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

Prerequisites (0-3 courses)
- Calculus prerequisite: students must complete or place out of second semester calculus.
  - MATH 0100 Introductory Calculus, Part II
  - or MATH 0170 Advanced Placement Calculus
  - or MATH 0190 Advanced Placement Calculus (Physics/Engineering)

Concentration Requirements

Core-Computer Science:
- Select one of the following introductory course Series:

**Series A**
- CSCI 0150 & CSCI 0160 Introduction to Object-Oriented Programming and Computer Science and Introduction to Algorithms and Data Structures

**Series B**
- CSCI 0170 & CSCI 0180 Computer Science: An Integrated Introduction and Computer Science: An Integrated Introduction

**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 0180, an intermediate-level course, or an advanced course)

Thirteen CS courses numbered 0220 or higher.
  # 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 course for the pathways used above

Intermediate Courses
- Students must complete the intermediate courses defined for the pathway they choose. In addition, ScB students must take at least one course from each intermediate course category to ensure they span all areas. Taking additional courses beyond those listed for the pathway may be required.

**Foundations**
- CSCI 0220 Introduction to Discrete Structures and Probability
- 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 Honors Linear Algebra
- CSCI 1450 Probability for Computing and Data Analysis
  - or APMA 1650 Statistical Inference I

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.
Intermediate Courses
CSCI 0320 Introduction to Software Engineering
CSCI 0330 Introduction to Computer Systems

Pathways
Completing a pathway entails taking two courses in the pathway of which at least one is a course 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: https://cs.brown.edu/degrees/undergrad/concentration-requirements-2020/pathways-for-undergraduate-and-masters-students/

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 1270 Database Management Systems
or CSCI 1310 Fundamentals of Computer 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 1730 Design and Implementation of Programming Languages
or CSCI 1760 Multiprocessor Synchronization
or CSCI 1950Y Logic for Systems
or ENGN 1640 Design of Computing Systems

Intermediate Courses
CSCI 0330 Introduction to Computer Systems
CSCI 0220 Introduction to Discrete Structures and Probability
or CSCI 0320 Introduction to Software Engineering

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 1730 Design and Implementation of Programming Languages
or CSCI 1950Y Logic for Systems

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

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

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
or CSCI 1580 Information Retrieval and Web Search
or ECON 1660 Big Data

Intermediate Courses
CSCI 0320 Introduction to Software Engineering
or CSCI 0330 Introduction to Computer Systems
MATH 0520 Linear Algebra
or MATH 0540 Honors Linear Algebra
or CSCI 0530 Coding the Matrix: An Introduction to Linear Algebra for Computer Science
CSCI 1450 Probability for Computing and Data Analysis
or APMA 1650 Statistical Inference I
or APMA 1655 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 1951R Introduction to Robotics

Related Courses
CSCI 1550 Probabilistic Methods in Computer Science
or CSCI 1951A Data Science
or CSCI 1951C Designing Humanity Centered Robots
or CSCI 1951K Algorithmic Game Theory
or ENGN 1610 Image Understanding

Intermediate Courses
CSCI 1450 Probability for Computing and Data Analysis
or APMA 1650 Statistical Inference I
or APMA 1655 Statistical Inference I
MATH 0520 Linear Algebra
or MATH 0540 Honors Linear Algebra
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
or CSCI 1550 Probabilistic Methods in Computer Science
or CSCI 1570 Design and Analysis of Algorithms
or CSCI 1760 Multiprocessor Synchronization

Related Courses
CSCI 1590 Introduction to Computational Complexity
or CSCI 1810 Computational Molecular Biology
or CSCI 1820 Algorithmic Foundations of Computational Biology
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1510</td>
<td>Introduction to Cryptography and Computer Security</td>
</tr>
<tr>
<td>or CSCI 1660</td>
<td>Introduction to Computer Systems Security</td>
</tr>
<tr>
<td>or CSCI 1650</td>
<td>Software Security and Exploitation</td>
</tr>
</tbody>
</table>

**Intermediate Courses**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1320</td>
<td>Creating Modern Web &amp; Mobile Applications</td>
</tr>
<tr>
<td>or CSCI 1380</td>
<td>Distributed Computer Systems</td>
</tr>
<tr>
<td>or CSCI 1670</td>
<td>Operating Systems</td>
</tr>
<tr>
<td>or CSCI 1680</td>
<td>Computer Networks</td>
</tr>
<tr>
<td>or CSCI 1730</td>
<td>Design and Implementation of Programming Languages</td>
</tr>
<tr>
<td>or CSCI 1800</td>
<td>Cybersecurity and International Relations</td>
</tr>
<tr>
<td>or CSCI 1805</td>
<td>Computers, Freedom and Privacy</td>
</tr>
<tr>
<td>or CSCI 1950Y</td>
<td>Logic for Systems</td>
</tr>
</tbody>
</table>

**Related Courses**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1230</td>
<td>Introduction to Computer Graphics</td>
</tr>
<tr>
<td>or CSCI 1250</td>
<td>Introduction to Computer Animation</td>
</tr>
<tr>
<td>or CSCI 1280</td>
<td>Intermediate 3D Computer Animation</td>
</tr>
<tr>
<td>or CSCI 1290</td>
<td>Computational Photography</td>
</tr>
<tr>
<td>or CSCI 1300</td>
<td>User Interfaces and User Experience</td>
</tr>
<tr>
<td>or CSCI 1370</td>
<td>Virtual Reality Design for Science</td>
</tr>
<tr>
<td>or CSCI 1430</td>
<td>Computer Vision</td>
</tr>
<tr>
<td>or CSCI 1950T</td>
<td>Advanced Animation Production</td>
</tr>
</tbody>
</table>

**Visual Computing: studies the creation, interaction, and analysis of images and visual information, including animation and games**

**Core Courses**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1950N</td>
<td>2D Game Engines</td>
</tr>
<tr>
<td>or CSCI 1470</td>
<td>Deep Learning</td>
</tr>
<tr>
<td>or CSCI 1950U</td>
<td>Topics in 3D Game Engine Development</td>
</tr>
<tr>
<td>or ENGN 1610</td>
<td>Image Understanding</td>
</tr>
<tr>
<td>or CLPS 1520</td>
<td>Computational Vision</td>
</tr>
<tr>
<td>or DATA 1200</td>
<td>Reality Remix - Experimental VR</td>
</tr>
</tbody>
</table>

**Intermediate Courses**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 0330</td>
<td>Introduction to Computer Systems</td>
</tr>
<tr>
<td>or CSCI 0320</td>
<td>Introduction to Software Engineering</td>
</tr>
<tr>
<td>or CSCI 0330</td>
<td>Introduction to Computer Systems</td>
</tr>
<tr>
<td>or CSCI 1450</td>
<td>Optimization Methods in Finance</td>
</tr>
<tr>
<td>or CSCI 1951G</td>
<td>Algorithmic Game Theory</td>
</tr>
<tr>
<td>or MATH 0520</td>
<td>Linear Algebra</td>
</tr>
<tr>
<td>or MATH 0540</td>
<td>Honors Linear Algebra</td>
</tr>
<tr>
<td>or MATH 0540</td>
<td>Linear Algebra for Computer Science</td>
</tr>
</tbody>
</table>

**Computer Architecture: studies the design, construction, and analysis of computer architecture and hardware**

**Core Courses**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENGN 1630</td>
<td>Digital Electronics Systems Design</td>
</tr>
<tr>
<td>or ENGN 1640</td>
<td>Design of Computing Systems</td>
</tr>
<tr>
<td>or ENGN 1650</td>
<td>Embedded Microprocessor Design</td>
</tr>
</tbody>
</table>

**Related Courses**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1600</td>
<td>Real-Time and Embedded Software</td>
</tr>
<tr>
<td>or CSCI 1760</td>
<td>Multiprocessor Synchronization</td>
</tr>
<tr>
<td>or ENGN 1600</td>
<td>Design and Implementation of Digital Integrated Circuits</td>
</tr>
</tbody>
</table>

**Intermediate Course**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 0330</td>
<td>Introduction to Computer Systems</td>
</tr>
</tbody>
</table>

**Computational Biology: studies the foundations and applications of algorithms for analyzing biological data and processes**

**Core Courses**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1810</td>
<td>Computational Molecular Biology</td>
</tr>
<tr>
<td>or CSCI 1820</td>
<td>Algorithmic Foundations of Computational Biology</td>
</tr>
<tr>
<td>or CSCI 1850</td>
<td>Deep Learning in Genomics</td>
</tr>
</tbody>
</table>

**Related Courses**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1420</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>or CSCI 1430</td>
<td>Computer Vision</td>
</tr>
<tr>
<td>or CSCI 1951A</td>
<td>Data Science</td>
</tr>
<tr>
<td>or CLPS 1520</td>
<td>Computational Vision</td>
</tr>
</tbody>
</table>

**Intermediate Courses**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 0220</td>
<td>Introduction to Discrete Structures and Probability</td>
</tr>
<tr>
<td>or CSCI 1010</td>
<td>Theory of Computation</td>
</tr>
<tr>
<td>or CSCI 1450</td>
<td>Probability for Computing and Data Analysis</td>
</tr>
<tr>
<td>or APMA 1650</td>
<td>Statistical Inference I</td>
</tr>
<tr>
<td>or APMA 1655</td>
<td>Statistical Inference I</td>
</tr>
</tbody>
</table>

**Design: studies the design, construction, and analysis of processes at the interface between humans and systems**

**Core Courses**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1300</td>
<td>User Interfaces and User Experience</td>
</tr>
<tr>
<td>or CSCI 1370</td>
<td>Virtual Reality Design for Science</td>
</tr>
<tr>
<td>or CSCI 1951C</td>
<td>Designing Humanity Centered Robots</td>
</tr>
</tbody>
</table>

**Related Courses**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1230</td>
<td>Introduction to Computer Graphics</td>
</tr>
<tr>
<td>or CSCI 1320</td>
<td>Creating Modern Web &amp; Mobile Applications</td>
</tr>
<tr>
<td>or CSCI 1600</td>
<td>Real-Time and Embedded Software</td>
</tr>
<tr>
<td>or CSCI 1900</td>
<td>csciStartup</td>
</tr>
<tr>
<td>or CSCI 1951A</td>
<td>Data Science</td>
</tr>
<tr>
<td>or CSCI 1951I</td>
<td>CS for Social Change</td>
</tr>
<tr>
<td>or CSCI 1951T</td>
<td>Surveying VR Data Visualization Software for Research</td>
</tr>
<tr>
<td>or VISA 1720</td>
<td>Physical Computing</td>
</tr>
</tbody>
</table>
Concentration Requirements (9 courses)

Core Computer Science:

Select one of the following series:

<table>
<thead>
<tr>
<th>Series A</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 0150 &amp; CSCI 0160</td>
<td>Introduction to Object-Oriented Programming and Computer Science and Introduction to Algorithms and Data Structures</td>
</tr>
</tbody>
</table>

Series B

<table>
<thead>
<tr>
<th>Series B</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 0170 &amp; CSCI 0180</td>
<td>Computer Science: An Integrated Introduction and Computer Science: An Integrated Introduction</td>
<td></td>
</tr>
</tbody>
</table>

Series C

<table>
<thead>
<tr>
<th>Series C</th>
<th>7</th>
</tr>
</thead>
<tbody>
<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 0180, an intermediate-level course, or an advanced course)</td>
</tr>
</tbody>
</table>

Seven CS courses numbered 0220 or higher

## One complete pathway (see ScB for pathways)

Requires two 1000-level courses as well as one-to-three intermediate 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 as part of the concentration). One course may be an approved 1000-level course from another department. Unless explicitly stated in a pathway, such non-CS courses may not be used as part of pathways.

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

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

Students must complete two two-to-four-month full-time professional experiences, doing work that is related to their concentration programs. Such work is normally done within an industrial organization, but may also be at a university under the supervision of a faculty member.

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.

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 two two-to-four-month full-time professional experiences, doing work that is related to their concentration programs. Such work is normally done within an industrial organization, but may also be at a university under the supervision of a faculty member.

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.

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

Prerequisites (0-3 courses)

Students must complete or place out of second semester calculus.

<table>
<thead>
<tr>
<th>MATH 0100</th>
<th>Introductory Calculus, Part II</th>
</tr>
</thead>
<tbody>
<tr>
<td>or MATH 0170</td>
<td>Advanced Placement Calculus</td>
</tr>
<tr>
<td>or MATH 0190</td>
<td>Advanced Placement Calculus (Physics/Engineering)</td>
</tr>
</tbody>
</table>

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 the student's seventh semester.

1. 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.

2. 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.

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 project. A senior thesis, which involved two semesters of work, may count as a capstone.


Prerequisites (3 courses):
- MATH 0100 Introductory Calculus, Part II
- MATH 0520 Linear Algebra
  or MATH 0540 Honors Linear Algebra
  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 Probability for Computing and Data Analysis 1
  or APMA 1650 Statistical Inference I
  or APMA 1655 Statistical Inference I

Select one of the following Series: 2

Series A
- CSCI 0150 & CSCI 0160 Introduction to Object-Oriented Programming and Computer Science and Introduction to Algorithms and Data Structures

Series B
- CSCI 0170 & CSCI 0180 Computer Science: An Integrated Introduction and Computer Science: An Integrated Introduction

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 0180, an intermediate-level CS course, or a 1000-level course.)

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)
- CSCI 1010 Theory of Computation (math) 2

A pair of 1000-level CS courses that, along with the intermediate courses and math courses, satisfy one of the CS Pathways. 3

Total Credits 17

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) 4
ECON 1210 Intermediate Macroeconomics 1
ECON 1630 Mathematical Econometrics I 1
ECON 1750 Investments II
ECON 1820 Theory of Behavioral Economics
ECON 1850 Theory of Economic Growth
ECON 1860 The Theory of General Equilibrium
ECON 1870 Game Theory and Applications to Economics

and any graduate Economics course

Two additional 1000-level Economics courses excluding 1620, 1960, 1970 5
One capstone course in either CS or Economics: a one-semester course, normally taken in the student’s last semester 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 (preferably at the intersection of computer science and economics) in depth, to produce a culminating artifact such as a paper or software project. A senior thesis, which involved two semesters of work, may count as a capstone. 6

Total Credits 17

1 CSCI 1450 was formerly known as CSCI 0450; they are the same course and hence only one may be taken for credit. APMA 1650 or APMA 1655 may be used in place of CSCI 1450 in CS pathway requirements. However, concentration credit will be given for only one for APMA 1650, APMA 1655, and CSCI 1450.

2 CSCI 1010 may be used either as a math-oriented intermediate course or as an advanced course. CSCI 1010 was formerly known as CSCI 0510: They are the same course and hence only one may be taken for credit.

3 A list of pre-approved pairs may be found at the approved-pairs web page (http://www.cs.brown.edu/ugrad/concentrations/approvedpairs.html). You are not restricted to pairs on this list, but any pair not on the list must be approved by the Computer Science director of undergraduate studies. CS Pathways can be found on the New Pathways (https://cs.brown.edu/degrees/undergrad/concentration-requirements/pathways-for-undergraduate-and-masters-students) page.

4 Or ECON 1110, with permission.

5 Students may use either ECON 1070 or ECON 1090 toward the concentration, but not both. Note that ECON 1620, ECON 1960, and ECON 1970 (independent study) cannot be used for concentration credit. However, 1960 and 1970 can be used for university credit and up to two 1970s may be used for university credit.
One capstone course (http://cs.brown.edu/degrees/undergrad/concentrations/capstone) in either Computer Science or Economics: 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 (preferably at the intersection of computer science and economics) in depth, to produce a culminating artifact such as a paper or software project.

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

**Prerequisites (3 courses):**
- MATH 0100 Introductory Calculus, Part II
- MATH 0520 Linear Algebra
  - or MATH 0540 Honors Linear Algebra
  - or CSCI 0530 Coding the Matrix: An Introduction to Linear Algebra for Computer Science
- ECON 0110 Principles of Economics

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

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 1450</td>
<td>Probability for Computing and Data Analysis</td>
</tr>
<tr>
<td>or APMA 1650</td>
<td>Statistical Inference I</td>
</tr>
<tr>
<td>or APMA 1655</td>
<td>Statistical Inference I</td>
</tr>
</tbody>
</table>

Select one of the following series:

**Series A**
- CSCI 0150 Introduction to Object-Oriented Programming and Computer Science and Introduction to Algorithms and Data Structures
- CSCI 0160

**Series B**
- CSCI 0170 Computer Science: An Integrated Introduction
  - and CSCI 0180

**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 0180, an intermediate-level course, or a 1000-level course)

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)
- CSCI 1010 Theory of Computation (math)

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.

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

Three courses from the "mathematical-economics" group: 2

- ECON 1170 Welfare Economics and Social Choice Theory
- ECON 1225 Advanced Macroeconomics: Monetary, Fiscal, and Stabilization Policies

**Applied Mathematics-Computer Science Concentration Requirements**

The Sc.B. concentration in Applied Math-Computer Science provides a foundation of basic concepts and methodology of mathematical analysis and computation and prepares students for advanced work in computer science, applied mathematics, and scientific computation. Concentrators must complete courses in mathematics, applied math, computer science, and an approved English writing course. While the concentration in
Applied Math-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.

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

Prerequisites - two semesters of Calculus, for example

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 090</td>
<td>Introductory Calculus, Part I</td>
</tr>
<tr>
<td>&amp; MATH 0100</td>
<td>and Introductory Calculus, Part II</td>
</tr>
<tr>
<td>MATH 0170</td>
<td>Advanced Placement Calculus</td>
</tr>
</tbody>
</table>

Concentration Requirements (17 courses)

Core-Math:

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 0180</td>
<td>Intermediate Calculus</td>
</tr>
<tr>
<td>or MATH 0350</td>
<td>Honors Calculus</td>
</tr>
<tr>
<td>MATH 0520</td>
<td>Linear Algebra</td>
</tr>
<tr>
<td>or MATH 0540</td>
<td>Honors Linear Algebra</td>
</tr>
<tr>
<td>or CSCI 0530</td>
<td>Coding the Matrix: An Introduction to Linear Algebra for Computer Science</td>
</tr>
</tbody>
</table>

Core-Applied Mathematics:

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>APMA 0350</td>
<td>Applied Ordinary Differential Equations</td>
</tr>
<tr>
<td>APMA 0360</td>
<td>Applied Partial Differential Equations I</td>
</tr>
<tr>
<td>APMA 1170</td>
<td>Introduction to Computational Linear Algebra</td>
</tr>
<tr>
<td>or APMA 1180</td>
<td>Introduction to Numerical Solution of Differential Equations</td>
</tr>
</tbody>
</table>

Core-Computer Science:

Select one of the following Series:

Series A

<table>
<thead>
<tr>
<th>Course</th>
<th>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

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

Series C

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<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 0180, an intermediate-level CS course, or a 1000-level course)</td>
</tr>
</tbody>
</table>

Select three of the following intermediate-level courses, one of which must be math-oriented and one systems-oriented. The intermediate courses must cover the requirements of the pathway chosen under additional requirements for CS.

<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>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>CSCI 1010</td>
<td>Theory of Computation (math)</td>
</tr>
<tr>
<td>CSCI 1450</td>
<td>Probability for Computing and Data Analysis (math)</td>
</tr>
<tr>
<td>or APMA 1650</td>
<td>Statistical Inference I</td>
</tr>
</tbody>
</table>

Three 1000-level Computer Science courses. Two of these courses and the intermediate courses must satisfy one of the CS pathways.

Three 1000-level Applied Mathematics courses approved by the concentration advisor, of which two should constitute a standard sequence or address a common theme. Typical sequences include: APMA 1200/1210 and APMA 1650 or 1655/1660. APMA 1910 cannot be used as an elective.

A capstone course: 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.

Note: CSCI 1010 and 1450 may be used either as a math-oriented intermediate course or as advanced courses. 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. However, concentration credit will be given for only one of Applied Math 1650, 1655, and CSCI 1450.

Total Credits: 17

1. APMA 1650 may only be used if not being used as an Applied Math course.
2. Pathways may be viewed here: https://cs.brown.edu/degrees/undergrad/concentrating-in-cs/concentration-requirements-2020/pathways-for-undergraduate-and-masters-students/
3. Capstone Options 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 two two-to-four-month full-time professional experiences, doing work that is related to their concentration programs. Such work is normally done within an industrial organization, but may also be at a university under the supervision of a faculty member.
- 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.
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 Computer Science department, 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  
MATH 0520  
MATH 0540 or CSCI 0530

Core Courses

MATH 1530 Abstract Algebra  
Select one of the following series:  
Series A  
CSCI 0150 & CSCI 0160 Introduction to Object-Oriented Programming and Computer Science and Introduction to Algorithms and Data Structures  
Series B  
CSCI 0170 & CSCI 0180 Computer Science: An Integrated Introduction and Computer Science: An Integrated Introduction  
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 0180, an intermediate-level CS course, or a 1000-level CS course)  
CSCI 0320 or CSCI 0330 Introduction to Software Engineering Introduction to Computer Systems  
CSCI 0220 Introduction to Discrete Structures and Probability or CSCI 1010 Theory of Computation  
Three 1000-level Mathematics courses  
Three advanced courses in Computer Science  
Three additional courses different from any of the above chosen from Mathematics, Computer Science, Applied Mathematics, or related areas  
A capstone course in Computer Science or Mathematics

Total Credits

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/concentration-requirements/pathways-for-undergraduate-and-masters-students). However, concentration credit will be given for only one of Applied Math 1650, 1655, and CSCI 1450.

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 two to four month full-time professional experiences, doing work that is related to their concentration programs. Such work is normally done within an industrial organization, but may also be at a university under the supervision of a faculty member.

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 coursework?
• 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

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.

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 1970 and 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/concentration-requirements/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 be used as part of options 1, 2, 3, and 4, but not 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 such experience to choose of 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
- CSCI 1230 Introduction to Computer Graphics
- CSCI 1250 Introduction to Computer Animation
- CSCI 1260 Compilers and Program Analysis
- CSCI 1270 Database Management Systems
- CSCI 1280 Intermediate 3D Computer Animation
- CSCI 1300 User Interfaces and User Experience
- CSCI 1310 Fundamentals of Computer Systems
- CSCI 1320 Creating Modern Web & Mobile Applications
- CSCI 1370 Virtual Reality Design for Science
- CSCI 1380 Distributed Computer Systems
- CSCI 1410 Artificial Intelligence
- CSCI 1420 Machine Learning
- CSCI 1430 Computer Vision
- CSCI 1450 Probability for Computing and Data Analysis
- CSCI 1460 Computational Linguistics
- CSCI 1510 Introduction to Cryptography and Computer Security
- CSCI 1550 Probabilistic Methods in Computer Science
- CSCI 1570 Design and Analysis of Algorithms
- CSCI 1590 Introduction to Computational Complexity
- CSCI 1600 Real-Time and Embedded Software
- CSCI 1610 Building High-Performance Servers
- CSCI 1660 Introduction to Computer Systems Security
- CSCI 1670 Operating Systems
- CSCI 1680 Computer Networks
- CSCI 1730 Design and Implementation of Programming Languages
- CSCI 1760 Multiprocessor Synchronization
- CSCI 1780 Parallel and Distributed Programming
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.

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, Photoshop, Excel and Python assignments. CSCI 0020 is a good introduction to a wide range of CS topics that have broad relevance in our society. No prerequisites.

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 16576 Arranged (K. Fisler)

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 16577 Arranged (T. Doeppner)

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 and use of digital information. Designed for both concentrators and non-concentrators, this is the first in an eventual 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 16579 MWF 10:00-10:50(14) (K. Fisler)

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 17520 MWF 1:00-1:50(06) (D. Woos)

CSCI 0120. User Interfaces and User Experience.
Topics include understanding when to use different interfaces, modeling and representing user interaction, principles of user experience design, eliciting requirements and feedback from users, methods for designing and prototyping interfaces, and user interface evaluation. Students interested in learning the process behind building a user interface and gaining hands-on experience designing a user interface should take this course. Programming experience is unnecessary. There will be assignments, readings, and design labs. CSCI 0130 is the same lecture, labs, and readings as CSCI 1300 but half of the assignments will be different (CSCI 1300 will have assignments with computer science prerequisites).

Website: http://cs.brown.edu/courses/csci1300/

CSCI 0150. Introduction to Object-Oriented Programming and Computer Science.
Emphasizes object-oriented design and programming in Java, an effective modern technique for producing modular, reusable, internet-aware programs. Also introduces interactive computer graphics, user interface design and some fundamental data structures and algorithms. A sequence of successively more complex graphics programs, including Tetris, and culminating in a significant final project, helps provide a serious introduction to the field intended for both potential concentrators and those who may take only a single course. No prerequisites, no prior knowledge of programming required.

CSCI 0160. Introduction to Algorithms and Data Structures.
Introduces fundamental techniques for problem solving by computer that are relevant to most areas of computer science, both theoretical and applied. Algorithms and data structures for sorting, searching, graph problems, and geometric problems are covered. Programming assignments conform with the object-oriented methodology introduced in CSCI 0150. Prerequisite: CSCI 0150 or written permission.

CSCI 0170. Computer Science: An Integrated Introduction.
CSCI0170/0180 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.

CSCI 0180. Computer Science: An Integrated Introduction.
A continuation of CSCI 0170. Students learn to program in Java while continuing to develop their algorithmic and analytic skills. Emphasis is placed on object-oriented design, imperative programming, and the implementation and use of data structures. Examples are drawn from such areas as databases, strategy games, web programming, graphical user interfaces, route finding, and data compression. Lab work done with the assistance of TAs. Prerequisite: CSCI 0170 or CSCI 0190.
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 0150-0160 and CSCI 0170-0180). Students wishing to take CSCI 0190 must pass a sequence of online placement assignments. 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/csci0190/. Students who do not successfully pass the placement process won't be allowed to register.

Fall CSCI0190 S01 16583 Arranged (S. Krishnamurthi)

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.

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 secure and performing 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, or 0190; 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.
Techniques for designing, building, and maintaining large, scalable, and reusable systems. We will cover advanced programming techniques using Java and Javascript. Course assignments will familiarize students with software testing, relational databases, concurrency techniques such as threads, and software engineering tools like git, profilers, and debuggers. A major component of the course will be a group software project of your own design.
Prerequisite: CSCI 0160, CSCI 0180 or CSCI 0190; 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 0150, 0180, or 0190.

Fall CSCI0330 S01 16584 MWF 2:00-2:50(10) (T. Doepner)
Fall CSCI0330 S02 18536 Arranged (T. Doepner)

CSCI 0530. 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 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 1450.

Fall CSCI1010 S01 17249 TTh 10:30-11:50(13) (L. De Stefani)

Fundamental concepts in 2D and 3D computer graphics, e.g., 2D raster graphics techniques, simple image processing, and user interface design. 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, or CSCI 0190. 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.

Fall CSCI1230 S01 16589 Arranged (J. Hughes)

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.

Fall CSCI1234 S01 16763 Arranged (J. Hughes)

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.

Please see course website for application procedure.

Fall CSCI1250 S01 16588 MW 12:00-1:50(15) (B. Meier)

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

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 0330 or CSCI 0320 is strongly recommended.

Fall CSCI1270 S01 16586 TTh 2:30-3:50(12) (S. Zdonik)
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 push instead of pull? 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/. Prerequisite: CSCI 0160, CSCI 0180; or permission of the instructor.

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. Prerequisites: CSCI 0160, 0180, or CSCI 0190; and CSCI 0150, 0170 if concurrently taking CSCI 0160 or 0180; or permission of 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) 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.

Fall CSCI1330 S01 18020 TTh 2:00-2:50(10) (T. Doepner)
Fall CSCI1330 C01 18107 F 10:00-10:50 (T. Doepner)

CSCI 1340. Innovating Game Development.
What technologies will shape the next generation of videos? This project-centered course focuses on computational innovations for game development. Students examine innovative game technology through case studies of existing games and talks by industrial and academic game professionals. In teams, students propose and implement a project demonstrating a novel technology for gaming. Recommended: strong computational or engineering background.

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 0320 or CSCI 0330.

CSCI 1410. Artificial Intelligence.
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. Prerequisites: CSCI 0160, CSCI 0180 or CSCI 0190; and one of CSCI0220 or CSCI1450 or APMA1650 or APMA1655.

Fall CSCI1410 S01 17200 TTh 9:00-10:20(02) (A. Greenwald)

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. Computation in Economics and Games.
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. Probability for Computing and Data Analysis.
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. Students cannot get concentration credit for both CSCI 1450 and APMA 1650/1655.

Fall CSCI1450  S01 16587 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. The prerequisite of CS 1470 (or the equivalent background) is very important.

Deep learning is the name for a particular version of neural networks—a version that emphasizes multiple layers of networks. Deep learning, plus the specialized techniques that it has inspired (e.g., convolutional features and word embeddings) have lead to rapid improvements in many applications such as computer vision, machine translation, and computer Go. This course intends to give students a practical understanding of deep learning as applied in these and other areas. It also teaches the Tensorflow programming language for the expression of deep leaning algorithms. (The primary API for Tensorflow is from Python.)

Fall CSCI1470  S01 16590 Arranged  (D. Ritchie)

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 1500. 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.

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 17250 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, those enclosed in devices such as cellular phones, game consoles, and car engines. Includes the overall embedded real-time software design and development processes, as well as aspects of embedded hardware and real-time, small-footprint operating systems. Major project component. Prerequisites: CSCI 0320 or 0330.

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 0380. CSCI 1670 or 1680 is recommended.
CSCI 1620 is a half-credit laboratory course intended to be taken concurrently with CSCI 1660 and provides students with a deeper understanding of the material by doing additional assignments, which include extensions of the 1660's assignments. Instructor permission required.

CSCI 1650. Software Security and Exploitation.
Covers software exploitation techniques and state-of-the-art mechanisms for protecting (vulnerable) software. It 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 newest goodies (just-in-time code reuse). For the most part, it focuses on defenses against certain vulnerability classes and exploitation methods. Students will learn about the boundaries and effectiveness of virtualization, stack and heap protections, and address space randomization, and analyze advanced exploitation techniques and countermeasures.

Fall CSCI1650 S01 16585 Arranged (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 or CSCI 0180 or CSCI 0190; and CSCI 0330.

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 0330.

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 the issues are, how they work, their shortcomings, and what improvements are on the horizon. Prerequisite: CSCI 0330 or consent of instructor.

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.

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 principles of modern programming languages by implementation. Examines linguistic features, especially control operators such as first-class functions, exceptions, and continuations. Studies data and their types, including polymorphism, type inference, and type soundness. Examines compiler and run-time system topics: continuation-passing style and garbage collection. Prerequisite: CSCI 0160. CSCI 0180 or CSCI 0190. Preferred: CSCI 0220, either CSCI 0320 or CSCI 0330, and CSCI 0510.

Fall CSCI1730 S01 16591 Arranged (S. Krishnamurthi)
Fall CSCI1730 C01 18086 M 10:00-10:50 (S. Krishnamurthi)
Fall CSCI1730 C02 18087 W 10:00-10:50 (S. Krishnamurthi)
Fall CSCI1730 C03 18088 F 10:00-10:50 (S. Krishnamurthi)
Fall CSCI1730 C04 18089 M 10:00-10:50 (S. Krishnamurthi)
Fall CSCI1730 C05 18090 W 10:00-10:50 (S. Krishnamurthi)
Fall CSCI1730 C06 18219 F 10:00-10:50 (S. Krishnamurthi)
Fall CSCI1730 C07 18552 Arranged "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

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.

CSCI 1805. Computers, Freedom and Privacy. 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 16846 TTh 9:00-10:20(02) (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, and 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 16593 TTh 2:30-3:50(12) (S. Istrail)
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 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.

CSCI 1950A. Computational Modeling and Algorithmic Thinking.
In this course you will learn how to apply tools from statistics and computer science to build computational models of physical and biological systems. Example applications include modeling and then simulating the behavior of a collection of genes, the spread of disease in a population, a single neuron in isolation or the complex of neurons comprising the primate visual cortex.

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 email the instructor. Within their override request, students should include a brief statement (three sentences max) on why they wish to take the course. Priority will be given to both seniors and juniors.

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.

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 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. Prerequisites: CSCI 1230 and one of CSCI 0320 or CSCI 1950N. 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.

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 constructors, model checkers, and proof assistants. Problems and projects will apply to real-world systems. Prerequisite: CSCI 0160, CSCI 0180, or CSCI 0190. Preferred but not required: CSCI 0220 and CSCI 0510, or instructor's permission.

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, or CSCI 0190. One of CSCI 0330 or CSCI 0320 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 Robots.
Offered by Brown’s Computer Science department under the auspices of the Humanity Centered Robotics Initiative. It is focused on the iterative design process and how it can be used to develop robots for solving tasks that help people. It will expose students to a suite of fabrication and prototyping technologies sufficient for creating a functioning robotic system.

https://www.youtube.com/watch?v=Dbvis_j_b78

The course has two tracks, one intended for CS concentrators, and one intended for non-concentrators with previous design experience. The non-concentrator track cannot be used toward fulfilling a Computer Science concentration requirement.

Fall CSCI1951C S01 16595 MW 9:00-11:50(16) (I. Gonsher)

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 1951L. 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 positionalities and ethics in engaging in social impact work and what it practically means to leverage technology to create social change on an everyday basis. Enrollment limited to 12. Entry to this course is through application only: https://docs.google.com/forms/d/1wmCbmB6dpOlb-FCjHe50IHgxA08qCE38mTdd71JuU/edit

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 1951K. Algorithmic Game Theory.
This course examines topics in game theory from a computer scientist’s perspective. Through the lens of computation, this course 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. Topics include: auctions and mechanism design, equilibria, and learning.

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 1951O. Design of Robotic Systems.
Designing kinetic systems (i.e., systems that require movement or motion) relies on both mechanical and electrical engineering. These systems include everything from mobile robots used for rescue operation to electrically powered moving sculptures. Through a series of projects, students combine their knowledge of electronic circuit design, sensors, actuators, motors, microcontrollers, control theory and programming to build interactive art and robotic systems. Projects culminate in the design of a creative kinetic system that incorporates several of the principles learned in class. Some programming experience is helpful but not required. While there are no specific prerequisites for this course, it is assumed that students have some prior programming experience (e.g., CSCI111, 150, 170, ENGN30, 31, 32). Some prior digital system/ circuit design, or robotics experience is also very helpful (e.g., ENGN500, ENGN520, ENGN1630, CSCI1951C, CSCI1951R), but not required.

CSCI 1951R. 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.

Fall CSCI1951R S01 16764 Arranged (S. Tellex)

CSCI 1951T. 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 surveying. Results will be documented in a courses wiki.
CSCI 191U. Software Engineering of Large Systems.
This course teaches the software engineering techniques used to create moderate to large sized programs. This includes design methodologies that scale and are geared toward larger, long-lifetime systems with multiple designers. It includes tools and techniques that assist the programmer and help manage programmer teams.
 Fall CSCI191U S01 16886 TTh 9:00-10:20(02) (S. Reiss)

CSCI 191V. Hypertext/Hypermedia Seminar: The Web Was Not the Beginning and the Web Is Not the End.
A "hypertext" system is for creating, linking, exploring, annotating, and searching for information. Designed 30-years ago, the World Wide Web is a hypertext system gone global. But the Web represents only a small part of past visions. Students will explore hypertext’s history through hands-on use of systems from the 1970s onward, identifying features still not in common use. They will study the architecture, design, and features of hypertext systems, examining topics such as permanence, collaboration, searching, content design patterns, and societal impact. A final project will explore what hypertext systems might look like in the 2020s. Prerequisites: an introductory CS sequence or equivalent experience
 Fall CSCI195V S01 16873 W 3:00-5:30 (A. van Dam)

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 in practice. We must instead turn to new algorithmic approaches and paradigms, which allow us to answer valuable questions about our data in runtime that is still feasible 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: (CSCI22 or equivalent); (CS145 or APMA1650/1655 or equivalent); (CS157 or CS155). Mathematical maturity is essential: this is a theory course with proofs. Recommended: CS155
 Fall CSCI191WS01 18537 TTh 2:30-3:50(12) (C. Lee)

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.


Important current topics in computer graphics. Course includes reading and discussing current research papers, multiple assignments and preliminary projects in which students implement recent papers, and a demanding final integrative project done in small groups. Prerequisite: Instructor’s permission or CSCI 1230.

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 to be widely visible or to be published. 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 will find some topics in CSCI 1300 relevant. Students should have taken CSCI 1300 or make a case for their interest.

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, 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. Prerequisite: CSCI 0320 or CSCI 0330.

CSCI 2370. Interdisciplinary Scientific Visualization.
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. Immersive CAVE applications will be a focus, but other interaction or visualization projects are possible. Prerequisites: all: programming experience; CS students: graphics experience; others: problem ideas.
Instructor permission required.
 Fall CSCI2370 S01 16596 TTh 10:30-11:50(13) (D. Laidlaw)

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 work on services, datacenter systems, distributed computing 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.
 Fall CSCI2390 S01 17379 TTh 2:30-3:50(12) (M. Schwarzkopf)

CSCI 2410. Statistical Models in Natural-Language Understanding.
Various topics in computer understanding of natural language, primarily from a statistical point of view. Topics include: hidden Markov models, word-tagging models, probabilistic context-free grammars, syntactic disambiguation, semantic word clustering, word-sense disambiguation, machine translation and lexical semantics. Prerequisite: CSCI 1410.
CSCI 2420. 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. Computation in Economics and Games.
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 15464 Arranged To Be Arranged

Deep learning is the name for a particular version of neural networks--a version that emphasizes multiple layers of networks. Deep learning, plus the specialized techniques that it has inspired (e.g. convolutional features and word embeddings) have lead to rapid improvements in many applications such as computer vision, machine translation, and computer Go. This course intends to give students a practical understanding of deep learning as applied in these and other areas. It also teaches the Tensorflow programming language for the expression of deep learning algorithms. A final project will implement an advanced piece of work in one of these areas. Pre Requisites: A basic programming course: (CSCI 0150, 0170 or 0190) A linear algebra course: (CSCI 0530, MATH 0520 or 0540) A stats / probability course: (CSCI 0220, 1450, 0450, MATH 1610, APMA 1650 or 1655)

Fall CSCI2470 S01 16598 Arranged (D. Ritchie)

CSCI 2500A. Advanced Algorithms.
Typically, an algorithm solves one problem, whereas a well-designed data structure can help implement algorithms for a wide variety of problems. We will study the design, analysis and implementation of advanced data structures. Focus is on data structures that are fast, both theoretically and empirically. 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 sattifiability 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/semester 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 algorithms. 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 2950. 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 composability, anonymous credentials and e-cash, interplay of cryptography and game theory. May be repeated for credit. Prerequisite: CSCI 1510 or permission of the instructor.

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.
Explores widely-distributed systems that take advantage of resources, fault-tolerance, high-availability and diverse data collection. Topics include experimental) in efficient solution for such problems. We'll read and rearrangements and assembly; and protein and regulatory interaction models for spatial, temporal, or structured data; and variational or Monte Carlo approximations. Course meetings combine lectures with 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 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. 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 meets 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 student 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 an individual 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. An important example is the purely adversarial problem of a negotiation process, such as that found in online auctions. The course will cover recent work on modeling and solving these problems. The course is designed to be accessible to students with a background in computer science and some exposure to reinforcement learning and multiagent planning as well as autonomous agents that trade in their market simulation. Application domains will range from 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 from a systems perspective. Topics include vulnerabilities and defenses for automotive, computing, medical, and industrial control devices, intrusion detection, firewalls, secure network protocols, web spam, tracking of web users, JavaScript sandboxing, attacks and defenses for web applications, and security and privacy issues in cloud computing. Research papers and industry reports will be presented and discussed. Also, hands-on experiments and system demonstrations will be performed. CSCI 1660 or equivalent background is essential. Enrollment limited to 12. Instructor permission required.

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. Each student will be expected to present a published research paper and will participate in a group programming project. Prerequisite: a graduate-level computer science course and some exposure to reinforcement learning from a previous computer-science class or seminar.

CSCI 2951G. 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!
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. Instructor permission required.

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 undergarduates 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 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 0320 or CSCI 0330 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 goodries (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 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.

Fall CSCI2952C S01 16920 TTh 1:00-2:20(08) (S. Bach)

CSCI 2952F. Distributed Systems at Scale: Microservices Management. This course investigates and explores an emerging paradigm for enabling distributed systems and applications at scale. Microservices. In particular, this course builds on the foundations provided by the initial distributed systems offering (i.e., CSCI 0138) and explores how these concepts are used to realize, manage, and orchestrate microservices. Pre-req: CSCI 1380 For more information, please see http://cs.brown.edu/courses/info/ csci2952-f/

Fall CSCI2952F S01 17549 MW 3:20-4:20(10) (T. Benson)

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, GCNs, 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 16919 TTh 1:00-2:20(08) (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 2952J. Topics in Computing with Emerging Technologies.
In the past, computer system performance was driven by improvements in silicon fabrication technology. However, a number of promising candidates for new basic technologies have emerged recently, including single-molecule organic switches and non-volatile memory structures. This course will consider how these new basic devices will affect our past assumptions about computing from both hardware and software perspectives. Class will include a mix of lectures and discussion on assigned reading of recent publications. Students will be responsible for leading and participating in these discussions. A course project will also be required. Prerequisites: some knowledge of computer architecture is helpful, but not required.

CSCI 2952K. Topics in 3D Computer Vision and Deep Learning.
We live in a 3D world where all objects span 3 dimensions (length, width, and height). Cameras that image the world project 3D information to a 2D plane. How can we recover the 3D world back from 2D images? In this course, we will learn about computer vision and deep learning techniques to recover 3D information from the world images. We will study classical computer vision techniques but focus on cutting-edge deep learning methods. If you have ever wondered about the algorithms behind self-driving cars and AR face filter apps on your phone, then this course is for you.

CSCI 2952V. 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 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 2955A. 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 co-requisite.
Font Notice

This document should contain certain fonts with restrictive licenses. For this draft, substitutions were made using less legally restrictive fonts. Specifically:

Helvetica was used instead of Arial.

The editor may contact Leepfrog for a draft with the correct fonts in place.