks I
   d. CSDS 428 Computer Communications Networks II
   e. CSDS 438 High Performance Computing
   f. CSDS 444 Computer Security

5. **Databases and Data Mining:**
   a. CSDS 405 Data Structures and File Management
   b. CSDS 433 Database Systems
   c. CSDS 435 Data Mining
   d. CSDS 439 Web Data Mining
   e. CSDS 440 Machine Learning
   f. STAT 426 Multivariate Analysis and Data Mining
   g. PQHS 471 Machine Learning & Data Mining

6. **Security and Privacy:**
   a. CSDS 427 Internet Security and Privacy
   b. CSDS 444 Computer Security
   c. CSDS 448 Smartphone Security
   d. CSDS 456 Data Privacy
   e. CSDS 493 Software Engineering
   f. MATH 408 Introduction to Cryptology

7. **Software Engineering:**
   a. CSDS 425 Computer Networks I
   b. CSDS 433 Database Systems
   c. CSDS 438 High Performance Computing
   d. CSDS 442 Causal Learning from Data
   e. CSDS 444 Computer Security
   f. CSDS 448 Smartphone Security
   g. CSDS 493 Software Engineering

**PhD Degree Program (Computer Science)**

*The Doctor of Philosophy in Computer Science has replaced the Doctor of Philosophy in Computing and Information Science.*

**Admission**

Requirements for admission include a strong record of scholarship in a completed bachelor's degree program in computer science and related areas, and fluency in written and spoken English. The University requires all foreign applicants to show English proficiency by achieving a TOEFL score of at least 577 on the paper-based exam or 90 on the internet-based exam. It is required that all students submit original copies of GRE scores. When applying to this program, please select the Computer Science option in your application.

**Registration**

Course registration (including submission of the Academic Program) can be performed through SIS. Each semester before registration, students should update any personal information that may have changed by logging onto the SIS and editing the appropriate information. All registration holds must be lifted in order to successfully complete the registration process.

**PhD Requirements**

Each student must satisfy requirements in the following categories:

- Course Work
- Mathematics and Science Requirement
- Research Proposal
- Qualifying Examination
- Dissertation

All programs of study must contain at least 36 hours of courses past the undergraduate degree. All courses must be at the 400-level or higher. Six hours must be in a basic science or mathematics. A minimum of 12 hours must be in courses outside the student’s thesis area. A student must have attained a minimum 3.25 grade point average (GPA) at the time of graduation. The minimum GPA is calculated based on all courses in the student’s Program of Study that carry quality points.

These regulations are in addition to the Academics Regulations of the School of Graduate Studies and the Specific Requirements for the PhD Degree of the Graduate Program in the Case School of Engineering as found in the General Bulletin of Case Western Reserve University.
Academic Advisor and Research Advisor
Upon arrival, each graduate student is assigned an academic advisor from the Computer Science (CS) program faculty, typically the CS graduate representative. By the end of the second semester of study, a PhD student should transfer to a research advisor who may or may not be the same as the academic advisor. Once a research advisor is selected, the research advisor will also serve as the academic advisor, with whom a student consults to ensure that the balance of the PhD course work constitutes a coherent program of study.

If the research advisor is not from the Computer Science program, the student must have an academic advisor from the CS program. The academic advisor may serve as a research co-advisor if so desired by the student and the research advisor.

Academic Program
Each PhD student must submit an Academic Program detailing his or her course work, Qualifying Examination and dissertation schedules. The Academic Program lists all courses taken beyond the undergraduate degree and shows how these courses satisfy the following course requirements for the PhD:

1. The minimum course requirement beyond the BS level is 36 credit hours of courses taken for credit, at least 18 hours of which must be taken at CWRU. The following courses taken for credit will be acceptable:
   • All 400, 500, 600 level courses.
   • Graduate level courses taken at other institutions approved by the student’s advisor.

The above courses must include the following:
   • 1.1 - A minimum of 12 credit hours in the student’s dissertation research area.
   • 1.2 - A minimum of 6 credit hours in mathematics or basic science.
   • 1.3 - A minimum of 12 credit hours of breadth courses that are not in the student’s dissertation research area.
     • Note: The courses for items 1.1, 1.2, and 1.3 must be disjoint.
   • 1.4 - A minimum of 18 credit hours of courses approved by the Computer Science program which can be courses used in item 1.1 and item 1.3. These approved Computer Science courses are listed below.

2. A minimum of 18 credit hours of CSDS 701 Dissertation Ph.D.

3. The cumulative grade-point average of all CWRU courses on the program of study must be at least 3.25.

4. All PhD students are required to register for and pass CSDS 400T Graduate Teaching I, CSDS 500T Graduate Teaching II and CSDS 600T Graduate Teaching III which are 0 credit hour courses that provide students with teaching experience.

5. All PhD students are required to register for and pass three semesters of “CSDS 500 CSDS Colloquium” (a 0 credit hour course) before advancement to candidacy and complete a public presentation of their work.

The Academic Program must be approved by the student’s academic advisor and the chair of the CDS Department, and it must be approved before the student advances to candidacy.

Qualifying Examination
The Computer Science PhD qualifying examination is in the form of an oral exam and a written report, assessing the student’s ability to survey a specific research topic, discuss the state-of-the-art in depth, provide a critical description of the literature, and propose creative ideas on improving the state-of-the-art. The written and oral parts of the exam are administered by a committee consisting of three faculty members. The exam is supplemented by the evaluation of the student’s course performance and the final decision is made in a meeting of faculty affiliated with the CS program.

Students who hold a MS degree are required to take the qualifying exam at the start of their 3rd semester, and pass the qualifying exam before the start of their 4th semester in the PhD program. Students who do not hold a MS degree are required to take the qualifying exam at the start of their 5th semester, and pass the qualifying exam before the start of their 6th semester in the PhD program.

Committee: The exam will be administered by a committee composed of 3 faculty members, at least 2 of them being faculty members whose primary affiliation is Computer Science. The student will provide 0 to 3 names as suggested committee members and the CS Graduate Committee will appoint the committee, taking into account the student’s suggestion and maintenance of the load balance of the faculty. If the student has research advisors at the time of the exam, then one advisor must be a member of the committee. Multiple advisors cannot serve on the committee, and no advisors can serve as the chair of the committee. The CS Graduate Committee will make every effort to include a faculty member outside the research area in the committee. The committee must be appointed at least 2 months prior to the exam.

Format: The student will select a research area from the following list:

1. Algorithms and Theory
2. Artificial Intelligence
3. Bioinformatics
4. Computer Networks and Systems
5. Databases and Data Mining
6. Security and Privacy
7. Software Engineering

The exam committee will ask the student to write a report that adequately demonstrates the student’s ability to perform research in their chosen area of research. Specific examples of this might be a survey of a broad area and/or approaches to a specific problem in the area. This will be followed by an oral examination.

Written Report: The student will submit a written report on the specific research problem. The report has to adequately describe the problem and justify its relevance, identify the challenges associated with the problem, provide a broad classification of existing approaches, point out their key differences and trade-offs, identify limitations, and propose solutions for these limitations. The report must be at most 10 pages in length (11 pt font, single-spaced, single column, 1” margins) and use illustrative figures, tables, and other visual material to communicate key ideas. In addition, the report must include a comprehensive list of references. The written report must be submitted to the exam committee three weeks before the date of the oral exam.

Oral Exam: In the oral exam, the student will answer questions by the committee members on a specific research problem chosen by the committee in the selected research area, assessing the knowledge, technical depth, and broader vision of the student on the problem. The
committee may also ask questions on the fundamentals of computer science as they relate to the specific research problem. The student may prepare slides in advance that will help answer questions or use the chalk board (or both), but the exam will not be in the format of a presentation. The exam will be one hour in duration. All students who are taking the exam for the first time have to take their oral exam in the week prior to the beginning or in the first two weeks of the Fall semester.

**Scoring:** Each of the three committee members will prepare a report rating the student’s exam performance according to the following criteria:

- **Fundamentals:** Does the student have broad knowledge of fundamental concepts in computer science that will enable the student to understand and tackle the challenges in the specific research area?
- **Knowledge of Chosen Area:** Does the student have sufficient technical depth and command of the key challenges and the state-of-the-art in the chosen area of research?
- **Vision:** Does the student demonstrate a solid understanding of the relevance of the problem in the context of scientific progress and societal needs? Does the student show creativity in innovating their chosen area of research?
- **Communication:** Can the student explain the concepts in an accessible and comprehensible manner and handle questions effectively?
- **Overall Score for the Written Report and the Oral Exam.**

Possible ratings are 2 (Pass), 1 (Retake), or 0 (Fail).

**Course Work:** The student’s performance in all courses taken before the exam will also be considered by the CS program faculty in making a decision. At the time of the exam, the student must have completed at least two 400-level Computer Science courses with a "B" or above. One of these courses must be CSDS 410 Analysis of Algorithms or CSDS 477 Advanced Algorithms. The second course must be relevant to their chosen area of research. The relevance of the additional course to the research area is subject to approval by the CS Graduate Studies Committee. The following courses are pre-approved for each area:

1. **Algorithms and Theory:**
   - CSDS 410 Analysis of Algorithms
   - CSDS 477 Advanced Algorithms
2. **Artificial Intelligence:**
   - CSDS 440 Machine Learning
   - CSDS 491 Artificial Intelligence: Probabilistic Graphical Models
3. **Bioinformatics:**
   - CSDS 458 Introduction to Bioinformatics
   - CSDS 459 Bioinformatics for Systems Biology
4. **Computer Networks and Systems:**
   - CSDS 425 Computer Networks I
   - CSDS 428 Computer Communications Networks II
   - CSDS 427 Internet Security and Privacy
5. **Databases and Data Mining:**
   - CSDS 433 Database Systems
   - CSDS 435 Data Mining
6. **Security and Privacy:**
   - CSDS 427 Internet Security and Privacy
   - CSDS 444 Computer Security
   - CSDS 448 Smartphone Security
   - CSDS 456 Data Privacy
7. **Software Engineering:**
   - CSDS 493 Software Engineering

**Outcome:** The final decision will be made by the CS program faculty based on the committee’s reports and the student’s coursework. The outcome of the exam will be one of Pass (the student advances to candidacy), Retake (the student has to retake the exam once more before the end of the following semester. The Retake decision can be partial, i.e. the student may be asked to retake the oral exam only, rewrite the report only, or take/retake a course), Fail (the student will be separated from the PhD program). The decision will be documented by the CS graduate chair and the student will be sent a notification letter.

**Advancement to Candidacy**
A student formally advances to candidacy after passing the Qualifying Examination and finding a faculty member who agrees to be the student’s research advisor. The student should advance to candidacy within one semester of passing the Qualifying Examination.

Students should submit documentation, approved by the academic and research advisor(s), to the Chairman of the Graduate Studies Committee of Computer Science to be admitted to candidacy.

Students who have failed to complete the conditions above within the time limit will be separated from the PhD program. Separation may also occur in the event of failure of the student to maintain a satisfactory GPA. A student who has been separated may not undertake further study for credit toward the PhD degree. With the approval of the Department and the Dean of Graduate Studies, such a student may complete a master’s degree, may register as a non-degree student or seek admission to the graduate program of another department.

**Dissertation Advisory Committee**
Each PhD student must form a Dissertation Advisory Committee which consists of at least 4 members of University faculty. The student’s academic advisor serves as the chair of this committee. Both the chair of the committee and at least one other member must be a regular faculty member whose primary affiliation is with the Computer Science program. The committee must also include one member whose primary appointment is not in the Computer Science program.

**Dissertation Proposal**
The PhD student must write a formal thesis proposal and defend it in an oral presentation to his or her Dissertation Advisory Committee. Normally this is done within a year of advancing to candidacy. A student who fails to defend his or her thesis proposal can attempt to defend it a second time after modifying the thesis proposal, but a second failure will cause the student to be separated from the program.

**Dissertation**
The student’s dissertation must be original research in CS which represents a significant contribution to existing knowledge in the student’s research area, a portion of which must be suitable for publication in reputable research journals or selective peer-reviewed conferences. In addition to the written dissertation, the doctoral candidate must pass an oral examination in defense of the dissertation. The Dissertation Advisory Committee is responsible for certifying that the quality and suitability of the material presented in the dissertation meet acceptable scholarly standards. If the student has not publicly presented their work at a conference or similar external venue, they must also present the dissertation research in a departmental seminar.
Course List for Program of Study Requirement
CSDS 405 Data Structures and File Management
CSDS 410 Analysis of Algorithms
CSDS 425 Computer Networks I
CSDS 427 Internet Security and Privacy
CSDS 428 Computer Communications Networks II
CSDS 433 Database Systems
CSDS 435 Data Mining
CSDS 438 High Performance Computing
CSDS 439 Web Data Mining
CSDS 440 Machine Learning
CSDS 442 Causal Learning from Data
CSDS 444 Computer Security
CSDS 448 Smartphone Security
CSDS 455 Applied Graph Theory
CSDS 456 Data Privacy
CSDS 458 Introduction to Bioinformatics
CSDS 459 Bioinformatics for Systems Biology
CSDS 466 Computer Graphics
CSDS 477 Advanced Algorithms
ECSE 484 Computational Intelligence I: Basic Principles
CSDS 491 Artificial Intelligence: Probabilistic Graphical Models
CSDS 493 Software Engineering
CSDS 496 Artificial Intelligence: Sequential Decision Making
CSDS 497 Artificial Intelligence: Statistical Natural Language Processing
CSDS 499 Algorithmic Robotics

Colloquium and Presentation Requirement
PhD Students
The requirement has two parts: (1) passing 3 semesters of CSDS 500 CSDS Colloquium, and (2) a public presentation.

(1) All PhD students are required to register for and pass CSDS 500 CSDS Colloquium for a total of three semesters of the PhD Program, and this is expected before Advancement to Candidacy. (This is a 0 credit hour required course.) Students (such as students working in industry) may propose an alternative arrangement for fulfilling this requirement, by submitting a written petition to the Graduate Studies Committee.

(2) All PhD students must give a public presentation of their research, in addition to their PhD dissertation defense. This presentation must be given either at a research conference or in the CDS seminar series.

Facilities

Computer Facilities
The department computer facilities incorporate both Unix/Linux and Microsoft Windows-based operating systems on high-end computing workstations for education and research. A number of file, printing, database, and authentication servers support these workstations, as well as the administrative functions of the department. Labs are primarily located in the Olin and Glennan buildings, but include Nord Hall, and are networked via the Case network.

The Case network is a state-of-the-art, high-speed fiber optic campus-wide computer network that interconnects laboratories, faculty and student offices, classrooms, and student residence halls. It is one of the largest fiber-to-desktop networks anywhere in the world. Every desktop has a 1 Gbps (gigabit per second) connection to a fault-tolerant 10 Gbps backbone. To complement the wired network, over 1,200 wireless access points (WAPs) are also deployed allowing anyone with a laptop or wireless enabled PDA to access resources from practically anywhere on campus.

Off-campus users, through the use of virtual private network (VPN) servers, can use their broadband connections to access many on-campus resources, as well as software, as if they were physically connected to the Case network. The department and the university participate in the Internet2 and National Lambda Rail projects, which provides high-speed, inter-university network infrastructure allowing for enhanced collaboration between institutions. The Internet2 infrastructure allows students, faculty and staff alike the ability to enjoy extremely high-performance connections to other Internet2 member institutions.

Aside from services provided through a commodity Internet connection, Case network users can take advantage of numerous online databases such as EUCLIDplus, the University Libraries’ circulation and public access catalog, as well as Lexus-Nexis™ and various CD-ROM based dictionaries, thesauri, encyclopedias, and research databases. Many regional and national institutional library catalogs are accessible over the network, as well.

Additional Department Facilities

Jennings Computer Center and Undergraduate Computer Lab
Supported by an endowment from the Jennings Foundation, this lab provides our students with the educational resources necessary for their classwork and exploration of the art of computing. This lab has both PCs and Linux/Unix workstations and includes two high-speed laser printers. This laboratory on the 8th floor of the Olin building supports the freshman computing classes: ENGR 131 Elementary Computer Programming and CSDS 132 Introduction to Programming in Java, as well as other classes that have a lab component.

Nord Computer Laboratory
This is a general-purpose computer facility that is open 24 hours a day, to all students. The lab contains 50 PCs running Windows and four Apple Macintosh computers. Facilities for color printing, faxing, copying and scanning are provided. Special software includes PRO/Engineer, ChemCAD and Visual Studio. Visit the website (https://engineering.case.edu/it/nord-computer-lab/) for more information.

Virtual Worlds (Gaming and Simulation) Laboratory
The Virtual Worlds Gaming and Simulation Laboratory provides software and hardware to support education and research in computer gaming and
simulation activities within the Computer and Data Sciences Department and the University at large. The lab has been leveraged to provide students with extensive gameplay opportunities and excellent, strongly experiential simulation and game development educational opportunities – primarily targeted to the CDS undergraduate population.

The lab also stimulates large amounts of cross-disciplinary collaboration in both education and research. Simulation and visualization techniques are of great value in all science and engineering fields, and the lab is capable of supporting advanced applications of these techniques in real-time applications. In addition, interactive technologies and video games require substantial artistic resources, which has resulted in excellent opportunities for educational and research collaboration with the Cleveland Institute of Art (CIA), the School of Nursing, the Medical School, and the Psychology Department. Of particular note has been the Advanced Game Project course (CSDS 390 Advanced Game Development Project) taught jointly by CWRU and CIA for juniors and seniors. This course has been very popular and has provided truly excellent student game design and production experiences while receiving industrial and popular recognition and acclaim. In addition, an entry-level computer game programming course (CSDS 290 Introduction to Computer Game Design and Implementation) is available for students who have taken both a Java-based programming course and a data structures course to provide an introduction to many of the technical aspects of computer game development. Many other courses in the department also use the lab as an important part of their curriculum including courses on computer graphics, artificial intelligence, simulation, digital signal processing, and control systems. The lab also supports research in the department requiring significant computational resources, e.g. GPU acceleration, VLSI simulation, etc.

A recent large donation for the lab has allowed for the update and renovation of the entire lab including the physical infrastructure (carpeting, furniture, etc.), the gaming PCs, and the gaming consoles. In addition, a new VR and AR room has been added to represent this new area connected strongly to computer gaming. The lab is now structured into a PC gaming area and an adjacent gaming console area, a VR/AR room, a portable gaming development room, and a team collaboration room.

The renovated lab includes the following primary equipment:

- 24 New Alienware PCs with Dell 27” 4K monitors
- 4 Sony Bravia Television monitors 75” 3DTV
- 2 Microsoft HoloLens AR Units
- 4 Oculus Rift VR units with Haptic Touch Input devices
- A 3D projector (and large wall screen) with 3D capability for common presentations
- 4 Xbox One Units with Xbox One controllers
- 4 PS4 Sony PlayStation units with controllers

**Sally & Larry Sears Undergraduate Design Laboratory**

This laboratory supports CDS/ECSE courses in circuits and includes a state-of-the-art lecture hall, a modernistic glass-walled lab, an electronics "store", and a student lounge and meeting area. Specialized lab space is available for senior projects and sponsored undergraduate programs. The lab is open to all undergraduates, and components are provided free of charge, so students can "play and tinker" with electronics and foster innovation and creativity. The laboratory provides access to PCs, oscilloscopes, signal generators, logic analyzers, and specialized equipment such as RF analyzers and generators. In addition, the lab includes full-time staff dedicated to the education, guidance and mentoring of undergraduates in the “art and practice” of hands-on engineering.

This is the central educational resource for students taking analog, digital, and mixed-signal courses in electronics, and has been supported by various corporations in addition to alumnus Larry Sears, a successful engineer and entrepreneur. Basic workstations consist of Windows-based computers equipped with LabView software, as well as Agilent 546xx oscilloscopes, 33120A Waveform Generators, 34401A Digital Multimeters, and E3631A power supplies. Advanced workstations are similarly configured, but with a wider variety of high-performance test equipment.

**Courses**

**CSDS 132. Introduction to Programming in Java. 3 Units.**

An introduction to modern programming language features, computer programming and algorithmic problem solving with an emphasis on the Java language. Computers and code compilation; conditional statements, subprograms, loops, methods; object-oriented design, inheritance and polymorphism, abstract classes and interfaces; types, type systems, generic types, abstract data types, strings, arrays, linked lists; software development, modular code design, unit testing; strings, text and file I/ O; GUI components, GUI event handling; threads; comparison of Java to C, C++, and C#. Offered as CSDS 132 and ECSE 132. Counts for CAS Quantitative Reasoning Requirement.

**CSDS 133. Introduction to Data Science and Engineering for Majors. 3 Units.**

This course is an introduction to data science and analytics. In the first half of the course, students will develop a basic understanding of how to manipulate, analyze and visualize large data in a distributed computing environment, with an appreciation of open source development, security and privacy issues. Case studies and team project assignments in the second half of the course will be used to implement the ideas. Topics covered will include: Overview of large scale parallel and distributed (cloud) computing; file systems and file i/o; open source coding and distributed versioning, data query and retrieval; basic data analysis; visualization; data security, privacy and provenance. Prereq: ENGR 131 or EECS 132.

**CSDS 233. Introduction to Data Structures. 4 Units.**

Different representations of data: lists, stacks and queues, trees, graphs, and files. Manipulation of data: searching and sorting, hashing, recursion and higher order functions. Abstract data types, templating, and the separation of interface and implementation. Introduction to asymptotic analysis. The Java language is used to illustrate the concepts and as an implementation vehicle throughout the course. Offered as CSDS 233 and ECSE 233. Prereq: CSDS 132 or ECSE 132.
CSDS 234. Structured and Unstructured Data. 3 Units.
This course is an introduction to types of data and their representation, storage, processing and analysis. The course has three parts. In the first part of the course, students will develop a basic understanding and the ability to represent, store, process and analyze structured data. Structured data include catalogs, records, tables, logs, etc., with a fixed dimension and well-defined meaning for each data point. Suitable representation and storage mechanisms include lists and arrays. Relevant techniques include keys, hashes, stacks, queues and trees. In the second part of the course, students will develop a basic understanding and the ability to represent, store, process and analyze semi-structured data. Semi-structured data include texts, web pages and networks, without a dimension and structure, but with well-defined meaning for each data point. Suitable representation and storage mechanisms include trees, graphs and RDF triples. Relevant techniques include XML, YAML, JSON, parsing, annotation, language processing. In the third part of the course, students will develop a basic understanding and the ability to represent, store, process and analyze unstructured data. Unstructured data include images, video, and time series data, without neither a fixed dimension and structure, nor well-defined meaning for individual data points. Suitable representation and storage mechanisms include large matrices, EDF, Dicom. Relevant techniques include feature extraction, segmentation, clustering, rendering, indexing, and visualization. Prereq: CSDS 133.

CSDS 236. Introduction to C/C++ Programming. 1 Unit.
This course provides an introduction to C and C++ programming languages. Prior knowledge of these languages is not expected. However, students are expected to know at least one programming language prior to the class. The lectures will cover low-level programming techniques and object-oriented class design. Students will learn specific topics in C/C++ including required background knowledge, pointers & references, memory management, inheritance, polymorphism, templates, exceptions, use of debugging and compiler tools, and others as time permits. Prereq: CSDS 132.

CSDS 275. Fundamentals of Robotics. 4 Units.
The Fundamentals of Robotics course will expose students to fundamental principles of robotics. Students will explore high level conceptual foundations of robotics beginning with Braitenberg vehicles and apply this knowledge to simulated and physical robot hardware in laboratory experiences and in a final project. Laboratory experiences will guide students through applying theory to practice increasingly complex tasks in a project oriented, group work environment. The course culminates in a robotics challenge project at the end of the semester. Topics covered are: sensors, actuators, kinematics, control, planning and programming. Programming languages and concepts (e.g., C++, object oriented programming) used in robotics will be introduced and used with modern robotics programming toolboxes and frameworks. Prior experience with these languages will not be necessary. Previous experience with robotics is not required for this course. Offered as CSDS 275 and ECSE 275. Prereq: (ENGR 131 or ECES 132) and PHYS 121 and MATH 121.

CSDS 281. Logic Design and Computer Organization. 4 Units.
Fundamentals of digital systems in terms of both computer organization and logic level design. Organization of digital computers; information representation; boolean algebra; analysis and synthesis of combinational and sequential circuits; datapaths and register transfers; instruction sets and assembly language; input/output and communication; memory. Offered as CSDS 281 and ECSE 281. Prereq: ENGR 131 or CSDS 132.

CSDS 285. Linux Tools and Scripting. 3 Units.
A fast paced introduction to linux that brings the project-oriented student from little or no linux experience to be fully capable of using scripting languages as tools in a linux command-line environment. This includes systems administration and agile web services with server-side scripting and server-side management. This also involves data cleaning, scraping, wrangling, and processing user data with regular expressions. Current main languages include php, bash, awk, with some client-side javascript, and very brief looks at tcl/lua, powershell, and python. Tools include vim, make, grep/sed, and curl/wget. Recommended preparation: one year of computer programming and familiarity with raw HTML/CSS. Prereq: CSDS 132.

CSDS 290. Introduction to Computer Game Design and Implementation. 3 Units.
This class begins with an examination of the history of video games and of game design. Games will be examined in a systems context to understand gaming and game design fundamentals. Various topics relating directly to the implementation of computer games will be introduced including graphics, animation, artificial intelligence, user interfaces, the simulation of motion, sound generation, and networking. Extensive study of past and current computer games will be used to illustrate course concepts. Individual and group projects will be used throughout the semester to motivate, illustrate and demonstrate the course concepts and ideas. Group game development and implementation projects will culminate in classroom presentation and evaluation. Offered as CSDS 290 and ECSE 290. Prereq: EECS 132 or CSDS 132.

CSDS 293. Software Craftsmanship. 4 Units.
A course to improve programming skills, software quality, and the software development process. Software design; Version control; Control issues and routines; Pseudo-code programming process and developer testing; Defensive programming; Classes; Debugging; Self-documenting code; Refactoring. Offered as CSDS 293 and CSDS 293N. Prereq: Computer Science Major and (CSDS 233 or ECSE 233 or ECES 233) with a C or higher.

CSDS 293N. Software Craftsmanship. 4 Units.
A course to improve programming skills, software quality, and the software development process. Software design; Version control; Control issues and routines; Pseudo-code programming process and developer testing; Defensive programming; Classes; Debugging; Self-documenting code; Refactoring. Offered as CSDS 293 and CSDS 293N. Prereq: (CSDS 233 or ECSE 233 or EEC 233) with a C or higher.

CSDS 296. Independent Projects. 1 - 3 Units.
Independent projects in Computer and Data Science. Recommended preparation: ENGR 131 or EECS/CSDS/ECES 132. Prereq: Limited to freshmen and sophomore students.

CSDS 297. Special Topics. 1 - 3 Units.
Special topics in Computer and Data Science. Prereq: Limited to freshmen and sophomores.

CSDS 302. Discrete Mathematics. 3 Units.
A general introduction to basic mathematical terminology and the techniques of abstract mathematics in the context of discrete mathematics. Topics introduced are mathematical reasoning, Boolean connectives, deduction, mathematical induction, sets, functions and relations, algorithms, graphs, combinatorial reasoning. Offered as CSDS 302, ECSE 302 and MATH 304. Prereq: MATH 122 or MATH 124 or MATH 126.
CSDS 305. Files, Indexes and Access Structures for Big Data. 3 Units.
Database management become a central component of a modern computing environment, and, as a result, knowledge about database systems has become an essential part of education in computer science and data science. This course is an introduction to the nature and purpose of database systems, fundamental concepts for designing, implementing and querying a database and database architectures.

Objectives: -An expert knowledge of basic data structures, basic searching, sorting, methods, algorithm techniques, (such as greedy and divide and conquer) -In-depth knowledge on Search and Index Structures for large, heterogeneous data including multidimensional data, high dimensional data and data in metric spaces (e.g., sequences, images), on different search methods (e.g. similarity searching, partial match, exact match), and on dimensionality reduction techniques. Prereq: CSDS 234 or ECSE 233.

CSDS 310. Algorithms. 3 Units.
Fundamentals in algorithm design and analysis. Loop invariants, asymptotic notation, recurrence relations, sorting algorithms, divide-and-conquer, dynamic programming, greedy algorithms, basic graph algorithms. Offered as CSDS 310 and CSDS 310N. Prereq: (Computer Science Major/Minor or Data Science Major or Biomedical Engineering Major on the Computing and Analysis Track) and (CSDS 302 or ECSE 302 or MATH 304 with a C or higher) and (CSDS 233 or ECSE 233 with a C or higher).

CSDS 310N. Algorithms. 3 Units.
Fundamentals in algorithm design and analysis. Loop invariants, asymptotic notation, recurrence relations, sorting algorithms, divide-and-conquer, dynamic programming, greedy algorithms, basic graph algorithms. Offered as CSDS 310 and CSDS 310N. Prereq: (CSDS 302 or ECSE 302 or MATH 304) with a C or higher and (CSDS 233 or ECSE 233) with a C or higher.

CSDS 312. Introduction to Data Science Systems. 3 Units.
An introduction to the software and hardware architecture of data science systems, with an emphasis on Operating Systems and Computer Architecture that are relevant to Data Sciences systems. At the end of the course, the student should understand the principles and architecture of storage systems, file systems (especially, HDFS), memory hierarchy, and GPU. The student should have carried out projects in these areas, and should be able to critically compare various design decisions in terms of capability and performance. Prereq: CSDS 132.

CSDS 313. Introduction to Data Analysis. 3 Units.
In this class we will give a broad overview of data analysis techniques, covering techniques from data mining, machine learning and signal processing. Students will also learn about probabilistic representations, how to conduct an empirical study and support empirical hypotheses through statistical tests, and visualize the results. Course objectives: -Expose students to different analysis approaches. -Understand probabilistic representations and inference mechanisms. -Understand how to create empirical hypotheses and how to test them. Prereq: EECS 340 and DSCI 234.

CSDS 314. Computer Architecture. 3 Units.
This course provides students the opportunity to study and evaluate a modern computer architecture design. The course covers topics in fundamentals of computer design, performance, cost, instruction set design, processor implementation, control unit, pipelining, communication and network, memory hierarchy, computer arithmetic, input-output, and an introduction to RISC and super-scalar processors. Offered as CSDS 314 and ECSE 314. Prereq: EECS 281.

CSDS 315. Scalable Parallel Data Analysis. 3 Units.
This course provides an introduction to scalable and parallel data analysis using the most common frameworks and programming tools in the age of big data. Covered topics include parallel programming models, parallel hardware architectures, multi-threaded, multi-core programming, cluster computing and GPU programming. The course is designed to provide a heavily hands-on experience with several programming assignments. Prereq: CSDS 312.

CSDS 325. Computer Networks I. 3 Units.
An introduction to computer networks and the Internet. Applications: http, ftp, e-mail, DNS, socket programming. Transport: UDP, TCP; reliable data transfer, and congestion control. Network layer: IP, routing, and NAT. Link layer: taxonomy, Ethernet, 802.11. Offered as CSDS 325 and CSDS 325N. Prereq: Computer Science Major with minimum Junior Standing and (CSDS 233 or ECSE 233 or ECES 233) with a C or higher.

CSDS 325N. Computer Networks I. 3 Units.
An introduction to computer networks and the Internet. Applications: http, ftp, e-mail, DNS, socket programming. Transport: UDP, TCP; reliable data transfer, and congestion control. Network layer: IP, routing, and NAT. Link layer: taxonomy, Ethernet, 802.11. Offered as CSDS 325 and CSDS 325N. Prereq: Minimum Junior Standing and (CSDS 233 or ECSE 233 or ECES 233) with a C or higher.

CSDS 335D. Data Mining for Big Data. 3 Units.
With the unprecedented rate at which data is being collected today in almost all fields of human endeavor, there is an emerging economic and scientific need to extract useful information from it. Data mining is the process of automatic discovery of patterns, changes, associations and anomalies in massive databases, and is a highly interdisciplinary field representing the confluence of several disciplines, including database systems, data warehousing, machine learning, statistics, algorithms, data visualization, and high-performance computing. This course is an introduction to the commonly used data mining techniques. In the first part of the course, students will develop a basic understanding of the basic concepts in data mining such as frequent pattern mining, association rule mining, basic techniques for data preprocessing such as normalization, regression, and classic matrix decomposition methods such as SVD, LU, and QR decompositions. In the second part of the course, students will develop a basic understanding of classification and clustering and be able to apply classic methods such as k-means, hierarchical clustering methods, nearest neighbor methods, association based classifiers. In the third part of the course, students will have a chance to study more advanced data mining applications such as feature selection in high-dimensional data, dimension reduction, and mining biological datasets. Prereq: CSDS 234 and CSDS 313.

CSDS 337. Compiler Design. 4 Units.
Design and implementation of compilers and other language processors. Scanners and lexical analysis; regular expressions and finite automata; scanner generators; parsers and syntax analysis; context free grammars; parser generators; semantic analysis; intermediate code generation; runtime environments; code generation; machine independent optimizations; data flow and dependence analysis. There will be a significant programming project involving the use of compiler tools and software development tools and techniques. Offered as CSDS 337 and ECSE 337. Prereq: (CSDS 233 or ECSE 233 or ECES 233) and (CSDS 281 or ECSE 281 or ECES 281).
CSDS 338. Intro to Operating Systems and Concurrent Programming. 4 Units.
Intro to OS: OS Structures, processes, threads, CPU scheduling, deadlocks, memory management, file system implementations, virtual machines, cloud computing. Concurrent programming: fork, join, concurrent statement, critical section problem, safety and liveness properties of concurrent programs, process synchronization algorithms, semaphores, monitors. UNIX systems programming: system calls, UNIX System V IPCs, threads, RPCs, shell programming. Offered as CSDS 338, ECSE 338, CSDS 338N and ECSE 338N. Prereq: Computer Science Major or Minor and (CSDS 233 or ECSE 233 or EECS 233) with a C or higher.

CSDS 338N. Intro to Operating Systems and Concurrent Programming. 4 Units.
Intro to OS: OS Structures, processes, threads, CPU scheduling, deadlocks, memory management, file system implementations, virtual machines, cloud computing. Concurrent programming: fork, join, concurrent statement, critical section problem, safety and liveness properties of concurrent programs, process synchronization algorithms, semaphores, monitors. UNIX systems programming: system calls, UNIX System V IPCs, threads, RPCs, shell programming. Offered as CSDS 338, ECSE 338, CSDS 338N and ECSE 338N. Prereq: (CSDS 233 or ECSE 233 or EECS 233) with a C or higher.

CSDS 339. Web Data Mining. 3 Units.
Web crawling technology, web search and information extraction, unsupervised and semi-supervised learning techniques and their application to web data extraction, social network analysis, various pagerank algorithms, link analysis, web resource discovery, web, resource description framework (RDF), XML, Web Ontology Language (OWL). Prereq: (CSDS 338 or ECSE 338) and CSDS 341 and (CSDS 302 or ECSE 302 or MATH 304).

CSDS 340D. Machine Learning for Big Data. 3 Units.
Machine learning is a sub-field of Artificial Intelligence that is concerned with the design and analysis of algorithms that "learn" and improve with experience. While the broad aim behind research in this area is to build systems that can simulate or even improve on certain aspects of human intelligence, algorithms developed in this area have become very useful in analyzing and predicting the behavior of complex systems. Machine learning algorithms have been used to guide diagnostic systems in medicine, recommend interesting products to customers in e-commerce, play games at human championship levels, and solve many other very complex problems. This course is an introduction to algorithms for machine learning and their implementation in the context of big data. We will study different learning settings, the different algorithms that have been developed for these settings, and learn about how to implement these algorithms and evaluate their behavior in practice. We will also discuss dealing with noise, missing values, scalability properties and talk about tools and libraries available for these methods. At the end of the course, you should be able to: --Understand when to use machine learning algorithms; --Understand, represent and formulate the learning problem; --Apply the appropriate algorithm(s) or tools, with an understanding of the tradeoffs involved including scalability and robustness; --Correctly evaluate the behavior of the algorithm when solving the problem. Prereq: CSDS 234 and CSDS 313.

CSDS 341. Introduction to Database Systems. 3 Units.
Relational model, ER model, relational algebra and calculus, SQL, OBE, security, views, files and physical database structures, query processing and query optimization, normalization theory, concurrency control, object relational systems, multimedia databases, Oracle SQL server, Microsoft SQL server. Offered as CSDS 341 and CSDS 341N. Prereq: Computer Science Major or Data Science and Analytics Major and (CSDS 302 or ECSE 302 or MATH 304) and (CSDS233 or ECSE 233) with a C or higher.

CSDS 341D. Introduction to Databases: DS Major. 3 Units.
Database management become a central component of a modern computing environment, and, as a result, knowledge about database systems has become an essential part of education in computer science and data science. This course is an introduction to the nature and purpose of database systems, fundamental concepts for designing, implementing and querying a database and database architectures. Weeks 1-6 provide an overview of basic database systems concepts including database design, database systems architecture, and database querying, using relational model and SQL as query language. Weeks 7-10 Objects, Semi structured data, XML and RDF basics. Weeks 11-14 provide an overview of more advanced topics including Database System Architectures (Parallel Databases and Distributed Databases), and Data Warehousing and Information Retrieval. Students cannot receive credit for both CSDS341 and CSDS341D. Prereq: CSDS 233 or ECSE 233 or CSDS 234.

CSDS 341N. Introduction to Databases: DS Major. 3 Units.
Relational model, ER model, relational algebra and calculus, SQL, OBE, security, views, files and physical database structures, query processing and query optimization, normalization theory, concurrency control, object relational systems, multimedia databases, Oracle SQL server. Offered as CSDS 341 and CSDS 341N. Prereq: (CSDS 302 or ECSE 302 or MATH 304) with a grade of C or higher and (CSDS 233 or ECSE 233) with C or higher.

CSDS 343. Theoretical Computer Science. 3 Units.
Introduction to different classes of automata and their correspondence to different classes of formal languages and grammars, computability, complexity and various proof techniques. Offered as CSDS 343 and MATH 343. Prereq: EECS 302 or MATH 304.

CSDS 344. Computer Security. 3 Units.
General types of security attacks; approaches to prevention; secret key and public key cryptography; message authentication and hash functions; digital signatures and authentication protocols; information gathering; password cracking; spoofing; session hijacking; denial of service attacks; buffer overruns; viruses, worms, etc., principles of secure software design, threat modeling; access control; least privilege; storing secrets; socket security; firewalls; intrusions; auditing; mobile security. Recommended preparation: EECS/CSDS/ECSE 132, EECS/CSDS 293, EECS/CSDS 325 and EECS/CSDS 338. Offered as CSDS 344 and CSDS 444.
CSDS 345. Programming Language Concepts. 3 Units.
This course examines the four main programming paradigms: imperative, object-oriented, functional, and logical. It is assumed that students will come to the course with significant exposure to object-oriented programming and some exposure to imperative programming. The course will teach the functional paradigm in depth, enhance the students’ knowledge of the object-oriented and imperative paradigms, and introduce the logical paradigm. The course will explore language syntax, semantics, names/scopes, types, expressions, assignment, subprograms, abstraction and inheritance. This exploration will have several forms. Students will study the programming language concepts at a theoretical level, use the concepts in functional language programming, and implement the concepts by designing language interpreters. Offered as CSDS 345 and CSDS 345N. Prereq: Computer Science Major and (CSDS 302 or ECSE 302 or MATH 304) with a C or higher and (CSDS 233 or ECSE 233) with a C or higher.

CSDS 345N. Programming Language Concepts. 3 Units.
This course examines the four main programming paradigms: imperative, object-oriented, functional, and logical. It is assumed that students will come to the course with significant exposure to object-oriented programming and some exposure to imperative programming. The course will teach the functional paradigm in depth, enhance the students’ knowledge of the object-oriented and imperative paradigms, and introduce the logical paradigm. The course will explore language syntax, semantics, names/scopes, types, expressions, assignment, subprograms, abstraction and inheritance. This exploration will have several forms. Students will study the programming language concepts at a theoretical level, use the concepts in functional language programming, and implement the concepts by designing language interpreters. Offered as CSDS 345 and CSDS 345N. Prereq: (CSDS 302 or ECSE 302 or MATH 304) with a C or higher and (CSDS 233 or ECSE 233) with a C or higher.

CSDS 356. Data Privacy. 3 Units.
Introduction to privacy, economics and incentives, crypto-based solution for privacy, hiding data from the database user, hiding access patterns from the database owner; anonymous routing and TOR, privacy in online social networks, privacy in cellular and Wi-Fi networks, location privacy, privacy in e-cash systems, privacy in e-voting, genomic privacy. Offered as CSDS 356 and CSDS 356N. Prereq: (EECS 132 and ECES 233 and any STAT course) or (Graduate student in Computing & Info Science or Computer Engineering).

CSDS 364. Computational Perception. 3 Units.
An introduction to the information processing and computational algorithms that underlie perception. The course focuses on vision and audition but also covers other senses and various types of perceptual processing in biological systems. The motivating question is: What are the fundamental computational problems faced by perceptual systems in complex environments? The course draws from fields in engineering and the study of biological perception. Specific topics include signal detection and processing; sound localization; motion perception and estimation; sensory coding; perception of structure; active perception; perceptual invariance; attention; object, sound and speech recognition; visual and auditory scene analysis. Offered as CSDS 364 and CSDS 464. Prereq: CSDS 132 or ENGR 131.

CSDS 366. Computer Graphics. 3 Units.
Theory and practice of computer graphics: object and environment representation including coordinate transformations image extraction including perspective, hidden surface, and shading algorithms; and interaction. Covers a wide range of graphic display devices and systems with emphasis in interactive shaded graphics. Offered as CSDS 366, ECSE 366, CSDS 466 and ECSE 466. Prereq: EECS 233.

CSDS 373. Modern Robot Programming. 3 Units.
The goal of this course is to learn modern methods for building up robot capabilities using the Robot Operating System (ROS). Through a sequence of assignments, students learn how to write software to control both simulated and physical robots. Material includes: interfacing software to robot I/O; path and trajectory planning for robot arms; object identification and localization from 3-D sensing; manipulation planning; and development of graphical interfaces for supervisory robot control. Laboratory assignments are scheduled in small groups to explore implementations on specific robots. Graduate students will also perform an independent project. Offered as CSDS 373, ECSE 373, CSDS 473 and ECSE 473.

CSDS 376. Mobile Robotics. 4 Units.
Design of software systems for mobile robot control, including motion control; sensory processing; localization and mapping; mobile-robot planning and navigation; and implementation of goal-directed behaviors. The course has a heavy lab component involving a sequence of design challenges and competitions performed in teams. Offered as CSDS 376 and ECSE 376. Prereq: CSDS 373 or CSDS 473.

CSDS 377. Introduction to Connected Devices. 3 Units.
Introduction to Connected Devices (e.g., Internet of Things). Undergraduates work in pairs to build a complete connected-device system, an embedded device with wireless networking, cloud and web, and mobile, and then develop hands-on experience with systems-level aspects of the connected-device system, including analytics, remote firmware update, load testing, and essential security. Students learn about current architectures, languages, and technologies, such as Pub/Sub (MQTT), Python, Objective-C, Python Django, JavaScript, HTML/CSS, and Bluetooth Low Energy. Offered as CSDS 377 and ECSE 377.

CSDS 386. Quantum Computing, Information, and Devices. 3 Units.
An introduction to the math, physics, engineering, and computer science underlying the rapidly emerging fields of quantum computing, quantum information, and quantum devices. The course is taught by a group of faculty from physics, engineering, computer science, and math, and is geared towards students with diverse backgrounds and interests in these fields. Students will select a concentration in one of these four areas, and the coursework, while still covering all topics, will be adjusted to focus on the selected area in the most detail. Note that the listed prerequisites depend on choice of concentration. Topics will include: 1. (Mathematics) Introduction to linear algebra, convex geometry, fundamental theory of quantum information. 2. (Physics) Introduction to the quantum mechanics of two-level systems (qubits). Survey of physics and materials for qubit technologies. 3. (Computer Science) Basic quantum gates and circuits, introduction to the theory of algorithms, survey of quantum algorithms. 4. (Engineering) Quantum architectures, mapping algorithms onto circuits. The course consists of lectures, homework, and group projects. Group projects will aim to synthesize the diverse backgrounds of the students and instructors to capture the interdisciplinary nature of the field. Students taking the course for graduate credit will complete an additional literature research project and presentation, in addition to enhanced problem sets. Offered as CSDS 386, CSDS 486, ECSE 386, ECSE 486, MATH 386, MATH 486, PHYS 386, and PHYS 486. Prereq: (CSDS 233 or ECSE 233) and (CSDS 281 or ECSE 281) and (MATH 201 or MATH 307) and (MATH 223 or MATH 227) and (PHYS 122 or PHYS 124).
CSDS 390. Advanced Game Development Project. 3 Units.
This game development project course will bring together an inter-
professional group of students in the fields of engineering, computer
science, and art to focus on the design and development of a complete,
fully functioning computer game as an interdisciplinary team. The
student teams are given complete liberty to design their own fully
functional games from their original concept to a playable game
published in an online marketplace. Student teams will experience
the entire game development cycle as they execute their projects.
Responsibilities include creating a game idea, writing a story, developing
the artwork, designing characters, implementing music and sound
effects, programming and testing the game, and publishing the final
project. Students enrolled in 487 will develop a healthcare or education
virtual environment or video game in collaboration with a mentor who has
expertise in the chosen area. Offered as CSDS 390, ECSE 390, CSDS 487,
and ECSE 487. Prereq: EECS 233 and ECES 290.

CSDS 391. Introduction to Artificial Intelligence. 3 Units.
This course is an introduction to artificial intelligence. We will study
the concepts that underlie intelligent systems. Topics covered include
problem solving with search, constraint satisfaction, adversarial games,
knowledge representation and reasoning using propositional and
first order logic, reasoning under uncertainty, introduction to machine
learning, automated planning, reinforcement learning and natural
language processing. Recommended: basic knowledge of probability
and statistics. Prereq: ENGR 131 or CSDS 132 or ECSE 132.

CSDS 392. App Development for iOS. 3 Units.
This course is an introduction to app development for iPhone and iPad
using Cocoa Touch Framework and Xcode development environment.
Topics include Swift programming language and iOS SDK/FOUNDATIONS,
object-oriented design and model-view-controller framework, user
interface design using Xcode. Additional topics may include data
management, map applications, animations and some recent
developments in iOS. Recommended preparation: experiences in object-
oriented programming and Mac OS; knowledge in software engineering
and databases. Prereq: ECES 293 and Junior or Senior standing.

CSDS 393. Software Engineering. 3 Units.
Topics: Introduction to software engineering; software lifecycle models;
development team organization and project management; requirements
analysis and specification techniques; software design techniques;
programming practices; software validation techniques; software
maintenance practices; software engineering ethics. Undergraduates
work in teams to complete a significant software development project.
Graduate students are required to complete a research project. Offered as
CSDS 393, CSDS 393N, and CSDS 493. Prereq: (Computer Science Major/
Minor or Data Science Major) and (CSDS 233 or ECSE 233) with a C or
higher.

CSDS 393N. Software Engineering. 3 Units.
Topics: Introduction to software engineering; software lifecycle models;
development team organization and project management; requirements
analysis and specification techniques; software design techniques;
programming practices; software validation techniques; software
maintenance practices; software engineering ethics. Undergraduates
work in teams to complete a significant software development project.
Graduate students are required to complete a research project. Offered as
CSDS 393, CSDS 393N, and CSDS 493. Prereq: CSDS 233 or ECSE 233
with a C or higher.

CSDS 394. Introduction to Information Theory. 3 Units.
This course is intended as an introduction to information and coding
theory with emphasis on the mathematical aspects. It is suitable for
advanced undergraduate and graduate students in mathematics, applied
mathematics, statistics, physics, computer science and electrical
engineering. Course content: Information measures-entropy, relative
entropy, mutual information, and their properties. Typical sets and
sequences, asymptotic equipartition property, data compression.
Channel coding and capacity channel coding theorem. Differential
entropy, Gaussian channel, Shannon-Nyquist theorem. Information theory
inequalities (400 level). Additional topics, which may include compressed
sensing and elements of quantum information theory. Recommended
preparation: MATH 201 or MATH 307. Offered as CSDS 394, CSDS 394,
ECSE 394, MATH 494, CSDS 494 and ECSE 494. Prereq: MATH 223 and
MATH 380 or requisites not met permission.

CSDS 395. Senior Project in Computer Science. 4 Units.
Capstone course for computer science seniors. Material from previous
and concurrent courses used to solve computer programming problems
and to develop software systems. Professional engineering topics such
as project management, engineering design, communications, and
professional ethics. Requirements include periodic reporting of progress,
plus a final oral presentation and written report. Scheduled formal project
presentations during last week of classes. Counts as SAGES Senior
Capstone. Prereq: Senior standing.

CSDS 396. Independent Projects. 1 - 6 Units.
Independent projects in Computer and Data Science. Prereq: Limited to
juniors and seniors.

CSDS 397. Special Topics. 1 - 6 Units.
Special topics in Computer and Data Science. Prereq: Limited to juniors
and seniors.

CSDS 400T. Graduate Teaching I. 0 Unit.
This course will provide the Ph.D. candidate with experience in teaching
undergraduate or graduate students. The experience is expected to
involve direct student contact but will be based upon the specific
departmental needs and teaching obligations. This teaching experience
will be conducted under the supervision of the faculty member who is
responsible for the course, but the academic advisor will assess the
educational plan to ensure that it provides an educational experience for
the student. Students in this course may be expected to perform one or
more of the following teaching related activities: grading homeworks,
quizzes, and exams, having office hours for students, tutoring students.
Recommended preparation: Ph.D. student in CDS department.

CSDS 405. Data Structures and File Management. 3 Units.
Fundamental concepts: sequential allocation, linked allocation, lists,
trees, graphs, internal sorting, external sorting, sequential, binary,
interpolation search, hashing file, indexed files, multiple level index
structures, btrees, hashed files. Multiple attribute retrieval; inverted files,
multi lists, multiple-key hashing, hd trees. Introduction to data bases.
Data models. Recommended preparation: EECS/CDS/ECSE 233 and
MATH 304. Prereq: ECES 341 or Graduate standing.

CSDS 410. Analysis of Algorithms. 3 Units.
This course covers fundamental topics in algorithm design and analysis
in depth. Amortized analysis, NP-completeness and reductions, dynamic
programming, advanced graph algorithms, string algorithms, geometric
algorithms, local search heuristics. Offered as CSDS 410 and OPRW 454.
Prereq: EECS 340.
CSDS 425. Computer Networks I. 3 Units.

CSDS 427. Internet Security and Privacy. 3 Units.
This course introduces students to research on Internet security and privacy. Covered topics include denial of service attacks, attacks enabled by man-in-the-middle surveillance, communication hijacking, botnet and fast-flux networks, email and Web spam, threats to privacy on the Internet, and Internet censorship. The course will be based on a collection of research papers. Students will be required to attend lectures, read the materials, prepare written summaries of discussed papers, present a paper in class, complete a course project and take the final exam (in the form of the course project presentation). Prereq: EECS 325 or EECS 425 or graduate standing in Computer Science or Computer Engineering.

CSDS 428. Computer Communications Networks II. 3 Units.
Introduction to topics and methodology in computer networks and middleware research. Traffic characterization, stochastic models, and self-similarity. Congestion control ( Tahoe, Reno, Sack). Active Queue Management ( RED, FQ) and explicit QoS. The Web: overview and components, HTTP, its interaction with TCP, caching. Overlay networks and CDN. Expected work includes a course-long project on network simulation, a final project, a paper presentation, midterm, and final test. Recommended preparation: EECS/CSDS 425.

CSDS 433. Database Systems. 3 Units.

CSDS 435. Data Mining. 3 Units.
Data Mining is the process of discovering interesting knowledge from large amounts of data stored either in databases, data warehouses, or other information repositories. Topics to be covered includes: Data Warehouse and OLAP technology for data mining, Data Preprocessing, Data Mining Primitives, Languages, and System Architectures, Mining Association Rules from Large Databases, Classification and Prediction, Cluster Analysis, Mining Complex Types of Data, and Applications and Trends in Data Mining. Prereq: EECS 341 or Graduate standing.

CSDS 436. High Performance Computing. 3 Units.
High performance computing (HPC) leverages parallel processing in order to maximize speed and throughput. This hands-on course will cover theoretical and practical aspects of HPC. Theoretical concepts covered include computer architecture, parallel programming, and performance optimization. Practical applications will be discussed from various information and scientific fields. Practical considerations will include HPC job management and Unix scripting. Weekly assignments and a course project will be required. Offered as CSDS 438 and ECSE 438. Prereq: EECS 233 or graduate standing.

CSDS 439. Web Data Mining. 3 Units.
Web crawling technology, web search and information extraction, unsupervised and semi-supervised learning techniques and their application to web data extraction, social network analysis, various pagerank algorithms, link analysis, web resource discovery, web resource description framework (RDF), XML, Web Ontology Language (OWL). Recommended preparation: EECS/CSDS/ECSE 338, EECS/CSDS 341.

CSDS 440. Machine Learning. 3 Units.
Machine learning is a subfield of Artificial Intelligence that is concerned with the design and analysis of algorithms that "learn" and improve with experience. While the broad aim behind research in this area is to build systems that can simulate or even improve on certain aspects of human intelligence, algorithms developed in this area have become very useful in analyzing and predicting the behavior of complex systems. Machine learning algorithms have been used to guide diagnostic systems in medicine, recommend interesting products to customers in e-commerce, play games at human championship levels, and solve many other very complex problems. This course is focused on algorithms for machine learning: their design, analysis and implementation. We will study different learning settings, including supervised, semi-supervised and unsupervised learning. We will study different ways of representing the learning problem, using propositional, multiple-instance and relational representations. We will study the different algorithms that have been developed for these settings, such as decision trees, neural networks, support vector machines, k-means, harmonic functions and Bayesian methods. We will learn about the theoretical tradeoffs in the design of these algorithms, and how to evaluate their behavior in practice. At the end of the course, you should be able to: --Recognize situations where machine learning algorithms are applicable; --Understand, represent and formulate the learning problem; --Apply the appropriate algorithm(s), or if necessary, design your own, with an understanding of the tradeoffs involved; --Correctly evaluate the behavior of the algorithm when solving the problem. Prereq: CSDS 391 or CSDS 491.

CSDS 442. Causal Learning from Data. 3 Units.
This course introduces key concepts and techniques for characterizing, from observational or experimental study data and from background information, the causal effect of a specific treatment, exposure, or intervention (e.g., a medical treatment) upon an outcome of interest (e.g., disease status). The fundamental problem of causal inference is the impossibility of observing the effects of different and incompatible treatments on the same individual or unit. This problem is overcome by estimating an average causal effect over a study population. Making valid causal inferences with observational data is especially challenging, because of the greater potential for biases (confounding bias, selection bias, and measurement bias) that can badly distort causal effect estimates. Consequently, this topic has been the focus of intense cross-disciplinary research in recent years. Causal inference techniques will be illustrated by applications in several fields such as computer science, engineering, medicine, public health, biology, genomics, neuroscience, economics, and social science. Course grading will be based on quizzes, homeworks, a class presentation, and a causal data analysis project. Specific topics: treatments, exposures, and interventions; causal effects and causal effect measures; confounding bias; potential outcomes and counterfactuals; randomized experiments; observational studies; causal directed acyclic graphs (DAGs); exchangeability and conditional exchangeability; effect modification; causal interactions; nonparametric structural equations; Pearl's Back-Door Criterion, Front-Door Criterion, and related results; covariate adjustment; matching on covariates; selection bias; measurement bias; instrumental variables; causal modeling; inverse probability weighting; marginal structural models; standardization; structural nested models; outcome regression; propensity scores; sensitivity analysis. Prereq: EECS 440 or MATH 380 or STAT 312 or STAT 313 or STAT 332 or STAT 333 or Requisites Not Met permission.
CSDS 444. Computer Security. 3 Units.
General types of security attacks; approaches to prevention; secret key and public key cryptograph; message authentication and hash functions; digital signatures and authentication protocols; information gathering; password cracking; spoofing; session hijacking; denial of service attacks; buffer overruns; viruses, worms, etc., principles of secure software design, threat modeling; access control; least privilege; storing secrets; socket security; firewalls; intrusions; auditing; mobile security. Recommended preparation: EECS/CSDS/ECSE 132, EECS/CSDS 293, ECSE/CSDS 325 and EECS/CSDS 338. Offered as CSDS 344 and CSDS 444.

CSDS 448. Smartphone Security. 3 Units.
This course is designed to better prepare undergraduate and graduate students for researching and developing in the neighborhood of mobile and software security. Lectures, paper readings and presentations, in-class discussions, and projects are the main components. The course covers the basics of Android programming and a wide range of security issues and solutions concerning mobile platforms, including permission analysis, textual artifacts analysis, malware analysis, program analysis, and UI analysis. Students should expect one literature survey paper and one system-building or empirical study project on one selected security solution in mobile app security. Prereq: EECS 132 and a Graduate or Undergraduate Computer Science major.

CSDS 455. Applied Graph Theory. 3 Units.
This course serves as an introduction to many of the important aspects of graph theory. Topics include connectivity, flows, matchings, planar graphs, and graph coloring with additional topics selected from extremal graphs, random graphs, bounded treewidth graphs, social networks and small world graphs. The class will explore the underlying mathematical theory with a specific focus on the development and analysis of graph algorithms. Prereq: Graduate Student or (EECS 302 or MATH 304 or MATH 305 or MATH 307).

CSDS 456. Data Privacy. 3 Units.
Introduction to privacy, economics and incentives, crypto-based solution for privacy, hiding data from the database user, hiding access patterns from the database owner, anonymous routing and TOR, privacy in online social networks, privacy in cellular and Wi-Fi networks, location privacy, privacy in e-cash systems, privacy in e-voting, genomic privacy. Offered as CSDS 356 and CSDS 456. Prereq: (EECS 132 and EECS 233 and any STAT course) or (Graduate student in Computing & Info Science or Computer Engineering).

CSDS 458. Introduction to Bioinformatics. 3 Units.
Fundamental algorithmic and statistical methods in computational molecular biology and bioinformatics will be discussed. Topics include introduction to molecular biology and genetics, DNA sequence analysis, polymorphisms and personal genomics, structural variation analysis, gene mapping and haplotyping algorithms, phylogenetic analysis, biological network analysis, and computational drug discovery. Much of the course will focus on the algorithmic techniques, including but not limited to, dynamic programming, hidden Markov models, string algorithms, graph theories and algorithms, and some representative data mining algorithms. Paper presentations and course projects are also required. Prereq: EECS 340 or Graduate standing.

CSDS 459. Bioinformatics for Systems Biology. 3 Units.

CSDS 464. Computational Perception. 3 Units.
An introduction to the information processing and computational algorithms that underlie perception. The course focuses on vision and audition but also covers other senses and various types of perceptual processing in biological systems. The motivating question is: What are the fundamental computational problems faced by perceptual systems in complex environments? The course draws from fields in engineering and the study of biological perception. Specific topics include signal detection and processing; sound localization; motion perception and estimation; sensory coding; perception of structure; active perception; perceptual invariance; attention; object, sound and speech recognition; visual and auditory scene analysis. Offered as CSDS 364 and CSDS 464. Prereq: CSDS 132 or ENGR 131.

CSDS 465. Computer Vision. 3 Units.
The goal of computer vision is to create visual systems that recognize objects and recover structures in complex 3D scenes. This course emphasizes both the science behind our understanding of the fundamental problems in vision and the engineering that develops mathematical models and inference algorithms to solve these problems. Specific topics include feature detection, matching, and classification; visual representations and dimensionality reduction; motion detection and optical flow; image segmentation; depth perception, multi-view geometry, and 3D reconstruction; shape and surface perception; visual scene analysis and object recognition. Offered as CSDS 465 and ECSE 465.

CSDS 466. Computer Graphics. 3 Units.
Theory and practice of computer graphics: object and environment representation including coordinate transformations image extraction including perspective, hidden surface, and shading algorithms; and interaction. Covers a wide range of graphic display devices and systems with emphasis in interactive shaded graphics. Offered as CSDS 366, ECSE 366, CSDS 466 and ECSE 466. Prereq: Graduate standing or Requisites Not Met permission.

CSDS 473. Modern Robot Programming. 3 Units.
The goal of this course is to learn modern methods for building up robot capabilities using the Robot Operating System (ROS). Through a sequence of assignments, students learn how to write software to control both simulated and physical robots. Material includes: interfacing software to robot I/O; path and trajectory planning for robot arms; object identification and localization from 3-D sensing; manipulation planning; and development of graphical interfaces for supervisory robot control. Laboratory assignments are scheduled in small groups to explore implementations on specific robots. Graduate students will also perform an independent project. Offered as CSDS 373, ECSE 373, CSDS 473 and ECSE 473.
CSDS 476. Mobile Robotics. 3 Units.
Design of software systems for mobile robot control, including: motion control; sensory processing; localization and mapping; mobile-robot planning and navigation; and implementation of goal-directed behaviors. The course has a heavy lab component involving a sequence of design challenges and competitions performed in teams. Offered as CSDS 476 and ECSE 476. Prereq: EECS 373 or ECES 473.

CSDS 477. Advanced Algorithms. 3 Units.

CSDS 478. Computational Neuroscience. 3 Units.
Computer simulations and mathematical analysis of neurons and neural circuits, and the computational properties of nervous systems. Students are taught a range of models for neurons and neural circuits, and are asked to implement and explore the computational and dynamic properties of these models. The course introduces students to dynamical systems theory for the analysis of neurons and neural learning, models of brain systems, and their relationship to artificial and neural networks. Term project required. Students enrolled in MATH 478 will make arrangements with the instructor to attend additional lectures and complete additional assignments addressing mathematical topics related to the course. Recommended preparation: MATH 223 and MATH 224 or BIOL 300 and BIOL 306. Offered as BIOL 378, COGS 378, MATH 378, BIOL 478, CSDS 478, EBME 478, ECSE 478, MATH 478 and NEUR 478.

CSDS 486. Quantum Computing, Information, and Devices. 3 Units.
An introduction to the math, physics, engineering, and computer science underlying the rapidly emerging fields of quantum computing, quantum information, and quantum devices. The course is taught by a group of faculty from physics, engineering, computer science, and math, and is geared towards students with diverse backgrounds and interests in these fields. Students will select a concentration in one of these four areas, and the coursework, while still covering all topics, will be adjusted to focus on the selected area in the most detail. Note that the listed prerequisites depend on choice of concentration. Topics will include: 1. (Mathematics) Introduction to linear algebra, convex geometry, fundamental theory of quantum information. 2. (Physics) Introduction to the quantum mechanics of two-level systems (qubits). Survey of physics and materials for qubit technologies. 3. (Computer Science) Basic quantum gates and circuits, introduction to the theory of algorithms, survey of quantum algorithms. 4. (Engineering) Quantum architectures, mapping algorithms onto circuits. The course consists of lectures, homework, and group projects. Group projects will aim to synthesize the diverse backgrounds of the students and instructors to capture the interdisciplinary nature of the field. Students taking the course for graduate credit will complete an additional literature research project and presentation, in addition to enhanced problem sets. Offered as CSDS 386, CSDS 486, ECSE 386, ECSE 486, MATH 386, MATH 486, PHYS 386, and PHYS 486. Prereq: (CSDS 233 or ECSE 233) and (CSDS 281 or ECSE 281) and (MATH 201 or MATH 307) and (MATH 223 or MATH 227) and (PHYS 122 or PHYS 124).

CSDS 487. Advanced Game Development Project. 3 Units.
This game development project course will bring together an interdisciplinary group of students in the fields of engineering, computer science, and art to focus on the design and development of a complete, fully functioning computer game as an interdisciplinary team. The student teams are given complete liberty to design their own fully functional games from their original concept to a playable game published in an online marketplace. Student teams will experience the entire game development cycle as they execute their projects. Responsibilities include creating a game idea, writing a story, developing the artwork, designing characters, implementing music and sound effects, programming and testing the game, and publishing the final project. Students enrolled in 487 will develop a healthcare or education virtual environment or video game in collaboration with a mentor who has expertise in the chosen area. Offered as CSDS 390, ECSE 390, CSDS 487, and ECSE 487. Prereq: Graduate student standing.

CSDS 489. Robotics I. 3 Units.

CSDS 490. Digital Image Processing. 3 Units.
Digital images are introduced as two-dimensional sampled arrays of data. The course begins with one-to-one operations such as image addition and subtraction and image descriptors such as the histogram. Basic filters such as the gradient and Laplacian in the spatial domain are used to enhance images. The 2-D Fourier transform is introduced and frequency domain operations such as high and low-pass filtering are developed. It is shown how filtering techniques can be used to remove noise and other image degradation. The different methods of representing color images are described and fundamental concepts of color image transformations and color image processing are developed. One or more advanced topics such as wavelets, image compression, and pattern recognition will be covered as time permits. Programming assignments using software such as MATLAB will illustrate the application and implementation of digital image processing. Offered as CSDS 490 and ECSE 490.

CSDS 491. Artificial Intelligence: Probabilistic Graphical Models. 3 Units.
This course is a graduate-level introduction to Artificial Intelligence (AI), the discipline of designing intelligent systems, and focuses on probabilistic graphical models. These models can be applied to a wide variety of settings from data analysis to machine learning to robotics. The models allow intelligent systems to represent uncertainties in an environment or problem space in a compact way and reason intelligently in a way that makes optimal use of available information and time. The course covers directed and undirected probabilistic graphical models, latent variable models, associated exact and approximate inference algorithms, and learning in both discrete and continuous problem spaces. Practical applications are covered throughout the course. Prereq: CSDS 391 or Requisites Not Met permission.
CSDS 493. Software Engineering. 3 Units.
Topics: Introduction to software engineering; software lifecycle models; development team organization and project management; requirements analysis and specification techniques; software design techniques; programming practices; software validation techniques; software maintenance practices; software engineering ethics. Undergraduates work in teams to complete a significant software development project. Graduate students are required to complete a research project. Offered as CSDS 393, CSDS 393N, and CSDS 493. Prereq: (Computer Science Major/Minor or Data Science Major) and (CSDS 233 or ECSE 233) with a C or higher.

CSDS 494. Introduction to Information Theory. 3 Units.
This course is intended as an introduction to information and coding theory with emphasis on the mathematical aspects. It is suitable for advanced undergraduate and graduate students in mathematics, applied mathematics, statistics, physics, computer science and electrical engineering. Course content: Information measures-entropy, relative entropy, mutual information, and their properties. Typical sets and sequences, asymptotic equipartition property, data compression. Channel coding and capacity: channel coding theorem. Differential entropy, Gaussian channel, Shannon-Nyquist theorem. Information theory inequalities (400 level). Additional topics, which may include compressed sensing and elements of quantum information theory. Recommended preparation: MATH 201 or MATH 307. Offered as MATH 394, CSDS 394, ECSE 394, MATH 494, CSDS 494 and ECSE 494.

CSDS 496. Artificial Intelligence: Sequential Decision Making. 3 Units.
This course will study the formulation and solution of decision making problems by automated agents. Topics covered include one-shot decision making (decision trees and influence diagrams), Markov decision processes (MDPs), automated classical and probabilistic planning, reinforcement learning (RL), hierarchical planning and RL, partially observable MDPs, Bayesian RL, collaborative multi-agent systems. Recommended preparation: EECS/CSDS 491 (Probabilistic Graphical Models). Prereq: EECS 391.

CSDS 497. Artificial Intelligence: Statistical Natural Language Processing. 3 Units.
This course gives students an overview of the state of the art in natural language processing. We will discuss computational aspects of language modeling through probabilistic models, computational approaches to syntax (parsing) and semantic representations, discourse and dialog. We will study the applications of these techniques to a variety of problems including information extraction, translation and summarization. At the end of the course a student should be able to (i) understand the various statistical models and algorithms for NLP (ii) modify them as needed or design novel approaches for specific NLP tasks and (iii) understand how to evaluate the performance of these models and compare them to alternatives. Prereq: EECS 440.

CSDS 499. Algorithmic Robotics. 3 Units.
This course introduces basic algorithmic techniques in robotic perception and planning. Course is divided into two parts. The first part introduces probabilistic modeling of robotic motion and sensing, Gaussian and nonparametric filters, and algorithms for mobile robot localization. The second part introduces fundamental deterministic and randomized algorithms for motion planning. Offered as CSDS 499 and ECSE 499. Prereq: Graduate Standing or Requisites Not Met permission.

CSDS 500. CSDS Colloquium. 0 Unit.
Seminars on current topics in Computer and Data Science.