## Computer Science Concentration Requirements

Computer science is now a critical tool for pursuing an ever-broadening range of topics, from outer space to the workings of the human mind. In most areas of science and in many liberal arts fields, cutting-edge work depends increasingly on computational approaches. The undergraduate program at Brown is designed to combine breadth in practical and theoretical computer science with depth in specialized areas. These areas range from traditional topics, such as analysis of algorithms, artificial intelligence, databases, distributed systems, graphics, mobile computing, networks, operating systems, programming languages, robotics and security, to novel areas including games and scientific visualization.

Our requirements are built on a collection of pathways, each representing a well-defined area within computer science. Concentrators interested in particular areas can choose the courses included in particular pathways. Conversely, concentrators who are unsure of their area of interest but who have particularly enjoyed certain courses can choose pathways that include these concentrations. Students may not use more than two CSCI 1970 courses to complete the requirements for the Sc.B. and one CSCI 1970 course for the A.B. requirements.

For up-to-date information on our concentration requirements please see https://cs.brown.edu/degrees/undergrad/concentrating-in-cs/concentration-requirements-2020/new-scb-requirements/ for ScB requirements and https://cs.brown.edu/degrees/undergrad/concentrating-in-cs/concentration-requirements-2020/new-ab-requirements/ for AB requirements. Please see https://cs.brown.edu/degrees/undergrad/concentrating-in-cs/concentration-handbook/ for further discussion regarding our concentration requirements.

### Core-Computer Science

#### Series A
- CSCI 0150 & CSCI 0200: Introduction to Object-Oriented Programming and Computer Science and Program Design with Data Structures and Algorithms

#### Series B
- CSCI 0170 & CSCI 0200: Computer Science: An Integrated Introduction and Program Design with Data Structures and Algorithms

#### Series C
- CSCI 0190: Accelerated Introduction to Computer Science (and an additional CS course not otherwise used to satisfy a concentration requirement; this course may be CSCI 0200, an intermediate-level course, or an advanced course)

#### Series D

### Thirteen more advanced courses.
- **# Two complete pathways (at least one core course from each)**
- **# Each requires two 1000-level courses as well as one-to-three intermediate courses**
- **# One of the courses used in one pathway must be a capstone course (defined below)**
- **# The core and related courses used in one pathway may not overlap with those used in another**
- **# 2000-level courses beyond those explicitly mentioned may also be used toward the concentration. They will be considered to be part of the same pathway as their thematically-related 1000-level courses**
- **# Additional intermediate courses so that a total of five are taken, with at least one from each of the three categories**
- **# One additional 1000-level course that is neither a core nor a related nor a graduate course for the pathways used above**
- **# No more than four arts, humanities, and social science oriented CS courses (currently CSCI 1250, 1280, 1370, 1800, 1805, and 1870) may be used for concentration credit.**

### Intermediate Courses

ScB students must take at least one course from each intermediate course category to ensure they span all areas. In addition, they must take whatever intermediate courses they haven’t yet taken that are required for their pathways.

#### Foundations
- CSCI 0220: Introduction to Discrete Structures and Probability
- or MATH 1001: The Art of Writing Mathematics
or MATH 1530  Abstract Algebra  
CSCI 1010  Theory of Computation  

Mathematics  

CSCI 0530  Coding the Matrix: An Introduction to Linear Algebra for Computer Science  
or MATH 0520  Linear Algebra  
or MATH 0540  Linear Algebra With Theory  

CSCI 1450  Advanced Introduction to Probability for Computing and Data Science  
or APMA 1650  Statistical Inference I  
or APMA 1655  Honors Statistical Inference I  

MATH 0160  Multivariable Calculus  
or MATH 0200  Multivariable Calculus (Physics/Engineering)  
or MATH 0350  Multivariable Calculus With Theory  

Systems  

CSCI 0320  Introduction to Software Engineering 
CSCI 0330  Introduction to Computer Systems  
or CSCI 0300  Fundamentals of Computer Systems  

Pathways  
Completing a pathway entails taking two courses in the pathway of which at least one is a core course for the pathway. One must also take the intermediate courses specified as part of the pathway. Certain graduate courses can also satisfy pathway requirements. See the CS Pathway page for more info: http://cs.brown.edu/degrees/undergrad/concentrating-in-cs/concentration-handbook/  

SYSTEMS: studies the design, construction, and analysis of modern, multi-faceted computing systems  

Core Courses  
CSCI 1380  Distributed Computer Systems  
or CSCI 1670  Operating Systems  
or CSCI 1680  Computer Networks  

Related Courses  
CSCI 1260  Compilers and Program Analysis  
CSCI 1270  Database Management Systems  
or CSCI 1320  Creating Modern Web & Mobile Applications  
or CSCI 1600  Real-Time and Embedded Software  
or CSCI 1650  Software Security and Exploitation  
or CSCI 1660  Introduction to Computer Systems Security  
or CSCI 1710  Logic for Systems  
or CSCI 1730  Design and Implementation of Programming Languages  
or CSCI 1760  Multiprocessor Synchronization  
or ENGN 1640  Design of Computing Systems  

Intermediate Courses  
CSCI 0300  Fundamentals of Computer Systems  
or CSCI 0330  Introduction to Computer Systems  
CSCI 0220  Introduction to Discrete Structures and Probability  

SOFTWARE PRINCIPLES: studies the design, construction, and analysis of modern software systems  

Core Courses  
CSCI 1260  Compilers and Program Analysis  
or CSCI 1320  Creating Modern Web & Mobile Applications  
or CSCI 1600  Real-Time and Embedded Software  
or CSCI 1710  Logic for Systems  
or CSCI 1730  Design and Implementation of Programming Languages  

Related Courses  
CSCI 1270  Database Management Systems  
or CSCI 1380  Distributed Computer Systems  
or CSCI 1650  Software Security and Exploitation  
or CSCI 1680  Computer Networks  
or CSCI 1951I  CS for Social Change  
or CSCI 1951T  Surveying VR Data Visualization Software for Research  
or CSCI 1951X  Formal Proof and Verification  

Intermediate Courses  
CSCI 0220  Introduction to Discrete Structures and Probability  
CSCI 0320  Introduction to Software Engineering  
CSCI 0330  Introduction to Computer Systems (Data)  
or CSCI 0300  Fundamentals of Computer Systems  

DATA: studies the management and use of large data collections  

Core Courses  
CSCI 1270  Database Management Systems  
or CSCI 1420  Machine Learning  
or CSCI 1951A  Data Science  

Related Courses  
CSCI 1550  Probabilistic Methods in Computer Science  

Intermediate Courses  
CSCI 0320  Introduction to Software Engineering  
or CSCI 0330  Introduction to Computer Systems  
or CSCI 0300  Fundamentals of Computer Systems  
MATH 0520  Linear Algebra  
or MATH 0540  Linear Algebra With Theory  
or CSCI 0530  Coding the Matrix: An Introduction to Linear Algebra for Computer Science  

CSCI 1450  Advanced Introduction to Probability for Computing and Data Science  
or APMA 1650  Statistical Inference I  
or APMA 1655  Honors Statistical Inference I  

ARTIFICIAL INTELLIGENCE / MACHINE LEARNING: studies the theory and application of algorithms for making decisions and inferences from rules and data  

Core Courses  
CSCI 1410  Artificial Intelligence  
or CSCI 1420  Machine Learning  
or CSCI 1430  Computer Vision  
or CSCI 1460  Computational Linguistics  
or CSCI 1470  Deep Learning  
or CSCI 1850  Deep Learning in Genomics  
or CSCI 1951R  Introduction to Robotics  

Related Courses  
CSCI 1440  Algorithmic Game Theory  
or CSCI 1550  Probabilistic Methods in Computer Science  
or CSCI 1951A  Data Science  
or CSCI 1951C  Designing Humanity Centered Technology  
or APMA 1740  Recent Applications of Probability and Statistics  
or ENGN 1610  Image Understanding  

Intermediate Courses  
CSCI 1450  Advanced Introduction to Probability for Computing and Data Science  
or APMA 1650  Statistical Inference I  
or APMA 1655  Honors Statistical Inference I  

CSCI 0520  Linear Algebra  
or MATH 0540  Linear Algebra With Theory  
or CSCI 0530  Coding the Matrix: An Introduction to Linear Algebra for Computer Science
THEORY: students the foundations of models and algorithms for computing in various contexts

Core Courses
- CSCI 1510 Introduction to Cryptography and Computer Security
- CSCI 1570 Design and Analysis of Algorithms
- CSCI 1760 Multiprocessor Synchronization
- CSCI 1951W Sublinear Algorithms for Big Data
- CSCI 1951X Formal Proof and Verification

Related Courses
- CSCI 1440 Algorithmic Game Theory
- CSCI 1550 Probabilistic Methods in Computer Science
- CSCI 1710 Logic for Systems
- CSCI 1810 Computational Molecular Biology
- CSCI 1820 Algorithmic Foundations of Computational Biology
- CSCI 1950H Computational Topology
- CSCI 1951G Optimization Methods in Finance
- CSCI 19590 Introduction to Computational Complexity
- CSCI 1710 Logic for Systems
- CSCI 1810 Computational Molecular Biology
- CSCI 1820 Algorithmic Foundations of Computational Biology
- CSCI 1950H Computational Topology
- CSCI 1951G Optimization Methods in Finance
- CSCI 1010 Theory of Computation
- CSCI 1450 Advanced Introduction to Probability for Computing and Data Science
- or APMA 1650 Statistical Inference I
- or APMA 1655 Honors Statistical Inference I
- MATH 0520 Linear Algebra
- or MATH 0540 Linear Algebra With Theory
- or CSCI 0530 Coding the Matrix: An Introduction to Linear Algebra for Computer Science

Intermediate Courses
- CSCI 1320 Introduction to Cryptography and Computer Security
- CSCI 1650 Software Security and Exploitation
- CSCI 1660 Introduction to Computer Systems Security
- CSCI 1320 Creating Modern Web & Mobile Applications
- CSCI 1380 Distributed Computer Systems
- CSCI 1670 Operating Systems
- CSCI 1680 Computer Networks
- CSCI 1710 Logic for Systems
- CSCI 1800 Design and Implementation of Programming Languages
- CSCI 1805 Computers, Freedom and Privacy
- CSCI 1951L Blockchains and Cryptocurrencies
- CSCI 0330 Introduction to Computer Systems
- or CSCI 0300 Fundamentals of Computer Systems
- CSCI 0220 Introduction to Discrete Structures and Probability (Or Probability and Statistics (see options below))

SECURITY: studies the design, construction, analysis, and defense of techniques to protect systems, data, and communications

Core Courses
- CSCI 1510 Introduction to Cryptography and Computer Security
- or CSCI 1550 Probabilistic Methods in Computer Science
- or CSCI 1570 Design and Analysis of Algorithms
- or CSCI 1760 Multiprocessor Synchronization
- or CSCI 1951W Sublinear Algorithms for Big Data
- or CSCI 1951X Formal Proof and Verification

Related Courses
- CSCI 1440 Algorithmic Game Theory
- CSCI 1550 Probabilistic Methods in Computer Science
- CSCI 1710 Logic for Systems
- CSCI 1810 Computational Molecular Biology
- CSCI 1820 Algorithmic Foundations of Computational Biology
- CSCI 1950H Computational Topology
- CSCI 19590 Optimization Methods in Finance
- CSCI 1010 Theory of Computation
- CSCI 1450 Advanced Introduction to Probability for Computing and Data Science
- or APMA 1650 Statistical Inference I
- or APMA 1655 Honors Statistical Inference I

Intermediate Courses
- CSCI 1320 Introduction to Cryptography and Computer Security
- CSCI 1650 Software Security and Exploitation
- CSCI 1660 Introduction to Computer Systems Security
- CSCI 1320 Creating Modern Web & Mobile Applications
- CSCI 1380 Distributed Computer Systems
- CSCI 1670 Operating Systems
- CSCI 1680 Computer Networks
- CSCI 1710 Logic for Systems
- CSCI 1800 Design and Implementation of Programming Languages
- CSCI 1805 Computers, Freedom and Privacy
- CSCI 1951L Blockchains and Cryptocurrencies
- CSCI 0330 Introduction to Computer Systems
- or CSCI 0300 Fundamentals of Computer Systems
- CSCI 0220 Introduction to Discrete Structures and Probability (Or Probability and Statistics (see options below))
Intermediate Courses

CSCI 0220 Introduction to Discrete Structures and Probability
CSCI 1010 Theory of Computation
CSCI 1450 Advanced Introduction to Probability for Computing and Data Science
or APMA 1650 Statistical Inference I
or APMA 1655 Honors Statistical Inference I

Related Courses

CSCI 1230 Introduction to Computer Graphics
or CSCI 1320 Creating Modern Web & Mobile Applications
or CSCI 1360 Human Factors in Cybersecurity
or CSCI 1600 Real-Time and Embedded Software
or CSCI 1951A Data Science
or CSCI 1951I CS for Social Change
or CSCI 1951T Surveying VR Data Visualization Software for Research
or CSCI 1951V Hypertext/Hypermedia: The Web Was Not the Beginning and the Web Is Not the End
or CSCI 1952B Responsible Computer Science in Practice
or ENGN 1931I Design of Robotic Systems
or VISA 1720 Physical Computing

Core Courses

CSCI 1300 User Interfaces and User Experience
or CSCI 1370 Virtual Reality Design for Science
or CSCI 1951C Designing Humanity Centered Technology

Concentration Requirements (9 courses)

♦︎ One complete pathway (see ScB for pathways)
♦︎ 0-3 courses
♦︎ Additional intermediate courses so that a total of three are taken with at least one in each of two different intermediate-course categories (see the ScB requirements for a listing of these categories)
♦︎ One additional 1000-level course that is neither a core nor a related course for the pathways used above
♦︎ Of the remaining two courses, at least one must be at the 1000-level or higher (i.e., one may be an intermediate course not otherwise used to satisfy a concentration requirement; this course may be CSCI 0200, an intermediate-level course, or an advanced course)
♦︎ No more than two arts, humanities, and social science oriented CS courses (currently CSCI 1250, 1280, 1370, 1800, 1805, and 1870) may be used for concentration credit.

SELF-DESIGNED: This pathway is modeled after the Brown programs for designing one’s own concentration. Students electing this pathway must write a proposal for their pathway and have it approved by an advisor and the director of undergraduate studies. The proposal must meet the breadth and overall course requirements. This must be done by the end of shopping period of the student’s seventh semester.

1 Students wishing to go directly from CSCI 0111 to CSCI 0200 (without CSCI 0112) will need to successfully complete additional exercises to receive an instructor override code for CSCI 0200.
2 Capstone: a one-semester course, taken in the student’s last undergraduate year, in which the student (or group of students) use a significant portion of their undergraduate education, broadly interpreted, in studying some current topic in depth, to produce a culminating artifact such as a paper or software project.
3 Certain 1000-level courses may be used to fill the additional 1000-level course requirements for both the AB and ScB. No more than one such course may be used for the AB concentration and no more than three for the ScB concentration. A list of approved non-CS courses is on our web page. Unless explicitly stated on our web page, such non-CS courses may not be used as part of pathways.

Requirements for the Standard Track of the A.B. degree

Prerequisites (0-3 courses) 0-3

Students must complete or place out of second semester calculus.

| MATH 0100 | Single Variable Calculus, Part II |
| MATH 0170 | Single Variable Calculus, Part II (Accelerated) |
| MATH 0190 | Single Variable Calculus, Part II (Physics/Engineering) |

Concentration Requirements (9 courses) 2

Core Computer Science:

Select one of the following series:

Series A

CSCI 0150 & CSCI 0200 Introduction to Object-Oriented Programming and Computer Science and Program Design with Data Structures and Algorithms

Series B

CSCI 0170 & CSCI 0200 Computer Science: An Integrated Introduction and Program Design with Data Structures and Algorithms

Series C

CSCI 0190 Accelerated Introduction to Computer Science (and an additional CS course not otherwise used to satisfy a concentration requirement; this course may be CSCI 0200, an intermediate-level course, or an advanced course)

Series D


Seven more advanced courses. 7

1 Students wishing to go directly from CSCI 0111 to CSCI 0200 (without CSCI 0112) will need to successfully complete additional exercises to receive an instructor override code for CSCI 0200.

Requirements for the Professional Track of the both the Sc. B. and A.B. degrees.

The requirements for the professional track include all those of the standard track, as well as the following:
Students must complete full-time professional experiences doing work that is related to their concentration programs, totaling 2-6 months, whereby each internship must be at least one month in duration in cases where students choose to do more than one internship experience. Such work is normally done at a company, but may also be at a university under the supervision of a faculty member. Internships that take place between the end of the fall and the start of the spring semesters cannot be used to fulfill this requirement.

On completion of each professional experience, the student must write and upload to ASK a reflective essay about the experience addressing the following prompts, to be approved by the student's concentration advisor:

- Which courses were put to use in your summer's work? Which topics, in particular, were important?
- In retrospect, which courses should you have taken before embarking on your summer experience? What are the topics from these courses that would have helped you over the summer if you had been more familiar with them?
- Are there topics you should have been familiar with in preparation for your summer experience, but are not taught at Brown? What are these topics?
- What did you learn from the experience that probably could not have been picked up from course work?
- Is the sort of work you did over the summer something you would like to continue doing once you graduate? Explain.
- Would you recommend your summer experience to other Brown students? Explain.

Honors

Honors candidates must have earned A's or S-with-distinction in 2/3 (rounding up) of the courses used towards the concentration, excluding introductory-sequence courses (CS courses numbered 0200 or below) and the calculus prerequisite (unless that course is also used as an intermediate math course in CS requirements).

Computer Science-Economics Concentration Requirements

The joint Computer Science-Economics concentration exposes students to the theoretical and practical connections between computer science and economics. It prepares students for professional careers that incorporate aspects of economics and computer technology and for academic careers conducting research in areas that emphasize the overlap between the two fields. Concentrators may choose to pursue either the A.B. or the Sc.B. degree. While the A.B. degree allows students to explore the two disciplines by taking advanced courses in both departments, its smaller number of required courses is compatible with a liberal education. The Sc.B. degree achieves greater depth in both computer science and economics by requiring more courses, and it offers students the opportunity to creatively integrate both disciplines through a design requirement. If you are interested in declaring a concentration in Computer Science-Economics, please refer to this page (https://economics.brown.edu/academics/undergraduate/concentrations declaración/) for more information regarding the process. For more information about the CS Pathways, see this (https://cs.brown.edu/degrees/undergrad/concentrating-in-cs/concentration-requirements-2020/pathways-for-undergraduate-and-masters-students/) page.


**Prerequisites (3 courses):**

- MATH 0100 Single Variable Calculus, Part II
- MATH 0520 Linear Algebra
  - or MATH 0540 Linear Algebra With Theory
  - or CSCI 0530 Coding the Matrix: An Introduction to Linear Algebra for Computer Science
- ECON 0110 Principles of Economics

**Required Courses: 17 courses: 8 Computer Science, 8 Economics, and a Capstone**

- CSCI 1450 Advanced Introduction to Probability for Computing and Data Science
- or APMA 1650 Statistical Inference I
- or APMA 1655 Honors Statistical Inference I

Select one of the following Series:

**Series A**

- CSCI 0150 & CSCI 0200 Introduction to Object-Oriented Programming and Computer Science and Algorithms

**Series B**

- CSCI 0170 & CSCI 0200 Computer Science: An Integrated Introduction and Program Design with Data Structures and Algorithms

**Series C**

- CSCI 0190 Accelerated Introduction to Computer Science (and an additional CS course not otherwise used to satisfy a concentration requirement; this course may be CSCI 0200, an intermediate-level CS course, or a 1000-level course.)

**Series D**

- Two of the following intermediate courses, one of which must be math-oriented and one systems-oriented.
  - CSCI 0220 Introduction to Discrete Structures and Probability (math)
  - CSCI 0320 Introduction to Software Engineering (systems)
  - CSCI 0330 Introduction to Computer Systems (systems)
  - or CSCI 0300 Fundamentals of Computer Systems
- CSCI 1010 Theory of Computation (math)

A pair of 1000-level CS courses that, along with the intermediate courses and math courses, satisfy one of the CS Pathways, as described for the CSCI ScB. An additional CS course that is either at the 1000-level or is an intermediate course not already used to satisfy concentration requirements. CSCI 1450 may not be used to satisfy this requirement.

- ECON 1130 Intermediate Microeconomics (Mathematical)
- ECON 1210 Intermediate Macroeconomics
- ECON 1630 Mathematical Econometrics I

Three courses from the "mathematical economics" group (CSCI 1951K can be counted as one of them, if it has not been used to satisfy the computer science requirements of the concentration and if the student has taken either ECON 1470 or ECON 1870):

- ECON 1170 Welfare Economics and Social Choice Theory
- ECON 1225 Advanced Macroeconomics: Monetary, Fiscal, and Stabilization Policies
- ECON 1255 Unemployment: Models and Policies
- ECON 1470 Bargaining Theory and Applications
- ECON 1490 Designing Internet Marketplaces
- ECON 1545 Topics in Macroeconomics, Development and International Economics
- ECON 1640 Mathematical Econometrics II
- ECON 1660 Big Data
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 0170</td>
<td>Computer Science: An Integrated Introduction</td>
</tr>
<tr>
<td>&amp; CSCI 0180</td>
<td>and Computer Science: An Integrated Introduction</td>
</tr>
<tr>
<td><strong>Series C</strong></td>
<td></td>
</tr>
<tr>
<td>CSCI 0190</td>
<td>Accelerated Introduction to Computer Science (and an additional CS course not otherwise used to satisfy a concentration requirement; this course may be CSCI 0200, an intermediate-level course, or a 1000-level course)</td>
</tr>
<tr>
<td><strong>Series D</strong></td>
<td></td>
</tr>
<tr>
<td>CSCI 0111</td>
<td>Computing Foundations: Data</td>
</tr>
<tr>
<td>&amp; CSCI 0180</td>
<td>and Computer Science: An Integrated Introduction</td>
</tr>
</tbody>
</table>

Two of the following intermediate courses, one of which must be math-oriented and one systems-oriented:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 0220</td>
<td>Introduction to Discrete Structures and Probability (math)</td>
</tr>
<tr>
<td>CSCI 0320</td>
<td>Introduction to Software Engineering (systems)</td>
</tr>
<tr>
<td>CSCI 0330</td>
<td>Introduction to Computer Systems (systems)</td>
</tr>
<tr>
<td>or CSCI 0300</td>
<td>Fundamentals of Computer Systems</td>
</tr>
<tr>
<td>CSCI 1010</td>
<td>Theory of Computation (math)</td>
</tr>
</tbody>
</table>

Two additional CS courses; at least one must be at the 1000-level. The other must either be at the 1000-level or be an intermediate course not already used to satisfy concentration requirements. CSCI 1450 may not be used to satisfy this requirement.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECON 1130</td>
<td>Intermediate Microeconomics (Mathematical)</td>
</tr>
<tr>
<td>ECON 1210</td>
<td>Intermediate Macroeconomics</td>
</tr>
<tr>
<td>ECON 1630</td>
<td>Mathematical Econometrics I</td>
</tr>
</tbody>
</table>

Three courses from the "mathematical-economics" group:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECON 1170</td>
<td>Welfare Economics and Social Choice Theory</td>
</tr>
<tr>
<td>ECON 1225</td>
<td>Advanced Macroeconomics: Monetary, Fiscal, and Stabilization Policies</td>
</tr>
<tr>
<td>ECON 1255</td>
<td>Unemployment: Models and Policies</td>
</tr>
<tr>
<td>ECON 1470</td>
<td>Bargaining Theory and Applications</td>
</tr>
<tr>
<td>ECON 1490</td>
<td>Designing Internet Marketplaces</td>
</tr>
<tr>
<td>ECON 1545</td>
<td>Topics in Macroeconomics, Development and International Economics</td>
</tr>
<tr>
<td>ECON 1640</td>
<td>Mathematical Econometrics II</td>
</tr>
<tr>
<td>ECON 1660</td>
<td>Big Data</td>
</tr>
<tr>
<td>ECON 1670</td>
<td>Advanced Topics in Econometrics</td>
</tr>
<tr>
<td>ECON 1680</td>
<td>Machine Learning, Text Analysis, and Economics</td>
</tr>
<tr>
<td>ECON 1750</td>
<td>Investments II</td>
</tr>
<tr>
<td>ECON 1805</td>
<td>Economics in the Laboratory</td>
</tr>
<tr>
<td>ECON 1820</td>
<td>Theory of Behavioral Economics</td>
</tr>
<tr>
<td>ECON 1850</td>
<td>Theory of Economic Growth</td>
</tr>
<tr>
<td>ECON 1860</td>
<td>The Theory of General Equilibrium</td>
</tr>
<tr>
<td>ECON 1870</td>
<td>Game Theory and Applications to Economics</td>
</tr>
<tr>
<td>or any graduate Economics course</td>
<td></td>
</tr>
</tbody>
</table>

Total Credits: 13

---

**Standard Program for the A.B. degree:**

**Prerequisites (3 courses):**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 0100</td>
<td>Single Variable Calculus, Part II</td>
</tr>
<tr>
<td>MATH 0520</td>
<td>Linear Algebra</td>
</tr>
<tr>
<td>or MATH 0540</td>
<td>Linear Algebra With Theory</td>
</tr>
<tr>
<td>or CSCI 0530</td>
<td>Coding the Matrix: An Introduction to Linear Algebra for Computer Science</td>
</tr>
<tr>
<td>ECON 0110</td>
<td>Principles of Economics</td>
</tr>
</tbody>
</table>

**Required Courses: 13 courses: 7 Computer Science and 6 Economics**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1450</td>
<td>Advanced Introduction to Probability for Computing and Data Science</td>
</tr>
<tr>
<td>or APMA 1650</td>
<td>Statistical Inference I</td>
</tr>
<tr>
<td>or APMA 1655</td>
<td>Honors Statistical Inference I</td>
</tr>
</tbody>
</table>

Select one of the following series:

**Series A**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 0150</td>
<td>Introduction to Object-Oriented Programming and Computer Science and Introduction to Algorithms and Data Structures</td>
</tr>
<tr>
<td>&amp; CSCI 0160</td>
<td></td>
</tr>
</tbody>
</table>

**Series B**

---

1 Or ECON 1110 with permission. For students matriculating at Brown in Fall 2021 or later, note that if ECON 1110 is used, then one additional course from the mathematical-economics group will be required.
CSCI 1951K can be counted as one of them, if it has not been used to satisfy the computer science requirements of the concentration and if the student has taken either ECON 1470 or ECON 1870.

Note that ECON 1620, ECON 1690, and ECON 1790 (independent study) cannot be used for concentration credit. However, 1620 and 1960 can be used for university credit and up to two 1970s may be used for university credit.

Honors

Students who meet stated requirements are eligible to write an honors thesis in their senior year. Students should consult the listed honors requirements of whichever of the two departments their primary thesis advisor belongs to, at the respective departments' websites. If the primary thesis advisor belongs to Economics (Computer Science), then students must have a reader in the Computer Science (respectively, Economics) department.

Professional Track

The requirements for the professional track include all those of the standard track, as well as the following:

Students must complete full-time professional experiences doing work that is related to their concentration programs, totaling 2-6 months, whereby each internship must be at least one month in duration in cases where students choose to do more than one internship experience. Each student is normally done at a company, but may also be at a university under the supervision of a faculty member. Internships that take place between the end of the fall and the start of the spring semesters cannot be used to fulfill this requirement.

On completion of each professional experience, the student must write and upload to ASK a reflective essay about the experience addressing the following prompts, to be approved by the student's concentration advisor:

- Which courses were put to use in your summer's work? Which topics, in particular, were important?
- In retrospect, which courses should you have taken before embarking on your summer experience? What are the topics from these courses that would have helped you over the summer if you had been more familiar with them?
- Are there topics you should have been familiar with in preparation for your summer experience, but are not taught at Brown? What are these topics?
- What did you learn from the experience that probably could not have been picked up from course work?
- Is the sort of work you did over the summer something you would like to continue doing once you graduate? Explain.
- Would you recommend your summer experience to other Brown students? Explain.

Applied Mathematics-Computer Science Concentration Requirements

The Sc.B. concentration in Applied Mathematics-Computer Science provides a foundation of basic concepts and methodology of mathematical analysis and computation and prepares students for advanced work in applied mathematics, computer science, and data science. Concentrators must complete courses in mathematics, applied math, computer science, and an approved English writing course. While the concentration in Applied Mathematics-Computer Science allows students to develop the use of quantitative methods in thinking about and solving problems, knowledge that is valuable in all walks of life, students who have completed the concentration have pursued graduate study, computer consulting and information industries, and scientific and statistical analysis careers in industry or government. This degree offers a standard track and a professional track.


Prerequisites – the equivalent of two semesters of single-variable calculus

A second semester of single-variable calculus is not an enforced requirement for our concentration, but it is a required prerequisite for many of our courses. At Brown, the second semester of calculus is taught in one of MATH 0100, MATH 0170, or MATH 0190.

Requirements – 17 courses

Completion of one CS pathway

Completion of one APMA pairing

Mathematical Requirements – 8 courses

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 0180</td>
<td>Multivariable Calculus</td>
</tr>
<tr>
<td>or MATH 0200</td>
<td>Multivariable Calculus (Physics/Engineering)</td>
</tr>
<tr>
<td>or MATH 0350</td>
<td>Multivariable Calculus With Theory</td>
</tr>
<tr>
<td>MATH 0520</td>
<td>Linear Algebra</td>
</tr>
<tr>
<td>or MATH 0540</td>
<td>Linear Algebra With Theory</td>
</tr>
<tr>
<td>or CSCI 0530</td>
<td>Coding the Matrix: An Introduction to Linear Algebra for Computer Science</td>
</tr>
<tr>
<td>or APMA 1170</td>
<td>Introduction to Computational Linear Algebra</td>
</tr>
<tr>
<td>APMA 0350</td>
<td>Applied Ordinary Differential Equations</td>
</tr>
<tr>
<td>or APMA 0360</td>
<td>Applied Partial Differential Equations</td>
</tr>
<tr>
<td>APMA 1160</td>
<td>An Introduction to Numerical Optimization</td>
</tr>
<tr>
<td>or APMA 1170</td>
<td>Introduction to Computational Linear Algebra</td>
</tr>
<tr>
<td>or APMA 1180</td>
<td>An Introduction to Numerical Solution of Differential Equations</td>
</tr>
<tr>
<td>or APMA 1690</td>
<td>Computational Probability and Statistics</td>
</tr>
<tr>
<td>or APMA 1740</td>
<td>Recent Applications of Probability and Statistics</td>
</tr>
</tbody>
</table>

Two approved 1000-level or higher APMA courses. The APMA pairing must be completed.

One 1000-level or higher APMA or MATH course

Computer Science Requirements – 8 courses

Select one of the following introductory course sequences

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 0150 &amp; CSCI 0200</td>
<td>Introduction to Object-Oriented Programming and Computer Science and Program Design with Data Structures and Algorithms</td>
</tr>
<tr>
<td>CSCI 0170 &amp; CSCI 0200</td>
<td>Computer Science: An Integrated Introduction and Program Design with Data Structures and Algorithms</td>
</tr>
<tr>
<td>CSCI 0111 &amp; CSCI 0200</td>
<td>Computing Foundations: Data and Program Design with Data Structures and Algorithms</td>
</tr>
<tr>
<td>CSCI 0190 and one CSCI course numbered 0200 or higher</td>
<td></td>
</tr>
</tbody>
</table>

Select three of the following five intermediate-level options, one of which must be math-oriented and one systems-oriented. The intermediate requirements of the chosen pathway must be completed.

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 0220</td>
<td>Introduction to Discrete Structures and Probability (math)</td>
</tr>
<tr>
<td>or MATH 1001</td>
<td>The Art of Writing Mathematics</td>
</tr>
<tr>
<td>or MATH 1530</td>
<td>Abstract Algebra</td>
</tr>
<tr>
<td>CSCI 1910</td>
<td>Theory of Computation (math)</td>
</tr>
<tr>
<td>APMA 1655</td>
<td>Honors Statistical Inference I (math)</td>
</tr>
<tr>
<td>or APMA 1650 &amp; CSCI 1450</td>
<td>Advanced Introduction to Probability for Computing and Data Science</td>
</tr>
<tr>
<td>or MATH 1620</td>
<td>Mathematical Statistics</td>
</tr>
<tr>
<td>CSCI 0320</td>
<td>Introduction to Software Engineering (systems)</td>
</tr>
<tr>
<td>CSCI 0330</td>
<td>Introduction to Computer Systems (systems)</td>
</tr>
<tr>
<td>or CSCI 0300</td>
<td>Fundamentals of Computer Systems</td>
</tr>
</tbody>
</table>
Three approved 1000-level or higher CSCI courses. The 3 advanced requirements of the chosen pathway must be completed.

### Additional Requirements – 1 course

<table>
<thead>
<tr>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

One approved capstone in computer science or applied mathematics taken in the student’s senior year.

### Total Credits

<table>
<thead>
<tr>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
</tr>
</tbody>
</table>
Is the sort of work you did over the summer something you would like to continue doing once you graduate? Explain.

Would you recommend your summer experience to other Brown students? Explain.

Honors Concentrators that demonstrate excellence in grades and in undergraduate research can be awarded departmental honors. Honors students with primary advisors in Applied Math should follow the guidelines, requirements, and deadlines for honors as described in the bulletin for Applied Math concentrators (https://bulletin.brown.edu/the-college/concentrations/apma/) and as published on the APMA departmental website (https://appliedmath.brown.edu/academics/undergraduate-program/honors/). Honors students with primary advisors in Computer Science should follow the guidelines, requirements, and deadlines for honors as described in the bulletin for Computer Science concentrators (https://bulletin.brown.edu/the-college/concentrations/comp/) and as published on the CS departmental website (https://cs.brown.edu/degrees/undergrad/concentrating-in-cs/honors/). Students wishing to do honors research with a non-APMA or CS advisor should contact the Directors of Undergraduate Studies in APMA and CS to discuss options.

Mathematics-Computer Science Concentration Requirements Students may opt to pursue an interdisciplinary Bachelor of Science degree in Math-Computer Science, a concentration administered cooperatively between the mathematics and computer science departments. Course requirements include math- and systems-oriented computer science courses, as well as computational courses in applied math. Students must identify a series of electives that cohere around a common theme. As with other concentrations offered by the departments, students have the option to pursue the professional track (http://www.cs.brown.edu/ugrad/concentrations/professional-track.html) of the ScB program in Mathematics-Computer Science.

Requirements for the Standard Track of the Sc.B. degree.

Prerequisites

Three semesters of Calculus to the level of MATH 0180, MATH 0200, or MATH 0350 3
MATH 0520 Linear Algebra 1
or MATH 0540 Linear Algebra With Theory
or CSCI 0530 Coding the Matrix: An Introduction to Linear Algebra for Computer Science

Core Courses

MATH 1530 Abstract Algebra 1
Select one of the following series: 2

Series A

CSCI 0150 & CSCI 0200 Introduction to Object-Oriented Programming and Computer Science and Program Design with Data Structures and Algorithms

Series B

CSCI 0170 & CSCI 0200 Computer Science: An Integrated Introduction and Program Design with Data Structures and Algorithms

Series C

CSCI 0190 Accelerated Introduction to Computer Science (and an additional CS course not otherwise used to satisfy a concentration requirement; an intermediate-level CS course, or a 1000-level CS course)

Series D 1

Intermediate Courses:

CSCI 0320 Introduction to Software Engineering 1
or CSCI 0300 Fundamentals of Computer Systems
or CSCI 0330 Introduction to Computer Systems
CSCI 0220 Introduction to Discrete Structures and Probability 1
or CSCI 1010 Theory of Computation

Three 1000-level Mathematics courses 3
Three advanced courses in Computer Science 2,3 3
Three additional courses different from any of the above chosen from Mathematics, Computer Science, Applied Mathematics, or related areas 4

A capstone course in Computer Science or Mathematics 5

Total Credits 19

1 Students wishing to go directly from CSCI 0111 to CSCI 0200 (without CSCI 0112) will need to successfully complete additional exercises to receive an instructor override code for CSCI 0200. In 2020-21, these exercises will be offered within CSCI 0111. Students from prior CSCI 0111 offerings should contact the current CSCI0111 instructor to arrange to do this work.

2 These must be CSCI courses at the 1000-level or higher. Two of these courses and the intermediate courses must satisfy one of the CS pathways (https://cs.brown.edu/degrees/undergrad/concentrating-in-cs/concentration-requirements-2020/pathways-for-graduate-and-masters-students/). At most one arts, humanities, or social science course can be used for concentration credit (currently CSCI 1250, 1280, 1360, 1470, 1800, 1801, 1810, 1952B, 1952X, 2002, 2952S).

3 Note: CSCI 1010 may be used either as a math-oriented intermediate course or as an advanced course. CSCI 1010 was formerly known as CSCI 510: they are the same course and hence only one may be taken for credit. CSCI 1450 was formerly known as CSCI 450: they are the same course and hence only one may be taken for credit. Applied Math 1650 or 1655 may be used in place of CSCI 1450 in CS pathway requirements (https://cs.brown.edu/degrees/undergrad/concentrating-in-cs/concentration-requirements-2020/pathways-for-graduate-and-masters-students/). However, concentration credit will be given for only one of Applied Math 1650, 1655, and CSCI 1450.

4 These must be approved by a concentration advisor.

5 A one-semester course, taken in the student's last undergraduate year, in which the student (or group of students) use a significant portion of their undergraduate education, broadly interpreted, in studying some current topic in depth, to produce a culminating artifact such as a paper or software project. The title and abstract of the artifact, along with the student's and faculty-sponsor's names, will be placed in the CS website. The inclusion of a relevant image or system diagram is strongly encouraged. The complete text of the best artifacts of each class will be featured on the CS website. A senior thesis, which involves two semesters of work, may count as a capstone Course-based capstones are currently only available through CS. Approved capstone courses and instructions may be found here: https://cs.brown.edu/degrees/undergrad/concentrating-in-cs/concentration-requirements-2020/capstone/

Requirements for the Professional Track of the Sc.B. degree.

The requirements for the professional track include all those of the standard track, as well as the following: Students must complete full-time professional experiences doing work that is related to their concentration programs, totaling 2-6 months, whereby each internship must be at least one month in duration in cases where students choose to do more than one internship experience. Such work is normally done at a company, but may also be at a university under the
supervision of a faculty member. Internships that take place between the end of the fall and the start of the spring semesters cannot be used to fulfill this requirement.

On completion of each professional experience, the student must write and upload to ASK a reflective essay about the experience addressing the following prompts, to be approved by the student's concentration advisor:

• Which courses were put to use in your summer's work? Which topics, in particular, were important?
• In retrospect, which courses should you have taken before embarking on your summer experience? What are the topics from these courses that would have helped you over the summer if you had been more familiar with them?
• Are any topics you should have been familiar with in preparation for your summer experience, but are not taught at Brown? What are these topics?
• What did you learn from the experience that probably could not have been picked up from course work?
• Is the sort of work you did over the summer something you would like to continue doing once you graduate? Explain.
• Would you recommend your summer experience to other Brown students? Explain.

Computer Science Graduate Program

The department of Computer Science offers two graduate degrees in computer science. The Master of Science (Sc.M.) degree for those who wish to improve their professional competence in computer science or to prepare for further graduate study, and the Doctor of Philosophy (Ph.D) degree.

For more information on admission, please visit the following website:
http://www.brown.edu/academics/gradschool/programs/computer-science
(http://www.brown.edu/academics/gradschool/programs/computer-science/)

Ph.D. Requirements

Requirements for the Ph.D. program can be found at https://cs.brown.edu/degrees/doctoral/reqs/reqs_phd.2015.pdf

Requirements for the Masters Degree

The requirements consist of a basic component and an advanced component. All courses must be at the 1000 level or higher. All courses must be completed with a grade of B or better.

Students who started the program in September 2020 or later must also complete UNIV 2020 "Professional Development". It is a half-credit, tuition-free course. International students must take the course in person (not online) during their first semester of residence in the program. Other students may take it at any time.

The courses in student's program must be approved by the director of the Master's program (as well as by the student's advisor).

Basic Component

The basic component consists of six courses. None of these courses may be reading and research courses such as CSCI 2980.

The six courses are chosen as follows:

• Two must be CS courses that form a pathway (see the explanation of pathways at https://cs.brown.edu/degrees/undergrad/concentrating-in-cs/concentration-requirements-2020/pathways-for-undergraduate-and-masters-students/)
• One must be a CS course in an area that’s not listed in the chosen pathway (it must not be a core course, must not be a grad course, and must not be a related course of the pathway).
• The three additional courses must be in CS or related and must be approved by your advisor or the director of graduate studies (master’s). Getting this approval will require you to show that the courses are relevant to your CS interests. In general, the more non-CS courses you wish to take, the stronger your justification must be.

Advanced Component

The advanced component requires the student to complete one of the following six options. Reading and research courses (such as CSCI 2980) may not be used as part of any of these options. Reading and research courses may not be used as part of options 5 and 6. An "advanced course," as used below, is either a 2000-level CS courses or a 1000-level CS courses that includes a Master's supplement. Master's supplement are nominally half-credit courses, but students may do the work of these courses without officially registering for them. Examples of such supplements are CSCI 1234 (supplementing CSCI 1230), CSCI 1690 (supplementing CSCI 1670), and CSCI 1729 (supplementing CSCI 1730).

"Internships", as used below, must be approved by the student's advisor and are paid work in the area of the student's master's studies. They may be full or part time. A full-time internship must last at least two months but no more than four months. A part-time internship must last at least four months but no more than six months. Normally the internship will be performed between the student's second and third semesters in the program.

The six options are:

1. Complete a thesis supervised by her or his advisor and approved by a committee consisting of the advisor and at least one other faculty member.
2. Complete a thesis supervised by her or his advisor and approved by a committee consisting of the advisor and at least one other faculty member, and complete an internship.
3. Complete a project supervised and approved by her or his advisor.
4. Complete a project supervised and approved by her or his advisor, and complete an internship.
5. Complete two advanced courses.
6. Complete two advanced courses and complete an internship.

Rationale

Students entering the Master's program typically have one of two goals: they intend to pursue research in Computer Science and are preparing themselves to enter Ph.D. programs, or they intend to become professional computer scientists and pursue careers in industry. In both cases, students should take collections of courses that not only give them strength in particular areas of Computer Science, but also include complementary areas that familiarize them with other ways of thinking about the field. For example, a student whose interests are in the practical aspects of designing computer systems should certainly take courses in this area, but should also be exposed to the mindset of theoretical computer science. In a rapidly changing discipline, there is much cross-fertilization among areas and students should have some experience in doing advanced work in areas not directly related to their own. A student whose goal is a research career should become involved as quickly as possible with a research group as part of their Master's studies, and demonstrate and learn about research by participating in it. The resulting thesis or project report will serve to establish her or his suitability for entering a Ph.D. program.

A student whose goal is to be a professional computer scientist should have some professional experience as part of her or his preparation. A certain amount of coursework is required before a student can qualify for a pedagogically useful internship. Students with limited experience in Computer Science should take a few advanced Computer Science courses before embarking on an internship. Other students, particularly those whose undergraduate degrees were at Brown, will have had internship experiences while undergraduates. Internships provide insights for subsequent courses and project work at Brown. Students without such experiences are at a disadvantage with respect to their peers. Thus we strongly encourage students who have not had the experience to choose of options 2, 4, and 6, for which internships are required.

Note that these internships are not courses and the work is not evaluated as it would be for a course. Students' advisors will assist them in choosing and obtaining internships, but it is up to students themselves to ensure that they get as much benefit as possible from their experiences. They must be able to take advantage of these experiences while completing their Master's projects – we expect as high-quality work from them as
we do from students who entered the program with prior internship experiences. A Master’s degree normally requires three to four semesters of full-time study, depending upon one’s preparation.

CSCI 1010 Theory of Computation 1
CSCI 1230 Introduction to Computer Graphics * 1
CSCI 1250 Introduction to Computer Animation 1
CSCI 1260 Compilers and Program Analysis 1
CSCI 1270 Database Management Systems 1
CSCI 1280 Intermediate 3D Computer Animation 1
CSCI 1290 Computational Photography 1
CSCI 1300 User Interfaces and User Experience 1
CSCI 1310 Fundamentals of Computer Systems 1
CSCI 1320 Creating Modern Web & Mobile Applications 1
CSCI 1330 Computer Systems 1
CSCI 1360 Human Factors in Cybersecurity 1
CSCI 1370 Virtual Reality Design for Science 1
CSCI 1380 Distributed Computer Systems 1
CSCI 1410 Artificial Intelligence 1
CSCI 1420 Machine Learning 1
CSCI 1430 Computer Vision 1
CSCI 1440 Algorithmic Game Theory 1
CSCI 1450 Advanced Introduction to Probability for Computing and Data Science 1
CSCI 1460 Computational Linguistics 1
CSCI 1470 Deep Learning 1
CSCI 1510 Introduction to Cryptography and Computer Security 1
CSCI 1550 Probabilistic Methods in Computer Science 1
CSCI 1570 Design and Analysis of Algorithms 1
CSCI 1590 Introduction to Computational Complexity 1
CSCI 1600 Real-Time and Embedded Software 1
CSCI 1610 Building High-Performance Servers 1
CSCI 1650 Software Security and Exploitation 1
CSCI 1660 Introduction to Computer Systems Security 1
CSCI 1670 Operating Systems * 1
CSCI 1680 Computer Networks 1
CSCI 1710 Logic for Systems 1
CSCI 1730 Design and Implementation of Programming Languages 1
CSCI 1760 Multiprocessor Synchronization 1
CSCI 1780 Parallel and Distributed Programming 1
CSCI 1800 Cybersecurity and International Relations 1
CSCI 1805 Computers, Freedom and Privacy 1
CSCI 1810 Computational Molecular Biology 1
CSCI 1820 Algorithmic Foundations of Computational Biology 1
CSCI 1850 Deep Learning in Genomics 1
CSCI 1860 Cybersecurity Law and Policy 1
CSCI 1870 Cybersecurity Ethics 1
CSCI 1880 Introduction to Computer Security 1
CSCI 1900 csciStartup 1
CSCI 1950N 2D Game Engines 1
CSCI 1950R Compiler Practice 1
CSCI 1950T Advanced Animation Production 1
CSCI 1950U Topics in 3D Game Engine Development 1
CSCI 1950X Software Foundations 1
CSCI 1950Z Computational Methods for Biology 1
CSCI 1951A Data Science 1
CSCI 1951C Designing Humanity Centered Technology 1
CSCI 1951G Optimization Methods in Finance 1
CSCI 1951I CS for Social Change 1
CSCI 1951J Interdisciplinary Scientific Visualization 1
CSCI 1951L Blockchains and Cryptocurrencies 1
CSCI 1951N VR+X, The Potential of Virtual Reality to Transform Nearly Everything 1
CSCI 1951P Design of Robotic Systems (ENGN 1931I) 0
CSCI 2240 Advanced Computer Graphics 1
CSCI 1951R Introduction to Robotics 1
CSCI 1951T Surveying VR Data Visualization Software for Research 1
CSCI 1951V Hypertext/Hypermedia: The Web Was Not the Beginning and the Web Is Not the End 1
CSCI 1951W Sublinear Algorithms for Big Data 1
CSCI 1951X Formal Proof and Verification 1
CSCI 1952V Algorithms for the People 1
CSCI 2270 Topics in Database Management 1
CSCI 2300 Human-Computer Interaction Seminar 1
CSCI 2310 Human Factors and User Interface Design 1
CSCI 2330 Programming Environments 1
CSCI 2340 Software Engineering 1
CSCI 2370 Interdisciplinary Scientific Visualization 1
CSCI 2390 Privacy-Conscious Computer Systems 1
CSCI 2410 Statistical Models in Natural-Language Understanding 1
CSCI 2420 Probabilistic Graphical Models 1
CSCI 2440 Advanced Algorithmic Game Theory 1
CSCI 2470 Deep Learning 1
CSCI 2500A Advanced Algorithms 1
CSCI 2500B Optimization Algorithms for Planar Graphs 1
CSCI 2510 Approximation Algorithms 1
CSCI 2520 Computational Geometry 1
CSCI 2530 Design and Analysis of Communication Networks 1
CSCI 2531 Internet and Web Algorithms 1
CSCI 2540 Advanced Probabilistic Methods in Computer Science 1
CSCI 2550 Parallel Computation: Models, Algorithms, Limits 1
CSCI 2590 Advanced Topics in Cryptography 1
CSCI 2730 Programming Language Theory 1
CSCI 2750 Topics in Parallel and Distributed Computing 1
CSCI 2840 Advanced Algorithms in Computational Biology and Medical Bioinformatics 1
CSCI 2950E Stochastic Optimization 1
CSCI 2950G Large-Scale Networked Systems 1
CSCI 2950J Cognition, Human-Computer Interaction and Visual Analysis 1
CSCI 2950K Special Topics in Computational Linguistics 1
CSCI 2950R Special Topics in Advanced Algorithms 1
CSCI 2950T Topics in Distributed Databases and Systems 1
CSCI 2950U Special Topics on Networking and Distributed Systems 1
CSCI 2950V Topics in Applied Cryptography 1
CSCI 2950W Online Algorithms 1
CSCI 2950X Topics in Programming Languages and Systems 1
CSCI 2951B Data-Driven Vision and Graphics 1
CSCI 2951E Topics in Computer Systems Security 1
CSCI 2951F Learning and Sequential Decision Making 1
CSCI 2951I Computer Vision for Graphics and Interaction 1
CSCI 2951K Topics in Collaborative Robotics 1
CSCI 2951M Advanced Algorithms Seminar 1
CSCI 2951N Advanced Algorithms in Computational Biology 1
CSCI 2951O Foundations of Prescriptive Analytics 1
CSCI 2951S Distributed Computing through Combinatorial Topology 1
CSCI 2951T Data-Driven Computer Vision 1
CSCI 2951U Topics in Software Security 1
CSCI 2951X Reintegrating AI 1
CSCI 2952B Topics in Computer Science Education Research 1
CSCI 2952C Learning with Limited Labeled Data 1
CSCI 2952F Distributed Systems at Scale: Microservices Management 1
CSCI 2952G Deep Learning in Genomics 1
CSCI 2952K Topics in 3D Computer Vision and Deep Learning 1

* Students may arrange with the instructor to receive 2000 level credit for additional coursework in CSCI 1230, 1660 or 1670

**Concurrent ScB (NUS) and ScM in Computational Biology (Brown University)**

The School of Computing at National University of Singapore and The Department of Computer Science at Brown have established a concurrent Bachelor’s and Master’s degree program in Computational Biology. After having first completed four years of undergraduate study at National University of Singapore (NUS), qualified students will attend Brown University to complete their fifth and final year of study in computational biology. After the successful completion of requirements set forth by both universities, the students will simultaneously earn both their Sc.B. and Sc.M. degrees. The Sc.B will be awarded by the National University of Singapore, while the Sc.M. is awarded by Brown University.

**Cybersecurity Graduate Program**

**Requirements for the Master of Science in Cybersecurity**

The Master of Science in Cybersecurity is designed to be completed in 4 semesters. It takes 8 courses to complete the program and students can take up to 3 courses per semester, but the Department strongly recommends taking no more than 2 courses per semester—especially during one’s first term at Brown. We do not currently offer summer term courses in the program, so courses are completed during fall and spring. Course availability varies and there is no guarantee that students will be able to take every course they are interested in.

Each Cybersecurity student must have their course choices approved by the Director of Graduate Studies for their Track and by the student’s advisor. Cybersecurity students must only register for courses at the 1000-level or higher only. Additionally, all courses must be completed with a grade of B or better. Furthermore, students must complete at least two 2000-level courses.

Choose from either the Computer Science Track or the Policy Track

**Computer Science Track:**

**Required Courses:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1360</td>
<td>Human Factors in Cybersecurity</td>
<td>1</td>
</tr>
<tr>
<td>CSCI 1860</td>
<td>Cybersecurity Law and Policy</td>
<td>1</td>
</tr>
<tr>
<td>CSCI 1880</td>
<td>Introduction to Computer Systems Security</td>
<td>1</td>
</tr>
</tbody>
</table>

**Select Three Track courses:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1310</td>
<td>Fundamentals of Computer Systems</td>
<td></td>
</tr>
<tr>
<td>CSCI 1510</td>
<td>Introduction to Cryptography and Computer Security</td>
<td></td>
</tr>
<tr>
<td>CSCI 1650</td>
<td>Software Security and Exploitation</td>
<td></td>
</tr>
<tr>
<td>CSCI 1670</td>
<td>Operating Systems</td>
<td></td>
</tr>
<tr>
<td>CSCI 1680</td>
<td>Computer Networks</td>
<td></td>
</tr>
<tr>
<td>CSCI 1690</td>
<td>Operating Systems Laboratory</td>
<td></td>
</tr>
<tr>
<td>CSCI 1730</td>
<td>Design and Implementation of Programming Languages</td>
<td></td>
</tr>
<tr>
<td>CSCI 2390</td>
<td>Privacy-Conscious Computer Systems</td>
<td></td>
</tr>
<tr>
<td>CSCI 2590</td>
<td>Advanced Topics in Cryptography</td>
<td></td>
</tr>
<tr>
<td>CSCI 2950V</td>
<td>Topics in Applied Cryptography</td>
<td></td>
</tr>
<tr>
<td>CSCI 2951E</td>
<td>Topics in Cybersecurity Security</td>
<td></td>
</tr>
<tr>
<td>CSCI 2951U</td>
<td>Topics in Software Security</td>
<td></td>
</tr>
</tbody>
</table>

**Select Two Elective Courses:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1010</td>
<td>Theory of Computation</td>
</tr>
<tr>
<td>CSCI 1270</td>
<td>Database Management Systems</td>
</tr>
<tr>
<td>CSCI 1300</td>
<td>User Interfaces and User Experience</td>
</tr>
<tr>
<td>CSCI 1380</td>
<td>Distributed Computer Systems</td>
</tr>
<tr>
<td>CSCI 1410</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>CSCI 1420</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>CSCI 1450</td>
<td>Advanced Introduction to Probability for Computing and Data Science</td>
</tr>
<tr>
<td>CSCI 1470</td>
<td>Deep Learning</td>
</tr>
<tr>
<td>CSCI 1570</td>
<td>Design and Analysis of Algorithms</td>
</tr>
<tr>
<td>CSCI 1800</td>
<td>Cybersecurity and International Relations</td>
</tr>
<tr>
<td>CSCI 1805</td>
<td>Computers, Freedom and Privacy</td>
</tr>
<tr>
<td>CSCI 1870</td>
<td>Cybersecurity Ethics</td>
</tr>
<tr>
<td>CSCI 1951R</td>
<td>Introduction to Robotics</td>
</tr>
<tr>
<td>CSCI 2980</td>
<td>Reading and Research</td>
</tr>
</tbody>
</table>

Other appropriate courses are possible with the approval of the Director of Graduate Studies and the instructor if you’ve satisfied their prerequisites.

**Policy Track:**

**Required Courses:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1360</td>
<td>Human Factors in Cybersecurity</td>
<td>1</td>
</tr>
<tr>
<td>CSCI 1860</td>
<td>Cybersecurity Law and Policy</td>
<td>1</td>
</tr>
<tr>
<td>CSCI 1880</td>
<td>Introduction to Computer Systems Security</td>
<td>1</td>
</tr>
</tbody>
</table>

**Select Three Track courses:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 2980</td>
<td>Reading and Research</td>
<td></td>
</tr>
<tr>
<td>CSCI 2001</td>
<td>Applied Cryptography and Data Privacy</td>
<td></td>
</tr>
<tr>
<td>CSCI 2002</td>
<td>Privacy and Personal Data Protection</td>
<td></td>
</tr>
</tbody>
</table>

Other appropriate courses are possible with the approval of the Director of Graduate Studies and the instructor if you have satisfied their prerequisites.

**Select Two Elective Courses:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1300</td>
<td>User Interfaces and User Experience</td>
</tr>
<tr>
<td>CSCI 1800</td>
<td>Cybersecurity and International Relations</td>
</tr>
<tr>
<td>CSCI 1805</td>
<td>Computers, Freedom and Privacy</td>
</tr>
<tr>
<td>CSCI 1870</td>
<td>Cybersecurity Ethics</td>
</tr>
<tr>
<td>CSCI 1951L</td>
<td>Blockchains and Cryptocurrencies</td>
</tr>
</tbody>
</table>
Courses

CSCI 0020. The Digital World.
Removes the mystery surrounding computers and the ever-growing digital world. Introduces a range of topics and many aspects of multimedia, along with explanations of the underlying digital technology and its relevance to our society. Other topics include artificial intelligence, IT security, ethics and the economics of computing as well as the effects of its pervasiveness in today's world. Introductory programming and analytic skills are developed through HTML, CSS, Javascript, and Python assignments. CSCI0020 is a good introduction to a wide range of CS topics that have broad relevance in our society. No prerequisites. Cannot be taken to fulfill CS Concentration.

Fall CSCI0020  S01  17412  TTh  9:00-10:20(05) (D. Stanford)

CSCI 0030. Introduction to Computation for the Humanities and Social Sciences.
Introduces students to the use of computation for solving problems in the social sciences and the humanities. We will investigate a series of real-world problems taken from the news, from books such as Freakonomics, and from current research. Topics covered include data gathering, analysis, and visualization; web-based interfaces; algorithms; and scripting. Enrollment limited to 20. Instructor permission required.

CSCI 0040. Introduction to Scientific Computing and Problem Solving
CSCI0040 provides an introduction to using computers to solve STEM (Science, Technology, Engineering and Mathematics) data analysis, visualization and simulation problems from engineering, neuroscience, biology, mathematics and finance.

Students will access and analyze a number of "real world" data sets while becoming fluent MATLAB programmers. Other tools utilized may include Excel, Wolframalpha and Python.

By course end, students should be able to use MATLAB to solve a large variety of scientific data analysis, visualization and simulation problems.

No prior programming experience is required (MATLAB is easy and fun to use).

CSCI 0050. A Data-Centric Introduction to Programming.
An introduction to computer programming with a focus on skills needed for data-intensive applications. Topics include core constructs for processing both tabular and structured data; decomposing problems into programming tasks; data structures; algorithms; and testing programs for correct behavior.

CSCI 0060. Practical System Skills.
An introduction to develop hands-on-computing skills necessary to comfortably work within a UNIX-like operating system. Topics include the shell, its filesystem, bash scripting, SSH, version control, as well as how to locally develop, deploy and publish a website. https://cs.brown.edu/ courses/csci0060/

CSCI 0080. A First Byte of Computer Science.
Introduces non-CS concentrators to the academic discipline of computer science, its thought processes, and its relevance to other fields and modern life more generally. The target audience is students who are interested in learning more about what computer science is about and the ideas it has to offer tomorrow's citizens and scholars. Topics include the basics of computation and programming, a taste of theoretical computer science and algorithms, and an introduction to codes and artificial intelligence. Although students will learn to read and understand short programs, the course will not teach or require advanced programming skills.

CSCI 0081. TA Apprenticeship: Full Credit.
Being an undergraduate TA is a learning experience: one not only gets a deeper understanding of the course material, but gains management and social skills that are invaluable for one's future. Students taking this course must first be selected as an undergraduate TA for a Computer Science course, a course the student has taken and done well in. Students will work with the course's instructor on a variety of course-related topics, including preparation of material and development of assignments. Whether CSCI 0081 or its half-credit version (CSCI 0082) is taken is up to the professor of the course being TA'd. Instructor permission required.

Fall CSCI0081 S01  17414  Arranged (T. Doeppner)

CSCI 0082. TA Apprenticeship: Half Credit.
Being an undergraduate TA is a learning experience: one not only gets a deeper understanding of the course material, but gains management and social skills that are invaluable for one's future. Students taking this course must first be selected as an undergraduate TA for a Computer Science course, a course the student has taken and done well in. Students will work with the course's instructor on a variety of course-related topics, including preparation of material and development of assignments. Whether CSCI 0082 or its full-credit version (CSCI 0081) is taken is up to the professor of the course being TA'd. Instructor permission required.

Fall CSCI0082 S01  17415  Arranged 'To Be Arranged'

CSCI 0100. Data Fluency for All.
This course is intended to introduce Brown students to computational techniques that data scientists use to tell stories. Data fluency encompasses both data literacy, the basics of statistics and machine learning, and data communication, which relies heavily on principles of design. Students will gain hands on experience using statistical tools such as 'R' to analyze real world data sets, and 'ggplot' to visualize them. Sample application domains include just about every field, since the only requirement is data, which there almost always are (e.g., the complete works of Shakespeare is a sample data set).

An introduction to computing and programming that focuses on understanding and manipulating data. Students will learn to write programs to process both tabular and structured data, to assess programs both experimentally and theoretically, to apply basic data science concepts, and to discuss big ideas around the communication, use, and social impacts of digital information. Designed for both concentrators and non-concentrators, this is the first course in either a two- or three-course introductory sequence leading into advanced CS courses. Programming assignments will be smaller scale than in CSCI 0150/0170, thus allowing students time to practice programming and discuss computational ideas in a broader context.

Fall CSCI0111 S01  17625  MWF  10:00-10:50(04) (M. Zizyte)
Fall CSCI0111 S02  17626  Arranged(04)  (M. Zizyte)

Explores how organization of programs, data, and algorithms affects metrics such as time performance, space usage, social impacts, and data privacy. Students will learn how to choose between candidate data structures for a problem, how to write programs over several standard data structures, how to assess the quality of programs (from theoretical, practical, and social perspectives), and how to apply their skills to computational problems that could arise in a variety of fields. The course will teach object-oriented programming, in combination with basic functional and imperative programming concepts. The course is designed for both concentrators and non-concentrators. Prerequisite: CSCI 0111

Fall CSCI0112 S01  17465  MWF  1:00-1:50(08) (T. Nelson)
CSCI 0130. User Interfaces and User Experience.
Have you ever had trouble using someone else’s microwave? Have you ever wondered why keyboards are ordered “qwertyuiop”? We will focus on hands-on experience to learn when to use different interfaces, how to model and represent user interaction, how to elicit requirements and feedback from users, as well as the principles of user experience design, methods for designing and prototyping interfaces, and user interface evaluation. Students interested in gaining hands-on experience designing a user interface as well as learning the process behind building an effective interface should take this course. There will be assignments, readings, and workshop time, where students will have the opportunity to work alongside each other as they learn critical tools for interface and web design. This course is open to students that have not taken CSCI 1300 or CSCI 0130 in the past.

CSCI 0150. Introduction to Object-Oriented Programming and Computer Science.
Introduces programming in Java (a modern, widely-used programming language), interactive 2D computer graphics, and some fundamental data structures and algorithms. Students learn by programming a sequence of interactive graphics programs which gradually increase in complexity, including Doodle Jump, Tetris (http://bastilleweb.technohead.org/), and a significant final project. Lectures are supplemented by skits performed by the UTAs (Undergraduate Teaching Assistants) to teach course concepts and for a bit of added entertainment! This course is intended for both potential concentrators and those who may take only a single course. There are NO prerequisites, and no prior knowledge of programming is required, though students who do have prior programming experience are also encouraged to take the course!

CSCI 0190. Accelerated Introduction to Computer Science.
A one-semester introduction to CS covering programming integrated with core data structures, algorithms, and analysis techniques, similar to the two-course introductory sequences (CSCI 0150-0200 and CSCI 0170-0200). All students wishing to take CSCI 0190, irrespective of prior preparation, must pass a sequence of online placement assignments during the summer. Though the placement process is most appropriate for students who have had some prior programming experience, it is self-contained so all are welcome to try learning the provided material and attempting placement. Placement information will be available by June 1st at http://cs.brown.edu/courses/cscio190/. Students who do not successfully pass the placement process won’t be allowed to register. Fall CSCI0190 S01 17457 MWF 9:00-9:50(09) (S. Krishnamurthi)

CSCI 0200. Program Design with Data Structures and Algorithms.
Students extend their program-design skills while learning multiple data structures, common graph algorithms, different forms of societal impacts from programs, how to analyze programs for performance, and how to work effectively with multiple styles of programming languages. Examples and course projects draw from several areas of computer science to help students identify their broader interests within the field. There will be a required weekly lab session involving hands-on work with course material. Prerequisite: CSCI 0112, CSCI 0150, 0170, or CSCI 0190. In addition, CSCI 0111 can be used if additional work and the instructor’s permission. The first two weeks of the course will be taught as at least two parallel tracks based on which prerequisite course a student has taken. CSCI 0200 will be offered every semester (fall and spring).

CSCI 0210. Computer Science: An Integrated Introduction.
CSCI0210/0180 is an introductory sequence that helps students begin to develop the skills, knowledge, and confidence to solve computational problems elegantly, correctly, efficiently, and with ease. The sequence is unique in teaching both the functional and imperative programming paradigms—the first through the languages Scheme and ML in CSCI0170; the second through Java in CSCI0180. The sequence requires no previous programming experience. Indeed, few high school students are exposed to functional programming; hence even students with previous programming experience often find this sequence an invaluable part of their education.

Although students are taught to use programming languages as tools, the goal of CSCI0170/0180 is not merely to teach programming. On the contrary, the goal is to convey to students that computer science is much more than programming! All of the following fundamental computer science techniques are integrated into the course material: algorithms, data structures, analysis, problem solving, abstract reasoning, and collaboration. Concrete examples are drawn from different subareas of computer science: in 0170, from arbitrary-precision arithmetic, natural language processing, databases, and strategic games; in 0180, from discrete-event simulation, data compression, and client/server architectures.

Fall CSCI0170 S01 17445 MWF 11:00-11:50(16) (J. Hughes)

Fall CSCI0180 S01 17446 MWF 12:00-12:50(17) (J. Hughes)

CSCI 0220. Introduction to Discrete Structures and Probability.
Seeks to place on solid foundations the most common structures of computer science, to illustrate proof techniques, to provide the background for an introductory course in computational theory, and to introduce basic concepts of probability theory. Introduces Boolean algebras, logic, set theory, elements of algebraic structures, graph theory, combinatorics, and probability. No prerequisites.

Fall CSCI0220 S02 17459 Arranged(11) "To Be Arranged"
Covers fundamental concepts, principles, and abstractions that underlie the design and engineering of computer systems. Students will learn how a computer works, how to write safe and performant systems software, and what systems abstractions support today’s complex, high-performance systems developed in industry. Specific topics include machine organization, systems programming and performance, key concepts of operating systems, isolation, security, virtualization, concurrent programming, and the basics of distributed systems. Combined lectures, labs, and several hands-on projects involving programming exercises in C/ C++. Prerequisites: CSCI 0160, 0180, 0190, or 0200; or permission of the instructor.

CSCI 0310. Introduction to Computer Systems.
Basic principles of computer organization. Begins with machine representation of data types and logic design, then explores architecture and operations of computer systems, including I/O, pipelining, and memory hierarchies. Uses assembly language as an intermediate abstraction to study introductory operating system and compiler concepts. Prerequisite: CSCI 0150 or CSCI 0180 or CSCI 0190.

CSCI 0320. Introduction to Software Engineering.
Focuses on designing, building, testing, and maintaining systems collaboratively. It covers programming techniques (using Java and TypeScript with various frameworks), object-oriented design, advanced testing (e.g., fuzz testing), debugging approaches, and tools such as source control systems. The course concludes with a major project that students must present, which should be the design and implementation of the system and its documentation. Prerequisite: CSCI 0160, 0180, 0190 or CSCI 0200. CSCI 0220 is recommended.

CSCI 0330. Introduction to Computer Systems.
High-level computer architecture and systems programming. The course covers the organization of computer systems (in terms of storage units, caches, processors, and I/O controllers) and teaches students assembly-language programming and C-language programming. Extensive programming exercises introduce students to systems-level programming on Unix systems, as well as to multi-threaded programming with POSIX threads. Students will be introduced to the functions of operating systems. Prerequisite: CSCI 0160, 0180, 0190, or 0200.

CSCI 0330. Coding the Matrix: An Introduction to Linear Algebra for Computer Science.
Introduces vectors, matrices and their role in computer science in three components: (1) concepts, theorems, and proofs, (2) procedures and programs, (3) applications and working with data. Weekly lab sessions where students apply concepts to a real task with real data. Example labs: transformations in 2-d graphics, error-correcting codes, image compression using wavelets, synthesizing a new perspective in a photo, face recognition, news story categorization, cancer diagnosis using machine learning, matching airplanes to destinations, Google’s PageRank method. Other topics as time allows. Skills in programming and prior exposure to reading and writing mathematical proofs required.

CSCI 0535. Linear Algebra for Machine Learning.
The goal of this course is to provide firm foundations in linear algebra and optimization techniques that will enable students to analyze and solve problems arising in various areas of data science, especially machine learning and data analysis. The students will acquire a firm theoretical knowledge of these concepts and tools. You will also learn how to use these tools in practice by tackling various judiciously chosen projects (from Machine Learning, etc.).

CSCI 1010. Theory of Computation.
The course introduces basic models of computation including languages, finite-state automata and Turing machines. Proves fundamental limits on computation (incomputability, the halting problem). Provides the tools to compare the hardness of computational problems (reductions). Introduces computational complexity classes (P, NP, PSPACE and others). Prerequisite: CSCI 0220 or CSCI 01450 or CSCI 01550 or APMA 01650/01655 or CSCI 01570.

This course will cover cryptographic concepts such as data privacy, encryption, authentication, digital signatures, differential privacy, enhancing technologies, secure computation, and electronic money. The emphasis will be on how to use cryptographic systems correctly in a larger context, rather than on the mathematical details of how they work; although we will cover some of those details too, on a high level. This course will be aimed at practicing and aspiring poets, economists, software engineers, law and policy wonks, and business tycoons. No prerequisites.

Fundamental concepts in 2D and 3D computer graphics, e.g., 2D raster graphics techniques and simple image processing. Focuses on geometric transformations, and 3D modeling, viewing and rendering. A sequence of assignments in C++ culminates in a simple geometric modeler and ray tracer. Prerequisite: CSCI 0160, CSCI 0180, CSCI 0190, or CSCI 0200. Some knowledge of basic linear algebra is helpful but not required. Strong object-oriented programming ability (e.g., in C++, Java or Python) is required.

CSCI 1234 is a half-credit course intended to be taken concurrently with CSCI 1230 and provides students with a greater understanding of the material by having them extend each of 1230's assignments to greater depth.

CSCI 1250. Introduction to Computer Animation.
Introduction to 3D computer animation production including story writing, production planning, modeling, shading, animation, lighting, and compositing. The first part of the course leads students through progressive exercises that build on each other to learn basic skills in 2D and 3D animation. At each step, student work is evaluated for expressiveness, technical correctness and aesthetic qualities. Students then work in groups creating a polished short animation. Emphasis on in-class critique of ongoing work which is essential to the cycle of visually evaluating work in progress, determining improvements, and implementing them for further evaluation.

CSCI 1260. Compilers and Program Analysis.
Lexical analysis, syntactic analysis, semantic analysis, code generation, code optimization, translator writing systems. Prerequisites: CSCI 0220, or CSCI 0320, or CSCI 0300, or CSCI 0330, or CSCI 1310, or CSCI 1330.

CSCI 1270. Database Management Systems.
Introduction to database structure, organization, languages, and implementation. Relational model, query languages, query processing, query optimization, normalization, file structures, concurrency control and recovery algorithms, and distributed databases. Coverage of modern applications such as the Web, but with emphasis on Database Management Systems internals. Prerequisites: CSCI 0160, CSCI 0180, or CSCI 0190. One of CSCI 0300, 0330 or CSCI 0320 is strongly recommended.
CSCI 1280. Intermediate 3D Computer Animation.
Continues work begun in CSCI 1250 with deeper exploration of technical and artistic aspects of 3D computer animation including more sophisticated shading and lighting methods and character modeling, rigging, animation, and dynamics. After a series of individual exercises, students pursue an independent topic and then, working alone or in pairs, create a polished demonstration. Emphasis is on in-class critique of ongoing work. Prerequisite: CSCI 1250. Students may contact the instructor in December for permission.

CSCI 1290. Computational Photography.
Describes the convergence of computer graphics and computer vision with photography. Its goal is to overcome the limitations of traditional photography using computational techniques to enhance the way we capture, manipulate, and interact with visual media. Topics covered: cameras, human visual perception, image processing and manipulation, image based lighting and rendering, high dynamic range, single view reconstruction, photo quality assessment, non photorealistic rendering, the use of Internet-scale data, and more. Students are encouraged to capture and process their own data. Prerequisites: previous programming experience, basic linear algebra, calculus, and probability; previous knowledge of computer graphics or computer vision. Strongly recommended: CSCI 1230, CSCI 1430, ENGN 1610.

CSCI 1300. User Interfaces and User Experience.
Have you ever walked into a door thinking that you were supposed to pull instead of push? Have you ever been stuck on a website, not sure how to proceed next? Learn when to use different interfaces, how to model and represent user interaction, how to elicit requirements and feedback from users, as well as the principles of user experience design, methods for designing and prototyping interfaces, and user interface evaluation. Students interested in both learning the process behind building an effective interface and gaining hands-on experience designing a user interface should take this course. There will be assignments, readings, and studios, where students will have the opportunity to work alongside TAs and interact with industry guests as they learn critical tools for interface and web design. Website: http://cs.brown.edu/courses/csci1300/

Covers fundamental concepts, principles, and abstractions that underlie the design and engineering of computer systems, with reference to applications of these concepts in industry. Topics include machine organization, systems programming and performance, key concepts of operating systems, isolation, security, virtualization, concurrent programming, and the basics of distributed systems. Combined lectures, case studies, labs, and several hands-on projects involving programming exercises. This course is intended for Computer Science Master's students only. Anyone else wanting to take the course should contact the instructor.

This course covers all aspects of web application development, including initial concept, user-centric design, development methodologies, front and back end development, databases, security, testing, load testing, accessibility, and deployment. There will be a substantial team project. The course is designed for students with a programming background (equiv CSCI 0320/CSCI 0330/CSCI 0300) who want to learn how to build web applications, and for students with a background in web design, including HTML and Javascript, who are interested in learning how to extend design techniques to incorporate the technologies needed in modern web applications. Project teams will consist of students with both backgrounds.

High-level computer architecture and systems programming. The course covers the organization of computer systems (in terms of storage units, caches, processors, and I/O controllers) and teaches students assembly-language programming and C-language programming. Extensive programming exercises introduce students to systems-level programming on Linux systems, as well as to multi-threaded programming with POSIX threads. Students will be introduced to the functions of operating systems. Enrollment limited to Master's students only.

CSCI 1340. Introduction to Software Engineering.
CSCI 1340 focuses on designing, building, testing, and maintaining systems collaboratively. It covers programming techniques (using Java and TypeScript with various frameworks), object-oriented design, advanced testing (e.g., fuzz testing), debugging approaches, and tools such as source control systems. The course concludes with a major group project that students gather requirements for, then design and implement themselves.

Note: CSCI 1340 is for Master's students only (they may not register for CSCI 0320). It is identical to 0320 but with the addition of supplemental work for each sprint. Prerequisite: CSCI 0160, CSCI 0180, CSCI 0190 or CSCI 0200; CSCI 0220 is recommended.

CSCI 1360. Human Factors in Cybersecurity.
This course is designed to push you to think about cybersecurity as an idea with both physical and virtual elements. Throughout the course, we will examine the value of information, the importance of users, and the difficult balance between security and usability. The ultimate goal of this course is to give you the intellectual and scientific framework you need to create systems that are both secure and efficient to use. The course focuses on usable security practices, but also looks deeply at the way our society influences security.

Fall CSCI1360 S01 17480 W 3:00-5:30 (E. Zaldivar)
Fall CSCI1360 S02 18502 W 3:00-5:30(10) (E. Zaldivar)

Explores the visual and human-computer interaction design process for scientific applications in Brown's immersive virtual reality Cave. Joint with RISD. Computer Science and design students learn how to work together effectively; study the process of design; learn about scientific problems; create designs applications; critique, evaluate, realize and iterate designs; and demonstrate final projects. Instructor permission required.

CSCI 1380. Distributed Computer Systems.
Explores the fundamental principles and practice underlying networked information systems, first we cover basic distributed computing mechanisms (e.g., naming, replication, security, etc.) and enabling middleware technologies. We then discuss how these mechanisms and technologies fit together to realize distributed databases and file systems, web-based and mobile information systems. Prerequisite: CSCI 0300, CSCI 0320, CSCI 0330, CSCI 1310 or CSCI 1330.

CSCI 1410. Artificial Intelligence.
Algorithms and representations used in artificial intelligence. Introduction and implementation of algorithms for search, planning, perception, knowledge representation, logic, probabilistic representation and reasoning, robotics and machine learning.

How can artificial systems learn from examples and discover information buried in data? We explore the theory and practice of statistical machine learning, focusing on computational methods for supervised and unsupervised learning. Specific topics include empirical risk minimization, probably approximately correct learning, kernel methods, neural networks, maximum likelihood estimation, the expectation maximization algorithm, and principal component analysis. This course also aims to expose students to relevant ethical and societal considerations related to machine learning that may arise in practice. Please contact the instructor for information about the waitlist.

CSCI 1430. Computer Vision.
How can we program computers to understand the visual world? This course treats vision as inference from noisy and uncertain data and emphasizes probabilistic and statistical approaches. Topics may include perception of 3D scene structure from stereo, motion, and shading; segmentation and grouping; texture analysis; learning, object recognition; tracking and motion estimation. Strongly recommended: basic linear algebra, calculus, and probability.
CSCI 1440. Algorithmic Game Theory. 
This course examines topics in game theory and mechanism design from a computer scientist’s perspective. Through the lens of computation, the focus is the design and analysis of systems utilized by self-interested agents. Students will investigate how the potential for strategic agent behavior can/should influence system design, and the ramifications of conflicts of interest between system designers and participating agents. Emphasis on computational tractability is paramount, so that simple designs are often preferred to optimal. Students will learn to analyze competing designs using the tools of theoretical computer science, and empirical tools, such as empirical game-theoretic analysis. Application areas include computational advertising, wireless spectrum, and prediction markets.

CSCI 1450. Advanced Introduction to Probability for Computing and Data Science. 
Probability and statistics have become indispensable tools in computer science. Probabilistic methods and statistical reasoning play major roles in machine learning, cryptography, network security, communication protocols, web search engines, robotics, program verification, and more. This course introduces the basic concepts of probability and statistics, focusing on topics that are most useful in computer science applications. Topics include: modeling and solution in sample space, random variables, simple random processes and their probability distributions, Markov processes, limit theorems, and basic elements of Bayesian and frequentist statistical inference. Basic programming experience required for homework assignments.

Students cannot get concentration credit for both CSCI 1450 and APMA 1650/ APMA 1655.
Fall CSCI1450 S01 17482 TTh 2:30-3:50(12) (E. Upfal)

The application of computational methods to problems in natural-language processing. In particular we examine techniques due to recent advances in deep learning: word embeddings, recurrent neural networks (e.g., LSTMs), sequence-to-sequence models, and generative adversarial networks (GANs). Programming projects include parsing, machine translation, question answering, and chat-bots.

Fall CSCI1460 S01 17580 TTh 2:30-3:50(12) (E. Pavlick)

What is deep learning? How is it related to machine learning? How is it applied to perform tasks like classifying images or translating languages? Deep Learning belongs to a broader family of machine learning methods. Deep learning-based methods (e.g., convolutional neural networks, recurrent neural networks, autoencoders) have led to rapid improvements in applications like computer vision, natural language processing, robotics, and even genomics and health. In this course, you will get an overview of the prominent techniques of deep learning and their applications. This course is designed to help you understand the underlying concepts as well as the promise and pitfalls of deep learning. It also aims at providing hands-on practice of implementing and applying deep learning methods in Python.

CSCI 1480. Building Intelligent Robots. 
How do robots function autonomously in dynamic, unpredictable environments? This course focuses on programming mobile robots, such as the iRobot Roomba, to perceive and act autonomously in real-world environments. The major paradigms for autonomous control and robot perception are examined and compared with robotic notions in science fiction. Prerequisite: CSCI 0150, CSCI 0170 or CSCI 0190. Recommended: CSCI 1410 or CSCI 1230.

CSCI 1490. Introduction to Combinatorial Optimization. 
This course covers the algorithmic aspects of optimizing decisions in fully observable, non-changing environments. Students are introduced to state-of-the-art optimization methods such as linear programming, integer programming, local search, and constraint programming. Strongly recommended: CSCI 0160, CSCI 0180 or CSCI 0190; CSCI 0510; and CSCI 0530 or MATH 0520 or MATH 0540.

CSCI 1510. Introduction to Cryptography and Computer Security. 
This course studies the tools for guaranteeing safe communication and computation in an adversarial setting. We develop notions of security and give provably secure constructions for such cryptographic objects as cryptosystems, signature schemes and pseudorandom generators. We also review the principles for secure system design. Prerequisites: CSCI 0220, and either CSCI 0510 or CSCI 1010.
Fall CSCI1510 S01 17483 TTh 10:30-11:50(13) (P. Miao)

Randomization and probabilistic techniques play an important role in modern computer science, with applications ranging from combinatorial optimization and machine learning to communications networks and secure protocols. This course introduces the most fundamental probabilistic techniques used in computer science applications, in particular in randomized algorithms, probabilistic analysis of algorithms and machine learning.
Prerequisite: Basic background in probability theory course such as CSCI 1450.

CSCI 1570. Design and Analysis of Algorithms. 
A single algorithmic improvement can have a greater impact on our ability to solve a problem than ten years of incremental improvements in CPU speed. We study techniques for designing and analyzing algorithms. Typical problem areas addressed include hashing, searching, dynamic programming, graph algorithms, network flow, and optimization algorithms including linear programming. Prerequisites: CSCI 0160, CSCI 0180, or CSCI 0190, and one of CSCI 0220, CSCI 1010, CSCI 1450, MATH 0750, MATH 1010, MATH 1530.
Fall CSCI1570 S01 17484 TTh 2:30-3:50(12) (L. De Stefani)

Half-credit course intended to be taken with CSCI 1570. Students will explore each topic in greater depth by collaboratively solving homework problems that will reinforce valuable new perspectives on the material. Corequisite: CSCI 1570.

CSCI 1580. Information Retrieval and Web Search. 
Covers traditional material as well as recent advances in information retrieval (IR), the study of indexing, processing, and querying of textual data. The focus will be on newer techniques geared to hypertext documents available on the World Wide Web. Topics include efficient text indexing; Boolean and vector space retrieval models; evaluation and interface issues; Web crawling, link-based algorithms, and Web metadata; text/Web clustering, classification; text mining.

CSCI 1590. Introduction to Computational Complexity. 
Introduction to serial and parallel models of computation; time and space complexity classes on these models; the circuit model of computation and its relation to serial and parallel time complexity; space-time tradeoffs on serial computers; area-time tradeoffs on the VLSI computational model; interactive and probabilistically checkable proofs; the definition of NP in terms of probabilistically checkable proofs; hardness of approximations to solutions to NP-hard problems. Prerequisite: CSCI 0510.

CSCI 1600. Real-Time and Embedded Software. 
Comprehensive introduction to the design and implementation of software for programmable embedded computing systems, in applications such as Internet of Things, transportation, and mobile. Includes the overall embedded real-time software design and development processes, with a focus on engineering for reliability. Major project component. Prerequisites: one of CSCI 0300, CSCI 0320, CSCI 0330, CSCI 1310, or CSCI 1330.
Fall CSCI1600 S01 17486 MWF 12:00-12:50(15) (M. Zizyte)
Fall CSCI1600 L01 17487 T 4:00-7:00 (M. Zizyte)
Fall CSCI1600 L02 17488 M 5:30-8:30PM (M. Zizyte)

CSCI 1610. Building High-Performance Servers. 
In depth study of modern server design. Considers architectures for building high-performance, robust, scalable, and secure network servers. We will consider all aspects of "mission-critical" servers. Topics include multithreaded and asynchronous programming techniques, database access, performance profiling, security, and redundancy. Teams will build significant projects. Prerequisite: CSCI 0320 or 0360. CSCI 1670 or 1680 is recommended.
This course is a half-credit lab intended to be taken concurrently with CSCI 1660 and provides students with a deeper understanding of the material by doing advanced versions of the cs1660's projects. These advanced versions focus on real-world skills: performing attacks that are more difficult and rely on less serious vulnerabilities, performing attacks against systems with more real-world constraints, and creating attacks that achieve a higher standard of quality than a mere “proof of concept.” Instructor permission required.

CSCI 1650. Software Security and Exploitation.
CSCI 1650 covers software exploitation techniques and state-of-the-art mechanisms for hardening software. The course begins with a summary of prevalent software defects, typically found in applications written in memory unsafe languages, like C/C++, and proceeds with studying traditional and modern exploitation techniques, ranging from classical code injection and code reuse up to the latest goodies (e.g., JIT-ROP). For the most part, it focuses on defenses against certain vulnerability classes and the way(s) to bypass them. Students will be introduced to advanced software exploitation techniques and countermeasures, and study (in depth) the boundaries and effectiveness of standard hardening mechanisms, such as address space randomization and stack and heap protections.

Fall CSCI1650 S01 17489 MW 3:00-4:20(10) (V. Kemerlis)

This course teaches principles of computer security from an applied viewpoint and provides hands-on experience on security threats and countermeasures. Topics include code execution vulnerabilities (buffer overflow, sandboxing, mobile code), malware (trojans, viruses, and worms), access control (users, roles, policies), cryptosystems (hashing, signatures, certificates), network security (firewalls, TLS, intrusion detection, VPN), and human and social issues. Prerequisites: one of (CSCI 0160, 0180, or 0190) and (CSCI 0300, 0330, 1310, or 1330). To be added to the course waitlist, please fill out this form: https://forms.gle/pHPAy9ntQkAQ7xLD9

Covers not just the principles of operating systems but the intricacies of how they work. Topics include multithreaded programming, managing threads and interrupts, managing storage, processor scheduling, operating-system structure, virtualization, security, and the design of file systems (both local and distributed). Extensive examples are taken from actual systems, including Linux and Windows. Students are expected to complete both problem sets and programming assignments (in C). Prerequisite: CSCI 0300, 0330, 1310, or 1330.

Fall CSCI1670 S01 17490 TTh 9:00-10:20(05) (N. DeMarinis)

CSCI 1680. Computer Networks.
Covers the technologies supporting the Internet, from Ethernet and WiFi through the routing protocols that govern the flow of traffic and the web technologies that are generating most of it. A major concern is understanding the protocols used on the Internet: what are the issues, how they work, their shortcomings, and what improvements are on the horizon. Prerequisite: CSCI 0300, 0330, 1310, 1330 or consent of instructor.

Fall CSCI1680 S01 17490 TTh 9:00-10:20(05) (N. DeMarinis)

CSCI 1690. Operating Systems Laboratory
Half-credit course intended to be taken with CSCI 1670. Students individually write a simple operating system in C. Serves to reinforce the concepts learned in 1670 and provides valuable experience in systems programming. Corequisite: CSCI 1670.

Fall CSCI1690 S01 17490 T 9:00-10:20(05) (T. Edgar)

CSCI 1710. Logic for Systems.
The course will focus on proving properties about systems and programs. We will study the distinction between programs and specifications, and check for whether the former obey the latter. We will work with tools that have extensive automation such as model constructers, model checkers, and proof assistants. Problems and projects will apply to real-world systems. Prerequisite: CSCI 0160, CSCI 0180, CSCI 0190, or CSCI 0200. Preferred but not required: CSCI 0220

CSCI 1729. Programming Languages Lab.
Half-credit course intended to be taken concurrently with CSCI 1730. Students individually implement a full programming language chosen by the course. Reinforces the concepts learned in CSCI 1730 and provides valuable experience in implementing programming languages. Corequisite: CSCI 1730

CSCI 1730. Design and Implementation of Programming Languages.
Explores the design principles of modern programming languages through implementation, comparison, and reflection. Examines a variety of linguistic features that impact both control and data. Topics vary by year; more information on the course home page. Prerequisite: CSCI 0160, CSCI 0180 or CSCI 0190.

Fall CSCI1730 S01 17622 MWF 11:00-11:50(16) "To Be Arranged"

CSCI 1760. Multiprocessor Synchronization.
This course examines the theory and practice of multiprocessor synchronization. Subjects covered include multiprocessor architecture, mutual exclusion, wait-free and lock-free synchronization, spin locks, monitors, load balancing, concurrent data structures, and transactional synchronization. Prerequisites: CSCI 0330

Fall CSCI1760 S01 17491 TTh 1:00-2:20(06) (M. Herlihy)

CSCI 1780. Parallel and Distributed Programming.
Covers the practical aspects involved in designing, writing, tuning, and debugging software designed to run on parallel and distributed systems. Topics might include client-server computation, threads, networks of workstations, message passing, shared memory, partitioning strategies, load balancing, algorithms, remote procedure call, and synchronization techniques. Prerequisites: CSCI 0220 and either 0320 or 0360; 0510 recommended.

CSCI 1800. Cybersecurity and International Relations.
The global Internet shortens distances, makes businesses more efficient and facilitates greater social interaction. At the same time, it exposes vital national resources to exploitation and makes it easier for the international criminal element to prey on innocent Internet users. Cybersecurity is concerned with making the Internet a more secure and trustworthy environment. In this course we study this topic from the technological and policy points of view. The goal is to facilitate communication across the divide that normally characterizes the technological and policy communities.

Who is the Big Brother that we most fear? Is it the NSA -- or is it Google and Facebook? Rapidly changing social mores and the growing problem of cybersecurity have all contributed to a sense that privacy is dead. Laws protecting privacy and civil liberties are stuck in the analog age, while the capabilities for mass digital surveillance continue to advance rapidly. This course will examine a variety of informational privacy and technology issues. A major theme: the historical and contemporary struggle to bring surveillance under democratic control to protect against abuses of privacy, civil liberties and human rights.

Fall CSCI1805 S01 17492 TTh 9:00-10:20(11) (T. Edgar)
Fall CSCI1805 S02 17493 Th 9:00-10:20(11) (T. Edgar)

CSCI 1810. Computational Molecular Biology.
High-throughput experimental approaches now allow molecular biologists to make large-scale measurements of DNA, RNA, and protein, the three fundamental molecules of the cell. The resulting datasets are often too large for manual analysis and demand computational techniques. This course introduces algorithms for sequence comparison and alignment; molecular evolution and phylogenetics; DNA/RNA sequencing and assembly; recognition of genes and regulatory elements; and RNA and protein structure. The course demonstrates how to model biological problems in terms of computer science.

Prerequisites: CSCI 0160, CSCI 0180 or CSCI 0190, or consent of instructor.

Fall CSCI1810 S01 17494 TTh 2:30-3:50(12) (S. Istrail)

18 Computer Science
The course is devoted to computational and statistical methods as well as software tools for DNA, RNA, and protein sequence analysis. The focus is on understanding the algorithmic and mathematical foundations of the methods, the design of associated genomics software tools, as well as on their applications. Topics include: sequence alignment, genome assembly, gene prediction, regulatory genomics, and SNP's variation. The course is open to computer and mathematical sciences students as well as biological and medical students.

CSCI 1850. Deep Learning in Genomics.
Deep learning models have achieved impressive performance in fields like computer vision and NLP. The collection of vast quantities of biological data naturally leads to the question -- can deep learning help us understand genomics? We will cover deep learning models like Auto-encoders and Convolutional Neural Networks and how have they been applied to solve problems in genomics. We will learn about different biological datasets, interpretation methods that help explain predictions, and what unique challenges are presented by the data in this field. Critical thinking and learning from the practical application of models to data are expected outcomes.

Course description: Cybersecurity and cyber conflict pose unique legal and policy challenges for governments, companies and citizens. The way those problems are resolved will shape the future of the internet. This course will examine cybersecurity as a legal and policy problem. How can government and society address network and computer insecurity while upholding privacy, civil liberties and other fundamental values?
Fall CSCI1860 S01 17495 Arranged (T. Edgar)

CSCI 1870. Cybersecurity Ethics.
This timely, topical course offers a comprehensive examination of ethical questions in cybersecurity. These issues pervade numerous, diverse aspects of the economy and society in the Information Age, from human rights to international trade. Students will learn about these topics, beginning first with acquaintance with the dominant ethical frameworks of the 20th and 21st centuries, then employing these frameworks to understand, analyze, and develop solutions for leading ethical problems in cybersecurity. The things that you learn in this course will stay with you and inform your personal and professional lives.
Fall CSCI1870 S01 17496 M 3:00-5:30(04) (D. Hurley)
Fall CSCI1870 S02 17497 Arranged(04) (D. Hurley)

CSCI 1880. Introduction to Computer Security.
This course examines the basic principles of computer security for an organization, recognizing which system components relate to which principles. Additionally, the course covers methodologies and skills for making informed security decisions and understanding how to apply security principles to design security mechanisms while considering tradeoffs. Topics include general security principles, cryptography, authentication authorization, identity, and access management, operating systems security, network security, web security, and applications security. Throughout the course, you will develop a preliminary cybersecurity plan for an organization.

CSCI 1900. csciStartup.
In csciStartup, you will incorporate and run a startup. Apply as a team to be part of a prototype class to remove the mystery from starting a company and to focus entirely on a product you're passionate about. Teams will incorporate, build a product for real customers, advertise their product, and improve it week after week. We'll spend half our class meetings with individual attention to each group's progress and how to improve your offerings. Assignments will be designed to apply to any company, with enough flexibility to ensure you're always working on things that make sense for your business.

CSCI 1950B. Computational Topology and Discrete Geometry.
This course will investigate (through a mixture of lectures and student presentations of recent papers) topics in computational topology, including Morse theory and discrete differential geometry. Other possible topics are knot polynomials, simplicial homology, and geometric probability theory. Some mathematical sophistication and programming skills required. No prerequisites.

We will study various algorithmic problems that arise in the study of topological phenomena, such as winding number, turning number, knot polynomials, topology of covering spaces (especially Riemann surfaces), and discrete Morse theory. The mathematical topics will be briefly introduced before we move to computations, but some a priori mathematical sophistication will make the course more valuable to the student. Prerequisite: CSCI 0160, 0180, or 0190.

CSCI 1950J. Introduction to Computational Geometry.
Geometric algorithms in two and three dimensions. Algorithmic and geometric fundamentals. Point location, convex hulls, proximity (Voronoi diagrams, Delaunay triangulations), intersections, the geometry of rectangles. Prerequisites: CSCI 0160, 0170, or 0190; and CSCI 0220.

CSCI 1950K. Innovating Game Development.
A project-centered course focused on technological, paradigm, and design innovations for game development. As teams, students will propose and implement a project demonstrating a novel technology for gaming. Examines the current state and future of game development through a seminar of speakers active in game development and research. A strong computer science or engineering background is recommended.

CSCI 1950M. Advanced Practical Combinatorial Algorithms.
We review recent as well as well-established advanced techniques in combinatorial optimization and constraint satisfaction. Students will study and individually present research papers and work on challenging software projects in small teams. Prerequisites: CSCI 0160, 0180, or 0190; and CSCI 0510; and CSCI 1490 or 2580, or instructor permission.

CSCI 1950N. 2D Game Engines.
2D Game Engines covers core techniques used in the development of the software that drives computer games and other interactive software. Projects involve building different varieties of 2D game engines as well as games that require use of the features implemented in the engines. Topics include high-level engine design, vector and raster graphics, animation, collision detection, physics, content management, and game AI. Prerequisite: CSCI 0160, 0180, or 0190. This course has also been offered as DISP CSCI1971. Students interested in an override should request on through Courses@Brown. Priority will be given to both seniors and juniors.
Fall CSCI1950N S01 17498 TTh 6:40-8:00PM(02) (J. Tompkin)

CSCI 1950R. Compiler Practice.
This class covers the practice of compiler writing, including lexical analysis, parsing, semantic analysis, code generation, and code optimization. Students design and implement a full compiler modularly for a modern functional language using a modern intermediate representation and modular backend. Instructor permission required.

CSCI 1950T. Advanced Animation Production.
Students will apply knowledge and skills gained in previous animation courses to produce a high quality short animated film as a group. Production will follow the industry standard pipeline that includes modeling, texturing, lighting, animating, rendering, and post production. Interested students will perform preproduction story and concept design prior to beginning of course. Prerequisite: CSCI 1250. Enrollment limited to 15. Instructor permission required.
Fall CSCI1950T S01 17579 M 3:00-5:30(03) (B. Meier)
CSCI 1950U. Topics in 3D Game Engine Development. Covers core techniques in 3D game development with an emphasis on engine architecture. Students independently develop their own engines using C++, OpenGL, and the QF framework, then work in groups to create a polished game. Topics include: spatial subdivision, player representation, collision detection and response, game networking, GPUs, and OpenGL. Prerequisites: CSCI 1230 or knowledge of C++ and one of CSCI0300, CSCI0320, or CSCI0330 or equivalent experience. Enrollment limited to 25.

CSCI 1950X. Software Foundations. Software Foundations will be a project-based course focusing on the challenges and techniques involved in proving non-trivial properties about real-world systems. We will base our exploration around formal development in a proof environment. Roughly half of the course will be a guided tutorial of proof techniques using one or more theorem provers; in the remainder, students will apply this knowledge to existing systems. No prior experience with theorem provers or proof assistants is necessary, but familiarity with and aptitude for functional programming will be a huge bonus. Prerequisite: CSCI 1730 or equivalent; mathematical maturity.

CSCI 1950Z. Computational Methods for Biology. This course will introduce algorithms from machine learning and combinatorial optimization with a focus on their application to biological data. Topics will include problems in phylogenetic inference, population genetics, and biological interaction networks.

CSCI 1951A. Data Science. Mastering big data requires skills spanning a variety of disciplines: distributed systems over statistics, machine learning, and a deep understanding of a complex ecosystem of tools and platforms. Data Science refers to the intersection of these skills and how to transform data into actionable knowledge. This course provides an overview of techniques and tools involved and how they work together: SQL and NoSQL solutions for massive data management, basic algorithms for data mining and machine learning, information retrieval techniques, and visualization methods. Prerequisites: CSCI 0160, CSCI 0180, CSCI 0190, or CSCI 0200. One of CSCI 0300, 0330, CSCI 0320, 1310 or 1330 strongly recommended.

CSCI 1951B. Virtual Citizens or Subjects? The Global Battle Over Governing Your Internet. The Internet began as a U.S. government research project, progressed to an open network run by free-spirited geeks, and transitioned in the late 1990’s to a unique governance model in which nations, corporations, and civil society were supposed to all have a voice. Where are the real decisions being made? Who is making them? How can you and citizens of other nations influence these decisions? The global battle to run the Internet, brewing for years, has broken wide open with revelations of American spying on a massive scale.

CSCI 1951C. Designing Humanity Centered Technology. This semester we will explore how emerging technologies might shape our lives in the near future, as we design and build working prototypes. We will proceed from a set of questions that will complement a deep understanding of a complex ecosystem of tools and platforms. Data Science refers to the intersection of these skills and how to transform data into actionable knowledge. This course provides an overview of techniques and tools involved and how they work together: SQL and NoSQL solutions for massive data management, basic algorithms for data mining and machine learning, information retrieval techniques, and visualization methods. Prerequisites: CSCI 0160, CSCI 0180, CSCI 0190, or CSCI 0200. One of CSCI 0300, 0330, CSCI 0320, 1310 or 1330 strongly recommended.

CSCI 1951D. Design of Robotic Systems (ENGN 1931I). Interested students must register for ENGN 1931I.

CSCI 19518. Introduction to Robotics. Each student will learn to program a small quad-rotor helicopter. We will provide each student with their own robot for the duration of the course. The course will cover PID controllers for stable flight, localization with a camera, mapping, and autonomous planning. At the end of the course, the aim is for students to understand the basic concepts of a mobile robot and aerial vehicle. Enrollment by instructor permission.

CSCI 1951G. Optimization Methods in Finance. Optimization plays an important role in financial decisions. Many computational finance problems ranging from asset allocation to risk management, from option pricing to model calibration can be solved efficiently using modern optimization techniques. This course discusses several classes of optimization problems (including linear, quadratic, integer, dynamic, stochastic, conic, and robust programming) encountered in financial models. For each problem class, after introducing the relevant theory and efficient solution methods, we discuss problems of mathematical finance that can be modeled within this problem class. Prerequisites: CSCI 1450 or APMA 1650, and CSCI 1570.

CSCI 1951I. CS for Social Change. Working in a studio environment to iteratively design, build, and test technical projects in partnership with different social change organizations, students will be placed in small teams to collaboratively work on projects that will range from developing a chatbot to aid community engagement to conducting geospatial data analytics. We will also reflect on our positionality and ethics in engaging in social impact work and what it practically means to leverage technology to create social change on an everyday basis.

CSCI 1951J. Interdisciplinary Scientific Visualization. Students will learn about solving scientific problems using computer graphics and visualization. Projects will involve the solution of scientific problems using computer graphics, modeling, and visualization. Working in small groups, students will identify scientific problems, propose solutions involving computational modeling and visualization, evaluate the proposals, design and implement the solutions, apply them to the problems, evaluate their success, and report on results. Example projects might include interactive software systems, immersive virtual reality cave applications, quantitative analysis tools, or new applications of existing visualizations methods. The focus will be on applications in the new virtual reality cave.

CSCI 1951L. Blockchains and Cryptocurrencies. Introduction to modern blockchain-based systems. Topics covered include consensus and distributed computing, examples cryptocurrencies, programming smart contracts, privacy and secrecy, transfer networks, atomic swaps and transactions, new-currency applications of blockchains, and legal and social implications. Students will do a programming project and a term project.

CSCI 1951N. VR+X, The Potential of Virtual Reality to Transform Nearly Everything. This course introduces students to the history, present, and future possibilities of virtual reality (VR) with a focus on addressing the question: What is the transformative potential of virtual reality? We’ll critically evaluate a variety of applications in fields as varied as healthcare, architecture, education, and storytelling. Students will learn discovery and design thinking processes of a kind that can lead to the development of VR solutions. Students will create a design concept for a VR use case in a field of their choosing.

CSCI 1951P. Surveying VR Data Visualization Software for Research. In a collaborative group effort, this course will search out, install, test, and critically evaluate VR software that supports data visualization for researchers. We will target several specific types of data, including volumetric data, and remote sensing data. We will investigate the capabilities of software for head-mounted displays (HMDs), big-metal displays like caves and the yurt, and, as a baseline, desktop displays. Software evaluation will include web research, hands-on case studies, and surveys. Results will be documented in a courses wiki.
CSCI 1951V. Hypertext/Hypermedia: The Web Was Not the Beginning and the Web Is Not the End.

Hypertext/Hypermedia systems -- first designed in the 1960s -- link information and people. Developed in the late 1980s, the Web was the first global hypermedia system; 30+ years later, it represents a small part of past visions. Students will identify still-uncommon features by exploring/using systems from the 1960s onwards. They will read papers for class discussion. They will study architecture and design topics such as annotating, note taking, searching, networking, collaboration, permanence, and social impact. Web programming projects, using TypeScript/MERN stack, will culminate in group projects to create their own hypertext/hypermedia systems. Prerequisites: An introductory CS sequence or equivalent experience

Fall CSCI1951V S01 17540 W 3:00-5:30(10) (N. Meyrowitz)


A huge quantity of data is worth little unless we can extract insights from it. Yet, the large quantities mean that classic algorithms (running in linear, quadratic or even more time) can be infeasible even when the data set is Facebook-sized. Surprisingly, to answer many computational and statistical questions, sometimes there is no need to read/store every piece of data! This course focuses on this exciting "sublinear" algorithmic regime. We will study practical algorithms, making clever use of randomness with strong theoretical guarantees

Prerequisites: (CS22 or equivalent); (CS145 or APMA1650/1655 or equivalent); (CS157 or CS155). Mathematical maturity is essential: this is a theory course with proofs.

Recommended: CS155

CSCI 1951X. Formal Proof and Verification.

Proof assistants are tools that are used to check the correctness of programs. Unlike tools like model checkers and SAT solvers, proof assistants are highly interactive. Machine-checked formal proofs lead to trustworthy programs and fully specified reliable mathematics. This course introduces students to the theory and use of proof assistants, using the system Lean. We will use Lean to verify properties of functional programs and theorems from pure mathematics. We will learn the theory of deductive reasoning and the logic that these tools are based on.

Text: "The Hitchhiker's Guide to Logical Verification" by Blanchette et al.

Prereq: CSCI 1710 Logic for Systems or a proof-based mathematics course. Basic familiarity with functional programming (e.g. Haskell, ML) is helpful but not required.

Fall CSCI1951X S01 17501 MW 3:00-4:20(10) (R. Lewis)

CSCI 1952B. Responsible Computer Science in Practice.

What can ethics and social and political theory tell us about how to navigate the social impacts of computing? How do these perspectives shape technical decisions computer scientists have to make? The role of computer scientists is rapidly evolving: as the systems they build affect everyone, from individuals to society at large, computer scientists become more than just coders. They must be able to assess the social impacts of the technology they develop and engage with experts from other disciplines which offer critical insights and normative perspectives on those impacts. The goal of this course is to enable you to understand and critically reflect on key concepts and ideas in ethics and social and political theory on topics ranging from fairness to consent, digital well-being to regulation, and to apply them to concrete technical decisions in practical exercises and project-oriented work.

CSCI 1952L. Language Processing in Humans and Machines (CLPS 1850).

Interested students must register for CLPS 1850.

CSCI 1952V. Algorithms for the People.

Computer science has transformed every aspect of society, including communication, transportation, commerce, finance, and health. The revolution enabled by computing has been extraordinarily valuable. The largest tech companies generate almost a trillion dollars a year and employ millions of people. But technology does not affect everyone in the same way. In this seminar, we will examine how new technologies, ranging from facial recognition to drones, are affecting marginalized communities.

CSCI 1952X. Contemporary Digital Policy and Politics.

This course will examine the politics and processes for making policies related to the internet and digital policy issues. We will examine current issues at the national level, including the White House and federal agencies, Congress, international institutions and industry on issues such as privacy and information security, and on debates like whether and how to regulate Big Tech. Topics covered include the creation of national policies at the White House, the regulatory process, legislation, standards, global implications and the politics of technological change.

Format and participation: This is an asynchronous version of IAPA 1811, available only to students enrolled in a completely online master's degree program, by permission of the instructor. Students will complete weekly activities in lieu of attending synchronous class discussions.


Independent study in various branches of Computer Science. Section numbers vary by instructor. Please check Banner for the correct section number and CRN to use when registering for this course.

CSCI 1971. Independent Study in 2D Game Engines.

2D Game Engines covers core techniques used in the development 2D game engines. Projects involve building different varieties of 2D game engines as well as games that require use of the features implemented in the engines. Topics include high-level engine design, vector and raster graphics, animation, collision detection, physics, content management, and game AI. Prerequisite: CSCI 0160, 0180, or 0190.

CSCI 1972. Topics in 3D Game Engine Development.

Covers core techniques in 3D game development with an emphasis on engine architecture. Students independently develop their own engines using C++, OpenGL and the Qt framework, then work in groups to create a polished game. Topics include: spatial subdivision, player representation, collision detection and response, game networking, GPUs, and OpenGL.

Prerequisite: CSCI 1230 and one of the following CSCI 0320, CSCI 0330, CSCI 1950N, OR CSCI 1971.


What does it mean to conduct research in computer science, and how might we be most effective at it? To help begin a fruitful career in research, this class will cover the philosophy and practice of forming ideas, executing research, presenting outcomes, and understanding and contributing to our community. The aim is to kick-start your time at Brown CS by being the 'missing semester' on how to be a PhD student, and by peeling back the curtain on why CS academia works like this to help you make the most of your time. Discussions include: motivating, pitching, and funding research; finding, reading, and reviewing research; selecting research areas and forming hypotheses; designing, performing, and evaluating research; communicating research; research collaborations; and research ethics. We will learn together through presentations, activities, discussions, plus readings and assignments out of class.


If you tried to live for one day without generating any digital personal data, how would you spend it? In the Information Age, the use of personal data has proliferated and is pervasive. This course offers a comprehensive examination of protection of privacy and personal data, which is central to autonomy, dignity, and liberty. Topics include identity, financial, health, educational, and other data. Students will learn about: Fair Information Practices; the development of modern privacy rules in the United States and around the world; Fourth Amendment privacy and the autonomy of the individual in relation to the state; key US laws (HIPAA, FERPA, GLBA, GINA, COPPA, etc.); significant international rules (European Union’s General Data Protection Regulation (GDPR), etc.); important institutions (Federal Trade Commission, Data Protection Authorities, etc.); standards; Privacy by Design and Default; and emerging issues.
CSCI 2240. Advanced Computer Graphics. CSCI 2240 explores several key areas of 3D graphics—rendering, geometry processing, optimization, and simulation—taking a sophisticated approach to each. This year, we are looking to improve the course’s coverage of optimization by adding more lecture content on the topic (optimization theory, methods for solving (sparse) linear systems, etc.) and by designing a new assignment (likely 3D as-rigid-as-possible shape manipulation). Prerequisites: one of CSCI 0530, MATH 0520, MATH 0540; CSCI 1230; and familiarity with multivariable calculus by e.g. having taken one of MATH 180, MATH 200, MATH 350

CSCI 2270. Topics in Database Management. In-depth treatment of advanced issues in database management systems. Topics vary from year to year and may include distributed databases, mobile data management, data stream processing and web-based data management. Prerequisite: CSCI 1270.

CSCI 2300. Human-Computer Interaction Seminar. Covers methods for conducting research in human-computer interaction (HCI). Topics will be pursued through independent reading, assignments, and class discussion. Comprises four assignments that apply to HCI research methods and push the envelope, which are designed to be meaningful and have the potential for real impact. Students will gain the background necessary to perform research in HCI and the skills to conduct human-centric research. There will be little content about user interfaces, but students may find some topics in CSCI 1300 relevant. Please see the course website when it's available (shortly before the semester begins) for information about overrides.

CSCI 2310. Human Factors and User Interface Design. Covers current research issues involving the implementation, evaluation and design of user interfaces, while also providing a basic background in the fundamentals of user interface evaluation, programming, tools, and techniques. A possible topic is programming and designing device-independent interfaces. Previous topics have included the development of pervasive internet-based interfaces and software visualization. Prerequisite: Consent of instructor.

CSCI 2330. Programming Environments. Programming tools; control and data integration; software understanding and debugging; environments for parallel and distributed programming; reverse engineering; configuration management and version control and debugging. Emphasis on current research areas. Prerequisite: consent of instructor.

CSCI 2340. Software Engineering. Topics in the design, specification, construction and validation of programs. Focus will be on tools to support each of these stages. Course will pay special attention to the concerns raised by the properties of modern software systems including distribution, security, component-based decomposition and implicit control. Prerequisites: CSCI 0320 or CSCI 0330

CSCI 2370. Interdisciplinary Scientific Visualization. Learn how to do research on using computer graphics, visualization, and interaction applied to scientific problems. Working in small multidisciplinary groups, students identify scientific problems, propose solutions involving computational modeling and visualization, design and implement the solutions, apply them to the problems, and evaluate their success. Prerequisites: programming experience, some graphics experience, problem ideas.

CSCI 2390. Privacy-Conscious Computer Systems. We will examine research papers on distributed system design, privacy-preserving, and secure computing techniques, and discuss how to apply these ideas in practice. The goal is to understand if, and how we can better protect the sensitive data we entrust to computer systems, both against leaks and against unauthorized or unethical use. We will look at web services, datacenter systems, distributed communication systems, and machine learning systems. During class, you will present and discuss papers, finish a set of hands-on assignments, work on a research project, and present your project at the end of the semester.

CSCI 2400. Probabilistic Graphical Models. Probabilistic graphical models provide a flexible framework for modeling large, complex, heterogeneous collections of random variables. After a brief introduction to their representational power, we provide a comprehensive survey of state-of-the-art methods for statistical learning and inference in graphical models. We discuss a range of efficient algorithms for approximate inference, including optimization-based variational methods, and simulation-based Monte Carlo methods. Several approaches to learning from data are explored, including conditional models for discriminative learning, and Bayesian methods for controlling model complexity. Programming experience required for homeworks and projects, which integrate mathematical derivations with algorithm implementations. PREREQUISITES: CSCI1420 or APMA1690.

CSCI 2430. Topics in Machine Learning. Machine learning from the artificial intelligence perspective, with emphasis on empirical validation of learning algorithms. Different learning problems are considered, including concept learning, clustering, speed-up learning, and behavior learning. For each problem a variety of solutions are investigated, including those from symbolic AI, neural and genetic algorithms, and standard statistical methods. Prerequisite: CSCI 1410 or familiarity with basic logic and probability theory.

CSCI 2440. Advanced Algorithmic Game Theory. This course examines topics in game theory and mechanism design from a computer scientist’s perspective. Through the lens of computation, the focus is the design and analysis of systems utilized by self-interested agents. Students will investigate how the potential for strategic agent behavior can/should influence system design, and the ramifications of conflicts of interest between system designers and participating agents. Emphasis on computational tractability is paramount, so that simple designs are often preferred to optimal. Students will learn to analyze competing designs using the tools of theoretical computer science, and empirical tools, such as empirical game-theoretic analysis. Application areas include computational advertising, wireless spectrum, and prediction markets.

CSCI 2450. Exchange Scholar Program. Fall CSCI2450 S01 16098 Arranged 'To Be Arranged'

CSCI 2470. Deep Learning. Deep Learning belongs to a broader family of machine learning methods. It is a particular version of artificial neural networks that emphasizes learning representation with multiple layers of networks. Deep Learning, plus the specialized techniques that it has inspired (e.g. convolutional neural networks, recurrent neural networks, and transformers), have led to rapid improvements in many applications, such as computer vision, machine learning, sound understanding, and robotics. This course gives students an overview of the prominent techniques of Deep Learning and its applications in computer vision, language understanding, and other areas. It also provides hands-on practice of implementing deep learning algorithms in Python. A final project will implement an advanced piece of work in one of these areas. Prerequisites: basic programming; (CSCI 0150, 0170, 0190); linear algebra. (CSCI 0530, MATH 0520, 0540); stats/probability: (CSCI 0220, 1450, 0450, MATH 1610, APMA 1650, 1655)

CSCI 2500A. Advanced Algorithms. In this course, we study a selection of advanced algorithms and data structures that are provably correct and fast. Our goal is to present a broad range of algorithmic ideas and techniques, especially those that have had significant impact on the field and/or have had or might have practical impact. Prerequisite: CSCI 1570 or the equivalent
CSCI 2500B. Optimization Algorithms for Planar Graphs.
Planar graphs arise in applications such as road map navigation and logistics, graph drawing and image processing. We will study graph algorithms and data structures that exploit planarity. Our focus will be on recent research results in optimization. Prerequisite: CSCI 1570 or the equivalent.

CSCI 2510. Approximation Algorithms.
Approximation Algorithms deal with NP-hard combinatorial optimization problems by efficiently constructing a suboptimal solution with some specified quality guarantees. We study techniques such as linear programming and semidefinite programming relaxations, and apply them to problems such as facility location, scheduling, bin packing, maximum satisfiability or vertex cover. Prerequisite - one of the following: CSCI 1510, 1550, 1810, 1950J, 1950L, any graduate-level course on algorithms (including 2500A, 2500B, 2580).

CSCI 2520. Computational Geometry.
Algorithms and data structures for fundamental geometric problems in two and three dimensions. Topics include point location, range searching, convex hull, intersection, Voronoi diagrams, and graph drawing. Applications to computer graphics, circuit layout, information visualization, and computer-aided design are also discussed. Prerequisite: CSCI 1570 or instructor permission.

CSCI 2530. Design and Analysis of Communication Networks.
A theory seminar focusing on algorithmic and combinatorial issues related to the design and analysis of communication networks for parallel and distributed systems. Topics include packet routing, circuit switching, distributed shared memory, fault tolerance, and more. Prerequisites: CSCI 1550, 1570, or equivalent.

CSCI 2531. Internet and Web Algorithms.
This advanced graduate course/seminar focuses on the mathematical foundations of algorithms for handling large amounts of data over networks. We'll read and discuss recent papers in information retrieval, search engines, link analysis, probabilistic modeling of the web and social networks, and more. Recommended: CSCI 1550 and CSCI 1570, or equivalent courses.

Advanced topics in applications of probabilistic methods in design and analysis of algorithms, in particular to randomized algorithms and probabilistic analysis of algorithms. Topics include the Markov chains Monte Carlo method, martingales, entropy as a measure for information and randomness, and more. Prerequisite: CSCI 1450. Recommended but not required: CSCI 1570.

The theoretical foundations of parallel algorithmics. Analysis of the most important models of parallel computation, such as directed-acyclic computation graphs, shared memory and networks, and standard data-exchange schemes (common address space and message-passing). Algorithmic techniques with numerous examples are cast mostly in the data-parallel framework. Finally, limitations to parallelizability (P-completeness) are analyzed. The content of the course is likely to change as technology evolves.

CSCI 2560. Advanced Complexity.
Advanced topics in computational complexity, such as: the polynomial hierarchy, interactive proofs, pseudorandomness, derandomization, probabilistically checkable proofs.

CSCI 2570. Introduction to Nanocomputing.
Nanoscale technologies employing materials whose smallest dimension is on the order of a few nanometers are expected to replace lithography in the design of chips. We give an introduction to computational nanotechnologies and explore problems presented by their stochastic nature. Nanotechnologies based on the use of DNA and semiconducting materials will be explored. Prerequisite: CSCI 0510.

The theory of combinatorial optimization and how it is embodied in practical systems. Explores issues encountered in implementing such systems. Emphasizes the wide variety of techniques and methodologies available, including integer programming, local search, constraint programming, and approximation algorithms. Problems addressed may include: scheduling, coloring, traveling salesman tours, and resource allocation. Prerequisites: CSCI 0320 and basic knowledge of linear algebra.

CSCI 2590. Advanced Topics in Cryptography.
Seminar-style course on advanced topics in cryptography. Example topics are zero-knowledge proofs, multi-party computation, extractors in cryptography, universal compositability, anonymous credentials and ecash, interplay of cryptography and game theory. May be repeated for credit. Prerequisite: CSCI 1510 or permission of the instructor.

Fall CSCI2590 S01 17585 W 3:00-5:30(10) (A. Lysyanskaya)

This course teaches computer security principles from an applied viewpoint and provides means on how to protect our systems against threats and countermeasures. The course additionally covers principles and skills useful for making informed security decisions and for understanding how security interacts with the world around it. The main topics covered are cryptography, authentication, access control, web security, and network security. Other topics include cybersecurity ethics and privacy. The course aims to balance Theory and Practice. These advanced versions focus on real-world skills: performing attacks that are more difficult and rely on less serious vulnerabilities, and creating attacks that achieve a higher standard of quality than a mere "proof of concept." This course covers the same material as CSCI 1620 and 1660 and shares their assignments. Graduate students only. If you are interested in this course, request an override and fill out this form: https://forms.gle/pHPAy9ntQkAQ7xLD9

Covers not just the principles of operating systems but the intricacies of how they work. Topics include multithreaded programming, managing threads and interrupts, managing storage, processor scheduling, operating-system structure, virtualization, security, and the design of file systems (both local and distributed). Extensive examples are taken from actual systems, including Linux and Windows. Students are expected to complete both problem sets and programming assignments (in C) and will individually write a simple operating system. Prerequisite: one of CSCI 0300, CSCI 0330, CSCI 1310, or CSCI 1330. Graduate students only. This course covers the same material as the combination of CSCI 1670 and 1690 and shares their assignments.

CSCI 2730. Programming Language Theory.
Theoretical models for the semantics of programming languages and the verification of programs. Topics will be drawn from operational semantics, denotational semantics, type theory and static analyses. Recommended prerequisite: CSCI 1730, CSCI 1950Y or instructor permission.

CSCI 2750. Topics in Parallel and Distributed Computing.
CSCI 2750 is a graduate seminar that will consider an advanced topic (to be determined) in distributed computing. May be repeated for credit.

CSCI 2810. Advanced Computational Molecular Biology.
High-throughput experimental approaches now allow molecular biologists to study large-scale measurements of, for example, gene expression, the three fundamental molecules of the cell. The resulting datasets are often too large for manual analysis and demand computational techniques. This course introduces algorithms for sequence comparison and alignment; molecular evolution and phylogenetics; DNA/RNA sequencing and assembly; recognition of genes and regulatory elements; and RNA and protein structure. The course demonstrates how to model biological problems in terms of computer science. CSCI 0160, 0180, 0190, or 0200. Recommended: CS 220, or some other course that introduces concepts from discrete math and probability. Course overrides are available at the instructor’s discretion.

Fall CSCI2810 S01 17504 TTh 2:30-3:50(12) (S. Istrail)
CSCI 2840. Advanced Algorithms in Computational Biology and Medical Bioinformatics.
Devoted to computational problems and methods in the emerging field of Medical Bioinformatics where genomics, computational biology and bioinformatics impact medical research. We will present challenging problems and solutions in three areas: Disease Associations, Protein Folding and Immunogenomics. This course is open to graduate students and advanced undergraduates with Computational or Life Science backgrounds. Prior background in Biology is not required.

CSCI 2890. Comprehensive Examination Preparation.
For graduate students who have met the tuition requirement and are paying the registration fee to continue active enrollment while preparing for a preliminary examination.

Fall CSCI2890 S01 16099 Arranged  "To Be Arranged"
Spr CSCI2890 S01 24852 Arranged  "To Be Arranged"

CSCI 2950C. Topics in Computational Biology.
This course will investigate active and emerging research areas in computational biology. Topics include cancer genomics; genome rearrangements and assembly; and protein and regulatory interaction networks. The course will be a mixture of lectures and student presentations of recent conference and journal papers.

CSCI 2950D. Sensor Data Management.
Sensor networks combine sensing, computing, actuation, and communication in a single infrastructure that allows us to observe and respond to phenomena in the physical and cyber world. The sensors range from tiny "smart dusts" to dime-sized RFID tags and large-scale weather sensors. This course will cover the state-of-the-art in designing and building sensor networks, focusing on issues that revolve around data and resource management. No prerequisites.

CSCI 2950E. Stochastic Optimization.
This advanced graduate course/seminar will focus on optimization under uncertainty, or optimization problems where some of the constrains include random (stochastic) components. Most practical optimization problems are stochastic (subject to future market conditions, weather, faults, etc.), and there has been substantial research (both theoretical and experimental) in efficient solution for such problems. We'll read and discuss some of the recent works in this area.

CSCI 2950F. Implementing Web-Based Software Systems.
Explores widely-distributed systems that take advantage of resources throughout the Internet. The systems leverage their large size and geographic diversity to provide bandwidth scalability, rapid responses, fault-tolerance, high-availability and diverse data collection. Topics include overlay networks, peer-to-peer systems, content distribution networks, distributed file systems and wide-scale measurement systems.

CSCI 2950H. Advanced Cryptography.
CSCI 2950. Computational Models of the Neocortex.
This course addresses the problem of modeling the perceptual neocortex using probabilistic graphical models, including Bayesian and Markov networks, and extensions to model time and change such as hidden Markov models and dynamic Bayesian networks. The emphasis is on problems of learning, inference, and attention. Sources include the literature in computational and cognitive neuroscience, machine learning, and other fields that bear on how biological and engineered systems make sense of the world. Prerequisites: basic probability theory, algorithms and statistics.

In this graduate seminar we will learn about models of human cognition and perception, and explore potential implications of the models on how computers and humans can interact effectively when performing scientific analyses. Participants will be responsible for reading assigned materials, taking turns guiding discussions of the readings, and preparing a final paper and presentation. It is recommended that participants have some background in at least one of the areas of study.

CSCI 2950K. Special Topics in Computational Linguistics.
Every year will cover a different topic in computational linguistics, from a statistical point of view, including parsing, machine translation, conference, summarization, etc. Prerequisites: CSCI 1460 or permission of the instructor.

CSCI 2950M. Computer Science, Algorithms and Economics.
Course investigates the interplay of economic theory and computer science. It is suitable for advanced senior undergraduates and for graduate students. We will study topics such as: algorithms for selfish routing; competitive combinatorial auctions; multicast cost sharing and cooperative games; graphical models for games; and related topics. This course will be organized around the presentation of recent research papers. Prerequisite: CSCI 1570 or equivalent.

CSCI 2950N. Special Topics in Autonomous Robotics.
No description available.

CSCI 2950O. Special Topics in Brain-Computer Interfaces.
Introduces the mathematical and computational foundations of brain-computer interfaces. Statistical learning, Bayesian inference, dimensionality reduction, information theory, and other topics are presented in the context of brain interfaces based on neural implants and EEG recordings. Prerequisites: Basic knowledge of probability, statistics and linear algebra (e.g., CSCI 1550, APMA 1650, APMA 1690, or APMA 2640). Enrollment limited to 20 students.

CSCI 2950P. Special Topics in Machine Learning.
This seminar course explores current research topics in statistical machine learning. Focus varies by year, and may include Bayesian nonparametrics; models for spatial, temporal, or structured data; and variational or Monte Carlo approximations. Course meetings combine lectures with presentation and discussion of classical and contemporary research papers. Students will apply some this material to a project, ideally drawn from their own research interests.

CSCI 2950Q. Topics in Computer Vision.
This course will cover current topics in computer vision by focusing on a single real problem in computer vision. Recent courses have focused on forensic video analysis of an unsolved murder and three-dimensional object recognition for a mobile robot. Readings from the literature are integrated with group projects to solve problems beyond the state of the art. Strong mathematical skills (probability, linear algebra, calculus) and previous exposure to computer vision (e.g. CSCI 1430) are essential.

CSCI 2950R. Special Topics in Advanced Algorithms.
We will study an advanced topic in the design and analysis of algorithms. Prerequisite: CSCI 1570 or the equivalent.

CSCI 2950S. Advanced Practical Combinatorial Algorithms.
We review recent as well as well-established advanced techniques in combinatorial optimization and constraint satisfaction. Students will study and individually present research papers and work on challenging software projects in small teams. Prerequisites: CSCI 0160, 0180, or 0190; and CSCI 0510; and CSCI 1490 or 2580, or instructor permission.

CSCI 2950T. Topics in Distributed Databases and Systems.
This course explores data and resource management issues that arise in the design, implementation, and deployment of distributed computing systems by covering the state of the art in research and industry. Typical topics include cloud computing and sensor networks. Strongly recommended: CSCI 0320, CSCI 1270, or CSCI 1951A.

CSCI 2950U. Special Topics on Networking and Distributed Systems.
Explores current research topics in networking, distributed and operating systems. Specific topics may include wireless and sensor networking, Internet-scale distributed systems, cloud computing, as well as the core problems, concepts, and techniques underlying these systems. The course has two components: reading and discussion of current and classical research papers, and a research project related to the topic but ideally drawn from students' own research interests. This is a graduate-level course, undergrads can join with the consent of the instructor.
CSCI 2950V. Topics in Applied Cryptography.
This course surveys recent developments in applied cryptography. Research in this field is motivated by privacy and security issues that arise in practice from areas like cloud computing, databases, surveillance and finance. Topics will vary each year. Pre Requisites: CSCI 1660 and CSCI 1510 recommended or instructor permission. This year’s theme is cryptography for social good.

CSCI 2950W. Online Algorithms.
Decisions must often be made before the entire data is available. Online algorithms solve problems in which commitments must be made as the data is arriving. Choosing which items to evict from a cache before knowing future requests, which advertisers to consider for displaying ads alongside the result of a search, or which most representative data to store when computing statistics about a huge stream of information. We will discuss the worst-case model, which hinges against the worst possible future data, and some stochastic and game-theoretic models.

CSCI 2950X. Topics in Programming Languages and Systems.
Examines contemporary research topics in software construction from the perspectives of programming languages, software engineering and computer-aided verification. The primary goals are to understand which theory applies to which problems and to convert that theory into tools. Topics include security, modularity, and new paradigms in software composition. Prerequisite: CSCI 1730 or written permission of the instructor.

CSCI 2950Y. Theorem Proving.
This course explores computer-assisted theorem proving with the Coq Proof Assistant. The course will teach students to formally specify software and model mathematical theories. We will then study techniques for mechanically proving theorems about these Coq. Prerequisites: CSCI 1730 or CSCI 0170 and permission of the instructor.

CSCI 2950Z. Robot Learning and Autonomy.
This seminar course will cover current research topics related to perceiving and acting in the real world. These topics will be pursued through independent reading, class discussion, and project implementations. Papers covered will be drawn from robotics, computer vision, animation, machine learning, and neuroscience. Special emphasis will be given to developing autonomous control from human performance. No prerequisites.

CSCI 2951A. Robots for Education.
This seminar will explore the potential for robotics to engage future generations of scientists and engineers, with a particular focus on broadening participation in computing across society. Academic papers describing existing models, systems, courses, and evaluation for teaching robotics at undergraduate and secondary levels will be covered through students presentations. A group project will be conducted to find viable and accessible “off-the-shelf” technology solutions suited to teaching robotics without requiring a technical background. Instructor permission required.

CSCI 2951B. Data-Driven Vision and Graphics.
Investigates current research topics in image-based graphics and vision. We will examine data sources, features, and algorithms for understanding and manipulating visual data. We will pay special attention to methods that use crowd-sourcing or Internet-derived data. Vision topics such as scene understanding and object detection will be linked to graphics applications such as photo editing and image-based rendering. These topics will be pursued through independent reading, class discussion and presentations, and a semester long research project. Strong mathematical skills and previous imaging (vision or computational photography) courses are essential.

CSCI 2951C. Autonomous Agents and Computational Market Design.
An important area of research in artificial intelligence is how to effectively automate decision making in time-critical, information-rich environments. Electronic markets are a prime example of such environments. In this course students will create their own simulated electronic market as well as autonomous agents that trade in their market simulation. Application domains will include supply chain management, the Dutch flower auctions, and ad auctions, such as those run by Google and Facebook. Enrollment limited to 40 graduate students.

CSCI 2951E. Topics in Computer Systems Security.
This course explores advanced topics and highlights current research in computer security and privacy. Recent research papers will be presented and discussed. Also, projects will provide an opportunity for creative work. Class attendance is required and active participation in class discussions is essential. The course has two sections, each with a different focus and prerequisites. Section S01 (Networks, Software, and Systems) addresses computer security and privacy from the perspective of networks, software, and systems. Section S02 (Human Factors, Law, and Policy) addresses computer security and privacy from the perspective of law, policy, and human factors. Either section of the course can be used toward satisfying the capstone requirement for the ScB degree in Computer Science. Instructor permission is required to register.

CSCI 2951F. Learning and Sequential Decision Making.
The course explores automated decision making from a computer-science perspective. It examines efficient algorithms, where they exist, for single agent and multiagent planning as well as approaches to learning near-optimal decisions from experience. Topics will include Markov decision processes, stochastic and repeated games, partially observable Markov decision processes, and reinforcement learning. Of particular interest will be issues of generalization, exploration, and representation. Participants should have taken a graduate-level computer science course and should have some exposure to machine learning from a previous computer science class or seminar; check with instructor if not sure. Recommended Prerequisites: CSCI 1950F or CSCI 1420

CSCI 2951I. Computer Vision for Graphics and Interaction.
Computer vision reconstructs real world information from image and video data; computer graphics synthesizes dynamic virtual worlds; interaction lets us explore these worlds; and machine learning allows us to map between domains across vision, graphics, and interaction. In visual computing, these fields converge to exploit both models of visual appearance and databases of examples to generate and interact with new images. This enables applications from the seemingly simple, like semantic photo editing, to the seemingly science fiction, like mixed reality. In this seminar, we will discover the state-of-the-art algorithmic contributions in computer vision which make this possible. Please join us!

Fall CSCI2951I S01 17628 MW 3:00-4:20(10) ‘To Be Arranged’

CSCI 2951K. Topics in Collaborative Robotics.
Practical approaches to designing intelligent systems. Topics include search and optimization, uncertainty, learning, and decision making. Application areas include natural language processing, machine vision, machine learning, and robotics. Prerequisite: CSCI 1410, 1420, 1460, 1480, or 1950F; or instructor permission.

CSCI 2951M. Advanced Algorithms Seminar.
Students in this course will read, present, and discuss recent breakthrough papers on the topic of algorithms, and the related areas needed to analyze algorithms. This course is aimed at current and potential future graduate students who want to gain technical depth and perspective on the field of algorithms. Topics will roughly alternate by year, with even years emphasizing fundamental techniques, and odd years emphasizing applications such as machine learning. Suggested prerequisites: CSCI 1570 and mathematical maturity. Instructor permission required. Enrollment will be limited to 24 students, based on an application that will be described on the first day of class. Ideal students will have a mix of the following: 1) motivation to learn how to read papers, 2) technical skills and background, 3) willingness to participate and contribute to discussions.

CSCI 2951N. Advanced Algorithms in Computational Biology.
This is a full-lecture, graduate course on algorithms and biomedical applications. The Foundations lectures are an introduction to the biological and medical genomics application areas. Each Algorithm section is devoted to an algorithmic method presented in rigorous depth, followed by an important open problem in the application area, together with the current most effective algorithmic solutions to the problem. Graduate students and advanced undergraduates in computational and mathematical sciences and engineering are welcome. Biological, life sciences and medical students and faculty are welcome as well and will be able to participate more in the applications areas.
CSCI 2951O. Foundations of Prescriptive Analytics. We are undoubtedly in the middle of an Analytics Revolution that enabled turning huge amounts of data into insights, and insights into predictions about the future. At its final frontiers, Prescriptive Analytics is aimed at identifying the best possible action to take given the constraints and the objective. To that end, this course provides students with a comprehensive overview of the theory and practice of how to apply Prescriptive Analytics through optimization technology. A wide variety of state-of-the-art techniques are studied including: Boolean Satisfiability, Constraint Programming, Linear Programming, Integer Programming, Local Search Meta-Heuristics, and Large-Scale Optimization. Pre Requisites: One of CSCI 0300, 0320, CSCI 0330, CSCI 1310, OR CSCI 1330 and recommended: one of CSCI 0530, CSCI 1570, MATH 0520 or MATH 0540.

CSCI 2951S. Distributed Computing through Combinatorial Topology. Although computer science itself is based on discrete mathematics, combinatorial topology and its applications may still be unfamiliar to many computer scientists. For this reason, this course provides a self-contained, elementary introduction to the concepts from combinatorial topology needed to analyze distributed computing. Conversely, while the systems and models used here are standard in computer science, they may be unfamiliar to students with a background in pure or applied mathematics. For this reason, this course also provides a self-contained, elementary description of standard notions of distributed computing. CSCI 0220 required, CSCI 1760 recommended.

CSCI 2951T. Data-Driven Computer Vision. Investigates current research topics in data-driven object detection, scene recognition, and image-based graphics. We will examine data sources, features, and algorithms useful for understanding and manipulating visual data. We will pay special attention to methods that harness large-scale or Internet-derived data. There will be an overview of the current crowdsourcing techniques used to acquire massive image datasets. Vision topics such as scene understanding and object detection will be linked to graphics applications such as photo editing. These topics will be pursued through independent reading, class discussion and presentations, and projects involving current research problems in Computer Vision.

CSCI 2951U. Topics in Software Security. This course investigates the state-of-the-art in software exploitation and defense. Specifically, the course is structured as a seminar where students present research papers to their peers. We will begin with a summary of prevalent software defects, typically found in applications written in memory unsafe languages, and proceed to surveying what we are up against: traditional and modern exploitation techniques, ranging from classical code injection and code reuse up to the newest goodies (JIT-ROP, Blind ROP). For the bulk part, we will focus on the latest advances in protection mechanisms, mitigation techniques, and tools against modern vulnerability classes and exploitation methods.

CSCI 2951X. Reintegrating AI. The goal of AI has been to build complete intelligent agents, yet the field has been fragmented into a collection of problem-specific areas of study. We will first spend a few weeks in lecture covering a new approach to integrating existing AI subfields into a single agent architecture, and remainder of the semester on self-directed, semester-long research projects. Grading based on a mid-semester project proposal, and a substantial open-ended final project. The projects will be multi-disciplinary in nature but students will have the opportunity to work in small groups, so they need not necessarily have expertise in the relevant areas. Graduate students welcome; undergraduates need instructor permission to enroll.

CSCI 2951Z. Advanced Algorithmic Game Theory. This course examines topics in game theory from a computer scientist's perspective. Through the lens of computation, it will focus on the design and analysis of systems involving self-interested agents, investigating how strategic behavior should influence algorithm design, which game-theoretic solution concepts are practical to implement, and the ramifications of conflicts of interest between system designers and participating agents. Students will create their own automated trading agents for various simulated market games. Topics include: auctions and mechanism design, equilibria, and learning. For graduate credit, students will complete additional homework exercises, and a significant programming project.

CSCI 2952B. Topics in Computer Science Education Research. How do people learn computing, and what can we do to teach them better? Answering these questions requires applying techniques from a variety of disciplines: computer science, naturally, but also cognitive science, psychology, linguistics, sociology, and more—even fields like economics can be relevant. This course studies different focused topics in computing education research (CER), drawing on these other disciplines as needed.

CSCI 2952C. Learning with Limited Labeled Data. As machine learning is deployed more widely, researchers and practitioners keep running into a fundamental problem: how do we get enough labeled data? This seminar course will survey research on learning when only limited labeled data is available. Topics covered include weak supervision, semi-supervised learning, active learning, transfer learning, and few-shot learning. Students will lead discussions on classic and recent research papers, and work in teams on final research projects. Previous experience in machine learning is required through CSCI 1420 or equivalent research experience.

CSCI 2952F. Distributed Systems at Scale: Microservices Management. This seminar investigates and explores cutting edge challenges and issues in the emerging Microservices paradigm. Microservices are a specific cloud paradigm for enabling distributed systems and applications at scale. In particular, this course builds on the foundations provided by the initial distributed systems, networking and operating systems offering (i.e., CSCI 1380, CSCI 1680, CSCI 1670) and explores how these concepts are used to realize, manage, and orchestrate microservices. The course is driven by materials from academic conferences and industrial blogs. The industrial blogs will provide context and motivation for different problems. The academic reasons will provide a deep divide into the technical details: we will focus on reading, analyzing, critiquing and brainstorming academic papers. Students taking this class should be familiar with reading academic literature, performing critical analysis, and working on open ended problems with undefined solutions. More information: http://cs.brown.edu/courses/info/csci2952-f/.

CSCI 2952G. Deep Learning in Genomics. Deep learning models have achieved impressive performance in fields like computer vision and NLP. Given an adequate amount of data, these models can extract meaningful representations to perform accurate predictions. The collection of vast quantities of biological data naturally leads to the question -- can deep learning help us understand genomics? In this seminar-style class, we will cover the recent research literature trying to answer this question. We will learn how state-of-the-art models like CNNs, RNNs, GGNs, GANs, etc. have been applied to solve significant problems in genomics and what unique challenges are presented by the data in this field.

Fall CSCI2952G S01 17505 TTh 1:00-2:20(06) (R. Singh)

CSCI 2952H. Recent Progress in Reinforcement Learning. Reinforcement learning is a framework for studying machines that interact with a sequential environment to achieve a goal. In the past decade, the RL framework has gained a lot of attention owing to its intriguing success in solving problems in complicated domains such as games, robotics, and dialog systems. We observe continual growth in the number of RL papers published in major machine-learning conferences. This growth calls for a careful investigation of the recent progress in the field. By reading selections of the current RL literature, this graduate-level course examines some of the latest theoretical and empirical progress in the field.
CSCI 2952L. Language Processing in Humans and Machines.
Understanding language requires transforming sequences of sounds into words, combining words into meaningful thoughts, and incorporating thoughts into an ongoing discourse. Psychologists and linguists have been trying to reverse-engineer how humans do this so easily, at the speed of conversation. In parallel, computer scientists have been trying to engineer machines to solve the same problems, leading to products like Siri and Alexa. This class will explore how these two kinds of research can help each other, bringing recent insights from machine learning into the study of human language processing, and insights from human processing into the architectures of machine language systems. For CS students: Machine Learning, Deep Learning, Computational Linguistics (or comparable experience). For CLPS students: At least one of CLPS 0200, 0300, 0800, or 1800

CSCI 2952K. Topics in 3D Computer Vision and Deep Learning.
We live in a world that spans 3 dimensions. Cameras and sensors image the 3D world by projecting to a 2D plane. How can we recover the 3D world back from these images? What techniques can we use to process 3D data? In this course we will study computer vision and machine learning techniques to recover 3D information of the world from images, and process and understand 3D data. We will learn about classical computer vision techniques but focus on cutting-edge deep learning methodologies. The techniques we will study are widely used, for instance, in self-driving cars and smartphone AR face filter apps.

CSCI 2952N. Advanced Topics in Deep Learning
Prepares graduate students with the knowledge they need to apply Deep Learning techniques for their own research. There has been tremendous success in developing unified neural architectures that achieve state-of-the-art performance on language understanding (GPT-3), visual perception (ViT), and even protein structure prediction (AlphaFold). We plan to understand how they work, and how the success of such unified models can give rise to further developments on self-supervised learning, a technique that trains machine learning models without requiring labeled data; and multimodal learning, a technique that utilizes multiple input sources, such as vision, audio, and text. We will study recent attempts to interpret these models, thus revealing potential risks on model bias. Paper reading, student presentations, and invited guest lectures. Students required to work on a final project that explores a novel direction along the line of the papers we cover.

CSCI 2952O. A Practical Introduction to Advanced 3D Robot Perception
This course is aimed at preparing graduate students and senior undergrads to do advanced work at the intersection of two important and popular fields: computer vision and robotics. The course will focus on the latest advances through lectures, readings, and discussion groups. The lectures and readings will be designed to represent a mix of classical techniques as well as the most recent advances in the two fields. The unique highlight of this course is the inclusion of a practical component: students will implement a project that combines computer vision and robotics by using cameras and a real robot arm. Students will form teams for this project and have exclusive access to a camera and a small robot arm both of which can be interfaced with the students’ laptops. Pre-reqs: One of CSCI 1430, CSCI 1470, CSCI 1951R, CSCI 1230.

CSCI 2952S. Topics in Cyber and Digital Policy
This online asynchronous course explores advanced topics in cybersecurity and digital policy, including privacy and civil liberties. Research papers and/or projects will provide an opportunity for creative work. Topics may include public policy and the international aspects of cybersecurity, legislation and legal requirements concerning digital policy issues, the history and background of privacy and civil liberties in information and information systems, cyber conflict, and related subjects. There is no set class time. Students and the instructor will coordinate work online and over Zoom through periodic meetings.
CSCI 1800, 1860, 1805, 1870 or equivalent background is a prerequisite. Instructor permission required.

CSCI 2955. The Design and Analysis of Trading Agents. The Dutch Flower Auctions (DFA) clear over 100,000 auctions per day, each lasting on average between 3 and 5 seconds! This semester, we’ll study the mechanism through which the DFA distribute 2/3 of the world’s flowers, focusing on both the sellers’ and buyers’ decision-making processes. More generally, we’ll research ways to automate and optimize decision-making in time-critical, information-rich environments, like the DFA. Undergraduate students require instructor permission, and should have already completed CSCI 0190, or CSCI 0150 and CSCI 0160, or CSCI 0170 and CSCI 0180.

CSCI 2956A. Design of Agents for Bidding in Sponsored Search Auctions.
This course investigates the new field of sponsored search auctions. Although students will be exposed to the field from the point of view of both the search engine and the advertiser, the course’s focus is on advertiser’s bidding algorithms. The students will implement novel bidding agents, and the course will culminate in a competition among the students’ agents. Undergraduate students who obtained permission from the instructor or completed CSCI 0910, or CSCI 0150 and CSCI 0160, or CSCI 0170 and CSCI 0180 can register for the course. CSCI 1410 is a corequisite.

CSCI 2956R. Multiplicative-Weights/Packing-Covering Method for Approximating Linear and Semidefinite Programs
We will study the method called, variously, multiplicative weights and packing-covering. We will in particular investigate the use of this method for finding approximately optimal solutions to linear programs and semidefinite programs. Prerequisite: A graduate-level course on algorithms. Enrollment limited to 10. Instructor permission required.

CSCI 2980. Reading and Research
Section numbers vary by instructor. Please check Banner for the correct section number and CRN to use when registering for this course.

CSCI 2990. Thesis Preparation
For graduate students who have met the residency requirement and are continuing research on a full time basis.

CSCI 2990. Thesis Preparation
For graduate students who have met the residency requirement and are continuing research on a full time basis.

CSCI XLIST. Courses of Interest to Concentrators in Computer Science.