**Computer Science**

Courses offered by the Department of Computer Science are listed under the subject code CS on the Stanford Bulletin’s ExploreCourses web site.

The Department of Computer Science (CS) operates and supports computing facilities for departmental education, research, and administration needs. Current CS students have access to a departmental student machine for general use and computer labs located in the Gates Building. In addition, most students have access to systems located in their research areas.

Each research group in Computer Science has systems specific to its research needs. These systems include workstations, computer clusters, GPU clusters, and local file servers. Servers and workstations running Linux, MacOS, or various versions of Windows are commonplace. Support for course work and instruction is provided on systems available through U (http://itservices.stanford.edu) University IT (https://uit.stanford.edu) (UIT) and the School of Engineering (http://engineering.stanford.edu) (SoE).

**Mission of the Undergraduate Program in Computer Science**

The mission of the undergraduate program in Computer Science is to develop students’ breadth of knowledge across the subject areas of computer science, including their ability to apply the defining processes of computer science theory, abstraction, design, and implementation to solve problems in the discipline. Students take a set of core courses. After learning the essential programming techniques and the mathematical foundations of computer science, students take courses in areas such as programming techniques, automata and complexity theory, systems programming, computer architecture, analysis of algorithms, artificial intelligence, and applications. The program prepares students for careers in government, law, and the corporate sector, and for graduate study.

**Learning Outcomes (Undergraduate)**

The department expects undergraduate majors in the program to be able to demonstrate the following learning outcomes. These learning outcomes are used in evaluating students and the department’s undergraduate program. Students are expected to be able to:

1. Apply the knowledge of mathematics, science, and engineering.
2. Design and conduct experiments, as well to analyze and interpret data.
3. Design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
4. Function on multidisciplinary teams.
5. Identify, formulate, and solve engineering problems.
6. Understand professional and ethical responsibility.
7. Communicate effectively.
8. Understand the impact of engineering solutions in a global, economic, environmental, and societal context.
9. Demonstrate a working knowledge of contemporary issues.
10. Apply the techniques, skills, and modern engineering tools necessary for engineering practice.
11. Transition from engineering concepts and theory to real engineering applications.

**Learning Outcomes (Graduate)**

The purpose of the master’s program is to provide students with the knowledge and skills necessary for a professional career or doctoral studies. This is done through course work in the foundational elements of the field and in at least one graduate specialization. Areas of specialization include artificial intelligence, biocomputation, computer and network security, human-computer interaction, information management and analytics, real-world computing, software theory, systems, and theoretical computer science.

The Ph.D. is conferred upon candidates who have demonstrated substantial scholarship and the ability to conduct independent research. Through course work and guided research, the program prepares students to make original contributions in Computer Science and related fields.

**Graduate Programs in Computer Science**

The University’s basic requirements for the M.S. and Ph.D. degrees are discussed in the “Graduate Degrees (http://exploredegrees.stanford.edu/graduatedegrees)” section of this bulletin.

**Computer Science Course Catalog Numbering System**

The first digit of a CS course number indicates its general level of sophistication:

<table>
<thead>
<tr>
<th>Digit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>001-099</td>
<td>Service courses for nontechnical majors</td>
</tr>
<tr>
<td>100-199</td>
<td>Other service courses, basic undergraduate</td>
</tr>
<tr>
<td>200-299</td>
<td>Advanced undergraduate/beginning graduate</td>
</tr>
<tr>
<td>300-399</td>
<td>Advanced graduate</td>
</tr>
<tr>
<td>400-499</td>
<td>Experimental</td>
</tr>
<tr>
<td>500-599</td>
<td>Graduate seminars</td>
</tr>
</tbody>
</table>

The tens digit indicates the area of Computer Science it addresses:

<table>
<thead>
<tr>
<th>Digit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00-09</td>
<td>Introductory, miscellaneous</td>
</tr>
<tr>
<td>10-19</td>
<td>Hardware and Software Systems</td>
</tr>
<tr>
<td>20-39</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>40-49</td>
<td>Software Systems</td>
</tr>
<tr>
<td>50-59</td>
<td>Mathematical Foundations of Computing</td>
</tr>
<tr>
<td>60-69</td>
<td>Analysis of Algorithms</td>
</tr>
<tr>
<td>70-79</td>
<td>Computational Biology and Interdisciplinary Topics</td>
</tr>
<tr>
<td>90-99</td>
<td>Independent Study and Practicum</td>
</tr>
</tbody>
</table>

**Bachelor of Science in Computer Science**

The department offers both a major in Computer Science and a minor in Computer Science. Further information is available in the Handbook for Undergraduate Engineering Programs (UGHB) (http://ughb.stanford.edu) published by the School of Engineering. The Computer Science major offers a number of tracks (programs of study) from which students can choose, allowing them to focus their program on the areas of most interest. These tracks also reflect the broad diversity of areas in computing disciplines. The department has an honors program.
In addition to Computer Science itself, Stanford offers several interdisciplinary degrees with a substantial computer science component. The Symbolic Systems major (in the School of Humanities and Sciences) offers an opportunity to explore computer science and its relation to linguistics, philosophy, and psychology. The Mathematical and Computational Sciences major (also Humanities and Sciences) allows students to explore computer science along with more mathematics, statistics, and operations research.

**Computer Science (CS)**

Completion of the undergraduate program in Computer Science leads to the conferral of the Bachelor of Science in Computer Science.

**Mission of the Undergraduate Program in Computer Science**

The mission of the undergraduate program in Computer Science is to develop students' breadth of knowledge across the subject areas of computer science, including their ability to apply the defining processes of computer science theory, abstraction, design, and implementation to solve problems in the discipline. Students take a set of core courses. After learning the essential programming techniques and the mathematical foundations of computer science, students take courses in areas such as programming techniques, automata and complexity theory, systems programming, computer architecture, analysis of algorithms, artificial intelligence, and applications. The program prepares students for careers in government, law, the corporate sector, and for graduate study.

**Requirements**

**Mathematics (26 units minimum)**

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 103</td>
<td>Mathematical Foundations of Computing</td>
<td>5</td>
</tr>
<tr>
<td>CS 109</td>
<td>Introduction to Probability for Computer Scientists</td>
<td>5</td>
</tr>
<tr>
<td>MATH 19</td>
<td>Calculus</td>
<td>1</td>
</tr>
<tr>
<td>MATH 20</td>
<td>Calculus</td>
<td>1</td>
</tr>
<tr>
<td>MATH 21</td>
<td>Calculus</td>
<td>1</td>
</tr>
<tr>
<td>Plus two electives</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Science (11 units minimum)**

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHYSICS 41</td>
<td>Mechanics</td>
<td>4</td>
</tr>
<tr>
<td>or PHYSICS 41E</td>
<td>Mechanics, Concepts, Calculations, and Context</td>
<td></td>
</tr>
<tr>
<td>PHYSICS 43</td>
<td>Electricity and Magnetism</td>
<td>4</td>
</tr>
<tr>
<td>Science elective</td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

**Technology in Society (3-5 units)**

One course; course chosen must be on the SoE Approved Courses list at https://ughb.stanford.edu/ the year taken; see Basic Requirements 4 in the School of Engineering section.

**Engineering Fundamentals (13 units minimum; see Basic Requirement 3 in the School of Engineering section)**

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 106B</td>
<td>Programming Abstractions</td>
<td>5</td>
</tr>
<tr>
<td>or CS 106X</td>
<td>Programming Abstractions</td>
<td></td>
</tr>
<tr>
<td>ENGR 40M</td>
<td>An Intro to Making: What is EE (or ENGR 40A and ENGR 40B)</td>
<td>3-5</td>
</tr>
<tr>
<td>Fundamentals Elective (May be an ENGR fundamentals or an additional CS Depth course. See Fig. 3-4 in the UGHB for approved ENGR fundamentals list. May not be any CS 106)</td>
<td></td>
<td>3-5</td>
</tr>
</tbody>
</table>

*Students who take ENGR 40A or 40M for fewer than 5 units are required to take 1-2 additional units of ENGR Fundamentals (13 units minimum), or 1-2 additional units of Depth.

**Writing in the Major**

- Select one of the following:
  - CS 181W: Computers, Ethics, and Public Policy
  - CS 191W: Writing Intensive Senior Project
  - CS 194W: Software Project
  - CS 210B: Software Project Experience with Corporate Partners
  - CS 294W: Writing Intensive Research Project in Computer Science

**Computer Science Core (15 units)**

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 107</td>
<td>Computer Organization and Systems</td>
<td>5</td>
</tr>
<tr>
<td>or CS 107E</td>
<td>Computer Systems from the Ground Up</td>
<td></td>
</tr>
<tr>
<td>CS 110</td>
<td>Principles of Computer Systems</td>
<td>5</td>
</tr>
<tr>
<td>CS 161</td>
<td>Design and Analysis of Algorithms</td>
<td>5</td>
</tr>
</tbody>
</table>

**Senior Project (3 units)**

- Select two courses, each from a different area:
  - Area I, A.I Methods:
    - CS 228: Probabilistic Graphical Models: Principles and Techniques
    - CS 229: Machine Learning
    - CS 234: Reinforcement Learning
    - CS 238: Decision Making under Uncertainty
  - Area II, Natural Language Processing:
    - CS 124: From Languages to Information
    - CS 224N: Natural Language Processing with Deep Learning
    - CS 224S: Spoken Language Processing
    - CS 224U: Natural Language Understanding
  - Area III, Vision:
    - CS 131: Computer Vision: Foundations and Applications

**Computer Science Depth B.S.**

Choose one of the following ten CS degree tracks (a track must consist of at least 25 units and 7 classes):

**Artificial Intelligence Track**

Select two courses, each from a different area:

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 221</td>
<td>Artificial Intelligence: Principles and Techniques</td>
<td>4</td>
</tr>
</tbody>
</table>

**Area I, A.I Methods**

- CS 228: Probabilistic Graphical Models: Principles and Techniques
- CS 229: Machine Learning
- CS 234: Reinforcement Learning
- CS 238: Decision Making under Uncertainty

**Area II, Natural Language Processing**

- CS 124: From Languages to Information
- CS 224N: Natural Language Processing with Deep Learning
- CS 224S: Spoken Language Processing
- CS 224U: Natural Language Understanding

**Area III, Vision**

- CS 131: Computer Vision: Foundations and Applications
CS 231A Computer Vision: From 3D Reconstruction to Recognition
CS 231N Convolutional Neural Networks for Visual Recognition

Area IV, Robotics:
CS 223A Introduction to Robotics
CS 237A Principles of Robot Autonomy I
Select one additional course from the Areas above or from the following:
AI Methods:
CS 157 Computational Logic
CS 205L Continuous Mathematical Methods with an Emphasis on Machine Learning
CS 230 Deep Learning
CS 236 Deep Generative Models
STATS 315A Modern Applied Statistics: Learning
STATS 315B Modern Applied Statistics: Data Mining

Comp Bio:
CS 235 Computational Methods for Biomedical Image Analysis and Interpretation
CS 279 Computational Biology: Structure and Organization of Biomolecules and Cells
CS 371 Computational Biology in Four Dimensions

Information and the Web:
CS 276 Information Retrieval and Web Search
CS 224W Machine Learning with Graphs

Other:
CS 151 Logic Programming
CS 227B General Game Playing
CS 379 Interdisciplinary Topics (Offered occasionally)

Robotics and Control:
CS 327A Advanced Robotic Manipulation
CS 329 Topics in Artificial Intelligence (with advisor approval)
ENGR 205 Introduction to Control Design Techniques
MS&E 251 Introduction to Stochastic Control with Applications
MS&E 351 Dynamic Programming and Stochastic Control

Track Electives: at least three additional courses selected from the Areas and lists above, general CS electives, or the courses listed below. Students can replace one of these electives with a course found at https://cs.stanford.edu/explore.  
CS 237B Principles of Robot Autonomy II
CS 238 Decision Making under Uncertainty
CS 257 Logic and Artificial Intelligence
CS 275 Translational Bioinformatics
CS 326 Topics in Advanced Robotic Manipulation
CS 330 Deep Multi-task and Meta Learning
CS 334A Convex Optimization I
or EE 364A Convex Optimization I
CS 336 Robot Perception and Decision-Making: Optimal and Learning-based Approaches
CS 398 Computational Education
CS 428 Computation and Cognition: The Probabilistic Approach
EE 263 Introduction to Linear Dynamical Systems

EE 278 Introduction to Statistical Signal Processing
EE 364B Convex Optimization II
ECON 286 Game Theory and Economic Applications
MS&E 252 Decision Analysis I: Foundations of Decision Analysis
MS&E 352 Decision Analysis II: Professional Decision Analysis
MS&E 355 Influence Diagrams and Probabilistic Networks
PHIL 152 Computability and Logic
PSYCH 204A Human Neuroimaging Methods
PSYCH 204B Computational Neuroimaging
PSYCH 209 Neural Network Models of Cognition
STATS 200 Introduction to Statistical Inference
STATS 202 Data Mining and Analysis
STATS 205 Introduction to Nonparametric Statistics

Biocomputation Track—

The Mathematics, Science, and Engineering Fundamentals requirements are non-standard for this track. See Handbook for Undergraduate Engineering Programs for details.

Select one of the following:
CS 221 Artificial Intelligence: Principles and Techniques
CS 228 Probabilistic Graphical Models: Principles and Techniques
CS 229 Machine Learning
CS 231A Computer Vision: From 3D Reconstruction to Recognition

Select one of the following:
CS 235 Computational Methods for Biomedical Image Analysis and Interpretation
CS 270 Modeling Biomedical Systems: Ontology, Terminology, Problem Solving
CS 273A The Human Genome Source Code
CS 274 Representations and Algorithms for Computational Molecular Biology
CS 275 Translational Bioinformatics
CS 279 Computational Biology: Structure and Organization of Biomolecules and Cells

One additional course from the lists above or the following:
CS 124 From Languages to Information
CS 145 Data Management and Data Systems
CS 147 Introduction to Human-Computer Interaction Design
CS 148 Introduction to Computer Graphics and Imaging
CS 248 Interactive Computer Graphics

One course selected from the following:
CS 108 Object-Oriented Systems Design
CS 124 From Languages to Information
CS 131 Computer Vision: Foundations and Applications
CS 140 Operating Systems and Systems Programming
or CS 140E Operating systems design and implementation
CS 142 Web Applications
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 143</td>
<td>Compilers</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 144</td>
<td>Introduction to Computer Networking</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 145</td>
<td>Data Management and Data Systems</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 146</td>
<td>Introduction to Game Design and Development</td>
<td>3</td>
</tr>
<tr>
<td>CS 147</td>
<td>Introduction to Human-Computer Interaction Design</td>
<td>3-5</td>
</tr>
<tr>
<td>CS 148</td>
<td>Introduction to Computer Graphics and Imaging</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 149</td>
<td>Parallel Computing</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 151</td>
<td>Logic Programming</td>
<td>3</td>
</tr>
<tr>
<td>CS 154</td>
<td>Introduction to Automata and Complexity Theory</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 155</td>
<td>Computer and Network Security</td>
<td>3</td>
</tr>
<tr>
<td>CS 157</td>
<td>Computational Logic</td>
<td>3</td>
</tr>
<tr>
<td>or PHIL 151</td>
<td>Metalogic</td>
<td>3</td>
</tr>
<tr>
<td>CS 166</td>
<td>Data Structures</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 168</td>
<td>The Modern Algorithmic Toolbox</td>
<td></td>
</tr>
<tr>
<td>CS 190</td>
<td>Software Design Studio</td>
<td>3</td>
</tr>
<tr>
<td>CS 195</td>
<td>Supervised Undergraduate Research (4 units max)</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 205L</td>
<td>Continuous Mathematical Methods with an Emphasis on Machine Learning</td>
<td>3</td>
</tr>
<tr>
<td>CS 210A</td>
<td>Software Project Experience with Corporate Partners</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 217</td>
<td>Hardware Accelerators for Machine Learning</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 221</td>
<td>Artificial Intelligence: Principles and Techniques</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 223A</td>
<td>Introduction to Robotics</td>
<td>3</td>
</tr>
<tr>
<td>CS 224N</td>
<td>Natural Language Processing with Deep Learning</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 224S</td>
<td>Spoken Language Processing</td>
<td>2-4</td>
</tr>
<tr>
<td>CS 224U</td>
<td>Natural Language Understanding</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 224W</td>
<td>Machine Learning with Graphs</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 225A</td>
<td>Experimental Robotics</td>
<td>3</td>
</tr>
<tr>
<td>CS 227B</td>
<td>General Game Playing</td>
<td>3</td>
</tr>
<tr>
<td>CS 228</td>
<td>Probabilistic Graphical Models: Principles and Techniques</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 229</td>
<td>Machine Learning</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 229T</td>
<td>Statistical Learning Theory</td>
<td>3</td>
</tr>
<tr>
<td>CS 230</td>
<td>Deep Learning</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 231A</td>
<td>Computer Vision: From 3D Reconstruction to Recognition</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 231N</td>
<td>Convolutional Neural Networks for Visual Recognition</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 232</td>
<td>Digital Image Processing</td>
<td>3</td>
</tr>
<tr>
<td>CS 233</td>
<td>Geometric and Topological Data Analysis</td>
<td>3</td>
</tr>
<tr>
<td>CS 234</td>
<td>Reinforcement Learning</td>
<td>3</td>
</tr>
<tr>
<td>CS 235</td>
<td>Computational Methods for Biomedical Image Analysis and Interpretation</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 236</td>
<td>Deep Generative Models</td>
<td>3</td>
</tr>
<tr>
<td>CS 238</td>
<td>Decision Making under Uncertainty</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 240</td>
<td>Advanced Topics in Operating Systems</td>
<td>3</td>
</tr>
<tr>
<td>CS 242</td>
<td>Programming Languages</td>
<td>3</td>
</tr>
<tr>
<td>CS 243</td>
<td>Program Analysis and Optimizations</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 244</td>
<td>Advanced Topics in Networking</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 244B</td>
<td>Distributed Systems</td>
<td>3</td>
</tr>
<tr>
<td>CS 245</td>
<td>Principles of Data-Intensive Systems</td>
<td>3</td>
</tr>
<tr>
<td>CS 246</td>
<td>Mining Massive Data Sets</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 247</td>
<td>(Any suffix)</td>
<td></td>
</tr>
<tr>
<td>CS 248</td>
<td>Interactive Computer Graphics</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 251</td>
<td>Cryptocurrencies and blockchain technologies</td>
<td></td>
</tr>
<tr>
<td>CS 252</td>
<td>Analysis of Boolean Functions</td>
<td>3</td>
</tr>
<tr>
<td>CS 254</td>
<td>Computational Complexity</td>
<td>3</td>
</tr>
<tr>
<td>CS 255</td>
<td>Introduction to Cryptography</td>
<td>3</td>
</tr>
<tr>
<td>CS 261</td>
<td>Optimization and Algorithmic Paradigms</td>
<td>3</td>
</tr>
<tr>
<td>CS 264</td>
<td>Beyond Worst-Case Analysis</td>
<td>3</td>
</tr>
<tr>
<td>CS 265</td>
<td>Randomized Algorithms and Probabilistic Analysis</td>
<td>3</td>
</tr>
<tr>
<td>CS 269I</td>
<td>(Not Given This Year)</td>
<td>3</td>
</tr>
<tr>
<td>CS 270</td>
<td>Modeling Biomedical Systems: Ontology, Terminology, Problem Solving</td>
<td>3</td>
</tr>
<tr>
<td>CS 272</td>
<td>Introduction to Biomedical Informatics Research Methodology</td>
<td>3-5</td>
</tr>
<tr>
<td>CS 273A</td>
<td>The Human Genome Source Code</td>
<td>3</td>
</tr>
<tr>
<td>CS 273B</td>
<td>Deep Learning in Genomics and Biomedicine</td>
<td>3</td>
</tr>
<tr>
<td>CS 274</td>
<td>Representations and Algorithms for Computational Molecular Biology</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 275</td>
<td>Translational Bioinformatics</td>
<td>3</td>
</tr>
<tr>
<td>CS 276</td>
<td>Information Retrieval and Web Search</td>
<td>3</td>
</tr>
<tr>
<td>CS 278</td>
<td>Social Computing</td>
<td>3</td>
</tr>
<tr>
<td>CS 279</td>
<td>Computational Biology: Structure and Organization of Biomolecules and Cells</td>
<td>3</td>
</tr>
<tr>
<td>CS 348B</td>
<td>Computer Graphics: Image Synthesis Techniques</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 348C</td>
<td>Computer Graphics: Animation and Simulation</td>
<td>3</td>
</tr>
<tr>
<td>CS 348K</td>
<td>Visual Computing Systems</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 371</td>
<td>Computational Biology in Four Dimensions</td>
<td>3</td>
</tr>
<tr>
<td>CME 108</td>
<td>Introduction to Scientific Computing</td>
<td>3</td>
</tr>
<tr>
<td>EE 180</td>
<td>Digital Systems Architecture</td>
<td>4</td>
</tr>
<tr>
<td>EE 263</td>
<td>Introduction to Linear Dynamical Systems</td>
<td>3</td>
</tr>
<tr>
<td>EE 282</td>
<td>Computer Systems Architecture</td>
<td>3</td>
</tr>
<tr>
<td>EE 364A</td>
<td>Convex Optimization I</td>
<td>3</td>
</tr>
<tr>
<td>BIOE 101</td>
<td>Systems Biology</td>
<td>3</td>
</tr>
<tr>
<td>MS&amp;E 152</td>
<td>Introduction to Decision Analysis</td>
<td>3</td>
</tr>
<tr>
<td>MS&amp;E 252</td>
<td>Decision Analysis I: Foundations of Decision Analysis</td>
<td>3-4</td>
</tr>
<tr>
<td>STATS 206</td>
<td>Applied Multivariate Analysis</td>
<td>3</td>
</tr>
<tr>
<td>STATS 315A</td>
<td>Modern Applied Statistics: Learning</td>
<td>3</td>
</tr>
<tr>
<td>STATS 315B</td>
<td>Modern Applied Statistics: Data Mining</td>
<td>3</td>
</tr>
<tr>
<td>GENE 211</td>
<td>Genomics</td>
<td>3</td>
</tr>
<tr>
<td>One course from the following:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS 145</td>
<td>Data Management and Data Systems</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 147</td>
<td>Introduction to Human-Computer Interaction Design</td>
<td>3-5</td>
</tr>
<tr>
<td>CS 221</td>
<td>Artificial Intelligence: Principles and Techniques</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 228</td>
<td>Probabilistic Graphical Models: Principles and Techniques</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 229</td>
<td>Machine Learning</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 235</td>
<td>Computational Methods for Biomedical Image Analysis and Interpretation</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 270</td>
<td>Modeling Biomedical Systems: Ontology, Terminology, Problem Solving</td>
<td>3</td>
</tr>
</tbody>
</table>
### Computer Science Track

**Units**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 273A</td>
<td>The Human Genome Source Code</td>
<td>3</td>
</tr>
<tr>
<td>CS 273B</td>
<td>Deep Learning in Genomics and Biomedicine</td>
<td>3</td>
</tr>
<tr>
<td>CS 274</td>
<td>Representations and Algorithms for Computational Molecular Biology</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 275</td>
<td>Translational Bioinformatics</td>
<td>4</td>
</tr>
<tr>
<td>CS 279</td>
<td>Computational Biology: Structure and Organization of Biomolecules and Cells</td>
<td>3</td>
</tr>
<tr>
<td>CS 371</td>
<td>Computational Biology in Four Dimensions</td>
<td>3</td>
</tr>
<tr>
<td>CS 373</td>
<td>Statistical and Machine Learning Methods for Genomics</td>
<td>3</td>
</tr>
<tr>
<td>EE 263</td>
<td>Introduction to Linear Dynamical Systems</td>
<td>3</td>
</tr>
<tr>
<td>EE 364A</td>
<td>Convex Optimization I</td>
<td>3</td>
</tr>
<tr>
<td>MS&amp;E 152</td>
<td>Introduction to Decision Analysis</td>
<td>3-4</td>
</tr>
<tr>
<td>MS&amp;E 252</td>
<td>Decision Analysis I: Foundations of Decision Analysis</td>
<td>3-4</td>
</tr>
<tr>
<td>STATS 206</td>
<td>Applied Multivariate Analysis</td>
<td>3</td>
</tr>
<tr>
<td>STATS 315A</td>
<td>Modern Applied Statistics: Learning</td>
<td>3</td>
</tr>
<tr>
<td>STATS 315B</td>
<td>Modern Applied Statistics: Data Mining</td>
<td>3</td>
</tr>
<tr>
<td>GENE 211</td>
<td>Genomics</td>
<td>3</td>
</tr>
<tr>
<td>CHEMENG 150</td>
<td>Biochemical Engineering</td>
<td>3</td>
</tr>
<tr>
<td>CHEMENG 174</td>
<td>Environmental Microbiology I</td>
<td>3</td>
</tr>
<tr>
<td>APPPHYS 294</td>
<td>Cellular Biophysics</td>
<td>3</td>
</tr>
<tr>
<td>BIO 104</td>
<td>Advanced Molecular Biology: Epigenetics and Proteostasis</td>
<td>5</td>
</tr>
<tr>
<td>BIO 118</td>
<td>(Not Given This Year)</td>
<td>4</td>
</tr>
<tr>
<td>BIO 214</td>
<td>Advanced Cell Biology</td>
<td>4</td>
</tr>
<tr>
<td>BIO 230</td>
<td>Molecular and Cellular Immunology</td>
<td>4</td>
</tr>
<tr>
<td>CHEM 141</td>
<td>The Chemical Principles of Life I</td>
<td>4</td>
</tr>
<tr>
<td>CHEM 171</td>
<td>Physical Chemistry I</td>
<td>4</td>
</tr>
<tr>
<td>BIOC 241</td>
<td>Biological Macromolecules</td>
<td>5</td>
</tr>
<tr>
<td>BIOE 220</td>
<td>Introduction to Imaging and Image-based Human Anatomy</td>
<td>3</td>
</tr>
<tr>
<td>CHEMENG 150</td>
<td>Biochemical Engineering</td>
<td>3</td>
</tr>
<tr>
<td>CHEMENG 174</td>
<td>Environmental Microbiology I</td>
<td>3</td>
</tr>
<tr>
<td>CS 235</td>
<td>Computational Methods for Biomedical Image Analysis and Interpretation</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 274</td>
<td>Representations and Algorithms for Computational Molecular Biology</td>
<td>3-4</td>
</tr>
<tr>
<td>CS 279</td>
<td>Computational Biology: Structure and Organization of Biomolecules and Cells</td>
<td>3</td>
</tr>
<tr>
<td>CS 371</td>
<td>Computational Biology in Four Dimensions</td>
<td>3</td>
</tr>
<tr>
<td>ME 281</td>
<td>Biomechanics of Movement</td>
<td>3</td>
</tr>
<tr>
<td>APPPHYS 294</td>
<td>Cellular Biophysics</td>
<td>3</td>
</tr>
<tr>
<td>BIO 104</td>
<td>Advanced Molecular Biology: Epigenetics and Proteostasis</td>
<td>5</td>
</tr>
<tr>
<td>BIO 112</td>
<td>Human Physiology</td>
<td>4</td>
</tr>
<tr>
<td>BIO 118</td>
<td>(Not Given This Year)</td>
<td>4</td>
</tr>
<tr>
<td>BIO 158</td>
<td>Developmental Neurobiology</td>
<td>4</td>
</tr>
<tr>
<td>BIO 183</td>
<td>Theoretical Population Genetics</td>
<td>3</td>
</tr>
<tr>
<td>BIO 214</td>
<td>Advanced Cell Biology</td>
<td>4</td>
</tr>
<tr>
<td>BIO 230</td>
<td>Molecular and Cellular Immunology</td>
<td>4</td>
</tr>
<tr>
<td>CHEM 171</td>
<td>Physical Chemistry I</td>
<td>4</td>
</tr>
<tr>
<td>BIOC 241</td>
<td>Biological Macromolecules</td>
<td>5</td>
</tr>
<tr>
<td>DBIO 210</td>
<td>Developmental Biology</td>
<td>4</td>
</tr>
<tr>
<td>GENE 211</td>
<td>Genomics</td>
<td>3</td>
</tr>
<tr>
<td>SURG 101</td>
<td>Regional Study of Human Structure</td>
<td>3</td>
</tr>
</tbody>
</table>

#### Computer Engineering Track—

**For this track there is a 10 unit minimum for ENGR Fundamentals and a 29 unit minimum for Depth (for track and elective courses)**

**Units**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE 108 &amp; EE 180</td>
<td>Digital System Design and Digital Systems Architecture</td>
<td>8</td>
</tr>
<tr>
<td>EE 101A</td>
<td>Circuits I</td>
<td>3</td>
</tr>
<tr>
<td>EE 101B</td>
<td>Circuits II</td>
<td>3</td>
</tr>
<tr>
<td>EE 102A</td>
<td>Signal Processing and Linear Systems I</td>
<td>3</td>
</tr>
<tr>
<td>EE 102B</td>
<td>Signal Processing and Linear Systems II</td>
<td>3</td>
</tr>
<tr>
<td>EE 144</td>
<td>Introduction to Computer Networking</td>
<td>3</td>
</tr>
<tr>
<td>CS 144</td>
<td>Parallel Computing</td>
<td>3</td>
</tr>
<tr>
<td>CS 190</td>
<td>Software Design Studio</td>
<td>3</td>
</tr>
<tr>
<td>CS 217</td>
<td>Hardware Accelerators for Machine Learning</td>
<td>3</td>
</tr>
<tr>
<td>CS 244</td>
<td>Advanced Topics in Networking</td>
<td>3</td>
</tr>
<tr>
<td>EE 273</td>
<td>Digital Systems Engineering</td>
<td>3</td>
</tr>
<tr>
<td>EE 282</td>
<td>Computer Systems Architecture</td>
<td>3</td>
</tr>
</tbody>
</table>

#### 1) Digital Systems Concentration

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 140</td>
<td>Operating Systems and Systems Programming</td>
<td>3</td>
</tr>
<tr>
<td>or CS 140E  &amp; CS 140E or CS</td>
<td>Operating Systems and Systems Programming (if not counted above)</td>
<td>3</td>
</tr>
<tr>
<td>EE 109</td>
<td>Digital Systems Design Lab</td>
<td>3</td>
</tr>
</tbody>
</table>

#### 2) Robotics and Mechatronics Concentration

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 205L</td>
<td>Continuous Mathematical Methods with an Emphasis on Machine Learning</td>
<td>3</td>
</tr>
<tr>
<td>CS 223A</td>
<td>Introduction to Robotics</td>
<td>3</td>
</tr>
<tr>
<td>ME 210</td>
<td>Introduction to Mechatronics</td>
<td>3</td>
</tr>
<tr>
<td>ENGR 105</td>
<td>Feedback Control Design</td>
<td>3</td>
</tr>
</tbody>
</table>

#### 3) Networking Concentration

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 140 &amp; CS 144</td>
<td>Operating Systems and Systems Programming and Introduction to Computer Networking (CS 140E can substitute for CS 140)</td>
<td>3</td>
</tr>
</tbody>
</table>

#### 4) Systems Concentration

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE 179</td>
<td>Analog and Digital Communication Systems</td>
<td>3</td>
</tr>
</tbody>
</table>
## Graphics Track—


Select one of the following: 3-5 units

- CS 205L | Continuous Mathematical Methods with an Emphasis on Machine Learning |
- CME 104 | Linear Algebra and Partial Differential Equations for Engineers (Note: students taking CME 104 are also required to take its prerequisite course, CME 102) |
- CME 108 | Introduction to Scientific Computing |
- MATH 52 | Integral Calculus of Several Variables |
- MATH 113 | Linear Algebra and Matrix Theory |

Select two of the following: 6-8 units

- CS 146 | Introduction to Game Design and Development |
- CS 231A | Computer Vision: From 3D Reconstruction to Recognition |
- or CS 131 | Computer Vision: Foundations and Applications |
- CS 233 | Geometric and Topological Data Analysis |
- CS 268 | Geometric Algorithms |
- CS 348C | Computer Graphics: Animation and Simulation |
- CS 348E | Character Animation: Modeling, Simulation, and Control of Human Motion |
- CS 348K | Visual Computing Systems |
- CS 448 | Topics in Computer Graphics |

Track Electives: at least two additional courses from the lists above, the general CS electives list, or the courses listed below. Students can replace one of these electives with a course found at: https://cs.stanford.edu/explore:

- ARTSTUDI 160 | Intro to Digital / Physical Design |
- ARTSTUDI 162 | Embodied Interfaces |
- ARTSTUDI 163 | Drawing with Code |
- ARTSTUDI 164 | DESIGN IN PUBLIC SPACES |
- ARTSTUDI 165 | Social Media and Performative Practices |
- ARTSTUDI 168 | Data as Material |
- ARTSTUDI 264 | Advanced Interaction Design |
- ARTSTUDI 266 | Sculptural Screens / Malleable Media |
- ARTSTUDI 267 | Emerging Technology Studio |
- SYMSYS 245 | Cognition in Interaction Design |

### Psychophysiology

- PSYCH 30 | Introduction to Perception |
- PSYCH 35 | Minds and Machines |
- PSYCH 45 | Introduction to Learning and Memory |
- PSYCH 50 | Introduction to Cognitive Neuroscience |
- PSYCH 60 | Introduction to Developmental Psychology |
- PSYCH 70 | Self and Society: Introduction to Social Psychology |
- PSYCH 75 | Introduction to Cultural Psychology |
- PSYCH 80 | Introduction to Personality and Affective Science |
- PSYCH 90 | INTRODUCTION TO CLINICAL PSYCHOLOGY: A NEUROSCIENCE PERSPECTIVE |
- PSYCH 95 | Introduction to Abnormal Psychology |

## Human-Computer Interaction Track—

| CS 147 | Introduction to Human-Computer Interaction Design | 4 |
| CS 247 | (Any suffix) | 4 |

Any three of the following: 6-8 units

- ARTSTUDI 160 | Intro to Digital / Physical Design |
- ARTSTUDI 162 | Embodied Interfaces |
- ARTSTUDI 163 | Drawing with Code |
- ARTSTUDI 164 | DESIGN IN PUBLIC SPACES |
- ARTSTUDI 165 | Social Media and Performative Practices |
- ARTSTUDI 168 | Data as Material |
- ARTSTUDI 264 | Advanced Interaction Design |
- ARTSTUDI 266 | Sculptural Screens / Malleable Media |
- ARTSTUDI 267 | Emerging Technology Studio |
- SYMSYS 245 | Cognition in Interaction Design |

### Psychophysiology

- PSYCH 30 | Introduction to Perception |
- PSYCH 35 | Minds and Machines |
- PSYCH 45 | Introduction to Learning and Memory |
- PSYCH 50 | Introduction to Cognitive Neuroscience |
- PSYCH 60 | Introduction to Developmental Psychology |
- PSYCH 70 | Self and Society: Introduction to Social Psychology |
- PSYCH 75 | Introduction to Cultural Psychology |
- PSYCH 80 | Introduction to Personality and Affective Science |
- PSYCH 90 | INTRODUCTION TO CLINICAL PSYCHOLOGY: A NEUROSCIENCE PERSPECTIVE |
- PSYCH 95 | Introduction to Abnormal Psychology |

### Empirical Methods

- PSYCH 30 | Introduction to Perception |
- PSYCH 35 | Minds and Machines |
- PSYCH 45 | Introduction to Learning and Memory |
- PSYCH 50 | Introduction to Cognitive Neuroscience |
- PSYCH 60 | Introduction to Developmental Psychology |
- PSYCH 70 | Self and Society: Introduction to Social Psychology |
- PSYCH 75 | Introduction to Cultural Psychology |
- PSYCH 80 | Introduction to Personality and Affective Science |
- PSYCH 90 | INTRODUCTION TO CLINICAL PSYCHOLOGY: A NEUROSCIENCE PERSPECTIVE |
- PSYCH 95 | Introduction to Abnormal Psychology |
COMM 314 Ethnographic Methods
MS&E 125 Introduction to Applied Statistics
PSYCH 251 Experimental Methods
PSYCH 252 Statistical Methods for Behavioral and Social Sciences
PSYCH 253 Advanced Statistical Modeling
STATS 203 Introduction to Regression Models and Analysis of Variance
EDUC 191 Introduction to Survey Research
HUMBIO 82A Qualitative Research Methodology

ME Design-
ME 101 Visual Thinking
ME 115A Introduction to Human Values in Design
ME 203 Design and Manufacturing
ME 210 Introduction to Mechatronics
ME 216A Advanced Product Design: Needfinding

Learning Design + Tech-
EDUC 236 Beyond Bits and Atoms: Designing Technological Tools
EDUC 281 Technology for Learners
EDUC 239 Educating Young STEM Thinkers
EDUC 338 Innovations in Education
EDUC 342 Child Development and New Technologies
MS&E 185 Global Work
MS&E 330 Law, Bias, & Algorithms

Computer Music-
MUSIC 220A Fundamentals of Computer-Generated Sound
MUSIC 220B Compositional Algorithms, Psychoacoustics, and Computational Music
MUSIC 220C Research Seminar in Computer-Generated Music
MUSIC 250A Physical Interaction Design for Music
MUSIC 256A Music, Computing, Design I: The Art of Design

Optional Elective

Information Track—

CS 124 From Languages to Information 4
CS 145 Data Management and Data Systems 4
Two courses, from different areas: 6-9

1) Information-based AI applications
CS 224N Natural Language Processing with Deep Learning
CS 224S Spoken Language Processing
CS 229 Machine Learning
CS 233 Geometric and Topological Data Analysis
CS 234 Reinforcement Learning

2) Database and Information Systems
CS 140 Operating Systems and Systems Programming
or CS 140E Operating systems design and implementation
CS 142 Web Applications
CS 151 Logic Programming

CS 245 Principles of Data-Intensive Systems
CS 246 Mining Massive Data Sets
CS 341 Project in Mining Massive Data Sets
CS 345 (Offered occasionally)

3) Information Systems in Biology
CS 235 Computational Methods for Biomedical Image Analysis and Interpretation
CS 270 Modeling Biomedical Systems: Ontology, Terminology, Problem Solving
CS 274 Representations and Algorithms for Computational Molecular Biology

4) Information Systems on the Web
CS 224W Machine Learning with Graphs
CS 276 Information Retrieval and Web Search
At least three additional courses from the above areas or the general CS electives list. Students can replace one of these electives with a course found at https://cs.stanford.edu/explore

Systems Track—

CS 140 Operating Systems and Systems Programming 4
or CS 140E Operating systems design and implementation
Select one of the following: 3-4
CS 143 Compilers
EE 180 Digital Systems Architecture

Two additional courses from the list above or the following: 6-8
CS 144 Introduction to Computer Networking
CS 145 Data Management and Data Systems
CS 149 Parallel Computing
CS 155 Computer and Network Security
CS 190 Software Design Studio
CS 217 Hardware Accelerators for Machine Learning
CS 240 Advanced Topics in Operating Systems
CS 242 Programming Languages
CS 243 Program Analysis and Optimizations
CS 244 Advanced Topics in Networking
CS 245 Principles of Data-Intensive Systems
EE 271 Introduction to VLSI Systems
EE 282 Computer Systems Architecture

Track Electives: at least three additional courses selected from the list above, the general CS electives list, or the courses listed below. Students can replace one of these electives with a course found at: https://cs.stanford.edu/explore

CS 241 Embedded Systems Workshop
CS 269Q Elements of Quantum Computer Programming
CS 316 Advanced Multi-Core Systems
CS 341 Project in Mining Massive Data Sets
CS 344 Topics in Computer Networks (3 or more units, any suffix)
CS 345 (Advanced Topics in Database Systems - 3 or more units, any suffix. Offered occasionally.)
CS 349 Topics in Programming Systems (with permission of undergraduate advisor)
CS 448 Topics in Computer Graphics
**Computer Science**

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE 108</td>
<td>Digital System Design</td>
</tr>
<tr>
<td>EE 382C</td>
<td>Interconnection Networks</td>
</tr>
<tr>
<td>EE 384A</td>
<td>Internet Routing Protocols and Standards</td>
</tr>
<tr>
<td>EE 384C</td>
<td>Wireless Local and Wide Area Networks</td>
</tr>
<tr>
<td>EE 384S</td>
<td>Performance Engineering of Computer Systems &amp; Networks</td>
</tr>
</tbody>
</table>

**Theory Track**

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 154</td>
<td>Introduction to Automata and Complexity Theory</td>
<td>4</td>
</tr>
</tbody>
</table>

Select one of the following: 3

- CS 168: The Modern Algorithmic Toolbox
- CS 255: Introduction to Cryptography
- CS 261: Optimization and Algorithmic Paradigms
- CS 264: Beyond Worst-Case Analysis
- CS 265: Randomized Algorithms and Probabilistic Analysis
- CS 268: Geometric Algorithms

Two additional courses from the list above or the following: 6-8

- CS 143: Compilers
- CS 151: Logic Programming
- CS 155: Computer and Network Security
- CS 157: Computational Logic or PHIL 151: Metalogic
- CS 166: Data Structures
- CS 205L: Continuous Mathematical Methods with an Emphasis on Machine Learning
- CS 228: Probabilistic Graphical Models: Principles and Techniques
- CS 233: Geometric and Topological Data Analysis
- CS 235: Computational Methods for Biomedical Image Analysis and Interpretation
- CS 236: Deep Generative Models
- CS 242: Programming Languages
- CS 250: Algebraic Error Correcting Codes
- CS 251: Cryptocurrencies and blockchain technologies
- CS 252: Analysis of Boolean Functions
- CS 254: Computational Complexity
- CS 259: (With permission of undergraduate advisor. Course offered occasionally.)
- CS 269I: (Not Given This Year)
- CS 352: Pseudo-Randomness
- CS 353: The Practice of Theory Research
- CS 354: Topics in Intractability: Unfulfilled Algorithmic Fantasies (Not given this year)
- CS 355: Advanced Topics in Cryptography (Not given this year)
- CS 357: Advanced Topics in Formal Methods (Not given this year)
- CS 358: Topics in Programming Language Theory
- CS 359: Topics in the Theory of Computation (with permission of undergraduate advisor)
- CS 369: Topics in Analysis of Algorithms (with permission of undergraduate advisor)
- MS&E 310: Linear Programming

**Track Electives:** at least three additional courses from the lists above, the general CS electives list, or the courses listed below. Students can replace one of these electives with a course found at: [https://cs.stanford.edu/explain](https://cs.stanford.edu/explain).

- CS 254B: Computational Complexity II
- CS 269G: Almost Linear Time Graph Algorithms
- CME 302: Numerical Linear Algebra
- CME 305: Discrete Mathematics and Algorithms
- PHIL 152: Computability and Logic

**Unspecialized Track**

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 154</td>
<td>Introduction to Automata and Complexity Theory</td>
<td>4</td>
</tr>
</tbody>
</table>

Select one of the following: 4

- CS 140: Operating Systems and Systems Programming
- CS 140E: Operating systems design and implementation
- CS 143: Compilers

One additional course from the list above or the following: 3-4

- CS 144: Introduction to Computer Networking
- CS 155: Computer and Network Security
- CS 190: Software Design Studio
- CS 242: Programming Languages
- CS 244: Advanced Topics in Networking
- EE 180: Digital Systems Architecture

Select one of the following: 3-4

- CS 221: Artificial Intelligence: Principles and Techniques
- CS 223A: Introduction to Robotics
- CS 228: Probabilistic Graphical Models: Principles and Techniques
- CS 229: Machine Learning
- CS 231A: Computer Vision: From 3D Reconstruction to Recognition

Select one of the following: 3-4

- CS 145: Data Management and Data Systems
- CS 147: Introduction to Human-Computer Interaction Design
- CS 148: Introduction to Computer Graphics and Imaging
- CS 235: Computational Methods for Biomedical Image Analysis and Interpretation
- CS 248: Interactive Computer Graphics

At least two courses from the general CS electives list 4

**Individually Designed Track**

Students may propose an individually designed track. Proposals should include a minimum of 25 units and seven courses, at least four of which must be CS courses numbered 100 or above. See Handbook for Undergraduate Engineering Programs for further information.

For additional information and sample programs see the Handbook for Undergraduate Engineering Programs (UGHB) ([http://ughb.stanford.edu](http://ughb.stanford.edu))

**Footnotes**

1. MATH 19, MATH 20, and MATH 21, or MATH 41 and MATH 42, or AP Calculus Credit may be used as long as at least 26 MATH units are taken. AP Calculus Credit must be approved by the School of Engineering.
The math electives list consists of: MATH 51, MATH 52, MATH 53, MATH 104, MATH 107, MATH 109, MATH 110, MATH 113; CS 157, CS 205L, PHIL 151; CME 100, CME 102, CME 103 (or EE 103), CME 104. Restrictions: CS 157 and PHIL 151 may not be used in combination to satisfy the math electives requirement. Students who have taken both MATH 51 and MATH 52 may not count CME 100 as an elective. Courses counted as math electives cannot also count as CS electives, and vice versa.

The science elective may be any course of 3 or more units from the School of Engineering Science list (Fig. 4-2 in the UGHB), PSYCH 30, or AP Chemistry Credit. Either of the PHYSICS sequences 61/63 or 21/23 may be substituted for 41/43 as long as at least 11 science units are taken. AP Chemistry Credit and AP Physics Credit must be approved by the School of Engineering.


CS 205L is strongly recommended in this list for the Graphics track. Students taking CME 104 Linear Algebra and Partial Differential Equations for Engineers are also required to take its prerequisite, CME 102 Ordinary Differential Equations for Engineers.

Independent study projects (CS 191 Senior Project or CS 191W Writing Intensive Senior Project) require faculty sponsorship and must be approved by the adviser, faculty sponsor, and the CS senior project adviser (Patrick Young). A signed approval form, along with a brief description of the proposed project, should be filed the quarter before work on the project is begun. Further details can be found in the Handbook for Undergraduate Engineering Programs (UGHB) (http://ughb.stanford.edu).

A course may only be counted towards one requirement; it may not be double-counted. All courses taken for the major must be taken for a letter grade if that option is offered by the instructor. Minimum Combined GPA for all courses in Engineering Fundamentals and Depth is 2.0.

Honors Program in Computer Science

The Department of Computer Science (CS) offers an honors program for undergraduates whose academic records and personal initiative indicate that they have the necessary skills to undertake high-quality research in computer science. Admission to the program is by application only. To apply for the honors program, students must be majoring in Computer Science, have a grade point average (GPA) of at least 3.6 in courses that count toward the major, and achieve senior standing (135 or more units) by the end of the academic year in which they apply. Coterminal master’s students are eligible to apply as long as they have not already received their undergraduate degree. Beyond these requirements, students who apply for the honors program must find a Computer Science faculty member who agrees to serve as the thesis adviser for the project. Thesis advisers must be members of Stanford’s Academic Council.

Students who meet the eligibility requirements and wish to be considered for the honors program must submit a written application to the CS undergraduate program office by May 1 of the year preceding the honors work. The application must include a letter describing the research project, a letter of endorsement from the faculty sponsor, and a transcript of courses taken at Stanford. Each year, a faculty review committee selects the successful candidates for honors from the pool of qualified applicants.

In order to receive departmental honors, students admitted to the honors program must, in addition to satisfying the standard requirements for the undergraduate degree, do the following:

1. Complete at least 9 units of CS 191 or CS 191W under the direction of their project sponsor.
2. Attend a weekly honors seminar Winter and Spring quarters.
3. Complete an honors thesis deemed acceptable by the thesis adviser and at least one additional faculty member.
4. Present the thesis at a public colloquium sponsored by the department.
5. Maintain the 3.6 GPA required for admission to the honors program.

Guide to Choosing Introductory Courses

Students arriving at Stanford have widely differing backgrounds and goals, but most find that the ability to use computers effectively is beneficial to their education. The department offers many introductory courses to meet the needs of these students.

For students whose principal interest is an exposure to the fundamental ideas behind computer science and programming, CS 101 or CS 105 are the most appropriate courses. They are intended for students in nontechnical disciplines who expect to make some use of computers, but who do not expect to go on to more advanced courses. CS 101 and CS 105 meet the Ways of Thinking Ways of Doing breadth requirements in Formal Reasoning and include an introduction to programming and the use of modern Internet-based technologies. Students interested in learning to use the computer should consider CS 1C, Introduction to Computing at Stanford.

Students who intend to pursue a serious course of study in computer science may enter the program at a variety of levels, depending on their background. Students with little prior experience or those who wish to take more time to study the fundamentals of programming should take CS 106A followed by CS 106B. Students in CS 106A need not have prior programming experience. Students with significant prior exposure to programming or those who want an intensive introduction to the field should take CS 106X or may start directly in CS 106B. CS 106A uses Python as its programming language; CS 106B and X use C++. No prior knowledge of these languages is assumed, and the prior programming experience required for CS 106B or X may be in any language. In all cases, students are encouraged to discuss their background with the instructors responsible for these courses.

After the introductory sequence, Computer Science majors and those who need a significant background in computer science for related majors in engineering should take CS 103, CS 107 and CS 110. CS 103 offers an introduction to the mathematical and theoretical foundations of computer science. CS 107 exposes students to a variety of programming concepts that illustrate critical strategies used in systems development; CS 110 builds on this material, focusing on the development of larger-scale software making use of systems and networking abstractions.

In summary:

For exposure:

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 1C</td>
<td>Introduction to Computing at Stanford</td>
</tr>
</tbody>
</table>

For nontechnical use:

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 101</td>
<td>Introduction to Computing Principles</td>
</tr>
<tr>
<td>or CS 105</td>
<td>Introduction to Computers</td>
</tr>
</tbody>
</table>

For scientific use:

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 106A</td>
<td>Programming Methodology</td>
</tr>
</tbody>
</table>
For a technical introduction:

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 106A</td>
<td>Programming Methodology</td>
</tr>
</tbody>
</table>

For significant use:

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 106A &amp; CS 106B or CS 106X</td>
<td>Programming Methodology and Programming Abstractions</td>
</tr>
<tr>
<td>CS 103</td>
<td>Mathematical Foundations of Computing</td>
</tr>
<tr>
<td>CS 107</td>
<td>Computer Organization and Systems</td>
</tr>
<tr>
<td>CS 110</td>
<td>Principles of Computer Systems</td>
</tr>
</tbody>
</table>

**Overseas Studies Courses in Computer Science**

For course descriptions and additional offerings, see the listings in the Stanford Bulletin’s ExploreCourses web site (http://explorecourses.stanford.edu) or the Bing Overseas Studies web site (http://bosp.stanford.edu). Students should consult their department or program’s student services office for applicability of Overseas Studies courses to a major or minor program.

**Joint Major Program: Computer Science and a Humanities Major**

The joint major program (JMP) was discontinued at the end of the academic year 2018-19. Students may no longer declare this program. All students with declared joint majors are permitted to complete their degree; faculty and departments are committed to providing the necessary advising support.

See the "Joint Major Program (http://exploredegrees.stanford.edu/undergraduatedeegreesandprograms/#jointmajortext)" section of this bulletin for a description of University requirements for the JMP. See also the Undergraduate Advising and Research JMP (https://majors.stanford.edu/more-ways-explore/joint-majors-csx) web site and its associated FAQs.

Students completing the JMP receive a B.A.S. (Bachelor of Arts and Science).

**Mission**

The Joint Major provides a unique opportunity to gain mastery in two disciplines: Computer Science and a selected humanities field. Unlike the double major or dual major, the Joint Major emphasizes integration of the two fields through a cohesive, transdisciplinary course of study and integrated capstone experience. The Joint Major not only blends the intellectual traditions of two Stanford departments; it does so in a way that reduces the total unit requirement for each major.

**Computer Science Major Requirements in the Joint Major Program**

(See the respective humanities department Joint Major Program section of this bulletin for details on humanities major requirements.)

The CS requirements for the Joint Major follow the CS requirements for the CS-BS degree with the following exceptions:

1. Two of the depth electives are waived. The waived depth electives are listed below for each CS track.
2. The Senior Project is fulfilled with a joint capstone project. The student enrolls in CS191 or 191W (3 units) during the senior year. Depending on the X department, enrollment in an additional Humanities capstone course may also be required. But, at a minimum, 3 units of CS191 or 191W must be completed.
3. There is no double-counting of units between majors. If a course is required for both the CS and Humanities majors, the student will work with one of the departments to identify an additional course – one which will benefit the academic plan - to apply to that major’s total units requirement.
4. For CS, WIM can be satisfied with CS181W or CS191W.

**Depth Electives for CS Tracks for students completing a Joint Major:**

- **Artificial Intelligence Track:**
  - One Track Elective (rather than three).
- **Biocomputation Track:**
  - One course from Note 3 of the Department Program Sheet, plus one course from Note 4 of the Program Sheet.
- **Computer Engineering Track:**
  - EE 108A and 108B
  - One of the following: EE 101A, 101B, 102A, 102B
  - Satisfy the requirements of one of the following concentrations:
    1. Digital Systems Concentration: CS 140 or 143; EE 109, 271; plus one of CS 140 or 143 (if not counted above), 144, 149, 240E, 244: EE 273, 282
    2. Robotics and Mechatronics Concentration: CS 205A, 223A, ME 210; ENGR 105
    3. Networking Concentration: CS 140, 144; plus two of the following, CS 240, 240E, 244, 244B, 244E, 249A, 249B, EE 179, EE 276
- **Graphics Track:**
  - No Track Electives required (rather than two)
- **HCI Track:**
  - No Interdisciplinary HCI Electives required
- **Information Track:**
  - One Track Elective (rather than three)
- **Systems Track:**
  - One Track Elective (rather than three)
- **Theory Track:**
  - One Track Elective (rather than three)
- **Unspecialized Track:**
  - No Track Electives required (rather than two)
- **Individually Designed Track:**
  - Proposals should include a minimum of five (rather than seven) courses, at least four of which must be CS courses numbered 100 or above.

**Dropping a Joint Major Program**

To drop the joint major, students must submit the Declaration or Change of Undergraduate Major, Minor, Honors, or Degree Program (https://stanford.box.com/change-UG-program). Students may also consult the Student Services Center (http://studentservicescenter.stanford.edu) with questions concerning dropping the joint major.

**Transcript and Diploma**

Students completing a joint major graduate with a B.A.S. degree. The two majors are identified on one diploma separated by a hyphen. There will be a notation indicating that the student has completed a “Joint Major.”

---

**For a technical introduction:**

- **CS 106A** Programming Methodology

**For significant use:**

- **CS 106A & CS 106B or CS 106X** Programming Methodology and Programming Abstractions
- **CS 103** Mathematical Foundations of Computing
- **CS 107** Computer Organization and Systems
- **CS 110** Principles of Computer Systems

**Overseas Studies Courses in Computer Science**

For course descriptions and additional offerings, see the listings in the Stanford Bulletin’s ExploreCourses web site (http://explorecourses.stanford.edu) or the Bing Overseas Studies web site (http://bosp.stanford.edu). Students should consult their department or program’s student services office for applicability of Overseas Studies courses to a major or minor program.

**Joint Major Program: Computer Science and a Humanities Major**

The joint major program (JMP) was discontinued at the end of the academic year 2018-19. Students may no longer declare this program. All students with declared joint majors are permitted to complete their degree; faculty and departments are committed to providing the necessary advising support.

See the "Joint Major Program (http://exploredegrees.stanford.edu/undergraduatedeegreesandprograms/#jointmajortext)" section of this bulletin for a description of University requirements for the JMP. See also the Undergraduate Advising and Research JMP (https://majors.stanford.edu/more-ways-explore/joint-majors-csx) web site and its associated FAQs.

Students completing the JMP receive a B.A.S. (Bachelor of Arts and Science).

**Mission**

The Joint Major provides a unique opportunity to gain mastery in two disciplines: Computer Science and a selected humanities field. Unlike the double major or dual major, the Joint Major emphasizes integration of the two fields through a cohesive, transdisciplinary course of study and integrated capstone experience. The Joint Major not only blends the intellectual traditions of two Stanford departments; it does so in a way that reduces the total unit requirement for each major.

**Computer Science Major Requirements in the Joint Major Program**

(See the respective humanities department Joint Major Program section of this bulletin for details on humanities major requirements.)

The CS requirements for the Joint Major follow the CS requirements for the CS-BS degree with the following exceptions:

1. Two of the depth electives are waived. The waived depth electives are listed below for each CS track.
2. The Senior Project is fulfilled with a joint capstone project. The student enrolls in CS191 or 191W (3 units) during the senior year. Depending on the X department, enrollment in an additional Humanities capstone course may also be required. But, at a minimum, 3 units of CS191 or 191W must be completed.
3. There is no double-counting of units between majors. If a course is required for both the CS and Humanities majors, the student will work with one of the departments to identify an additional course – one which will benefit the academic plan - to apply to that major’s total units requirement.
4. For CS, WIM can be satisfied with CS181W or CS191W.

**Depth Electives for CS Tracks for students completing a Joint Major:**

- **Artificial Intelligence Track:**
  - One Track Elective (rather than three).
- **Biocomputation Track:**
  - One course from Note 3 of the Department Program Sheet, plus one course from Note 4 of the Program Sheet.
- **Computer Engineering Track:**
  - EE 108A and 108B
  - One of the following: EE 101A, 101B, 102A, 102B
  - Satisfy the requirements of one of the following concentrations:
    1. Digital Systems Concentration: CS 140 or 143; EE 109, 271; plus one of CS 140 or 143 (if not counted above), 144, 149, 240E, 244: EE 273, 282
    2. Robotics and Mechatronics Concentration: CS 205A, 223A, ME 210; ENGR 105
    3. Networking Concentration: CS 140, 144; plus two of the following, CS 240, 240E, 244, 244B, 244E, 249A, 249B, EE 179, EE 276
- **Graphics Track:**
  - No Track Electives required (rather than two)
- **HCI Track:**
  - No Interdisciplinary HCI Electives required
- **Information Track:**
  - One Track Elective (rather than three)
- **Systems Track:**
  - One Track Elective (rather than three)
- **Theory Track:**
  - One Track Elective (rather than three)
- **Unspecialized Track:**
  - No Track Electives required (rather than two)
- **Individually Designed Track:**
  - Proposals should include a minimum of five (rather than seven) courses, at least four of which must be CS courses numbered 100 or above.

**Dropping a Joint Major Program**

To drop the joint major, students must submit the Declaration or Change of Undergraduate Major, Minor, Honors, or Degree Program (https://stanford.box.com/change-UG-program). Students may also consult the Student Services Center (http://studentservicescenter.stanford.edu) with questions concerning dropping the joint major.

**Transcript and Diploma**

Students completing a joint major graduate with a B.A.S. degree. The two majors are identified on one diploma separated by a hyphen. There will be a notation indicating that the student has completed a “Joint Major.”
Computer Science (CS) Minor

The following core courses fulfill the minor requirements. Prerequisites include the standard mathematics sequence through MATH 51 (or CME 100).

<table>
<thead>
<tr>
<th>Units</th>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>CS 106B</td>
<td>Programming Abstractions</td>
</tr>
<tr>
<td></td>
<td>or CS 106X</td>
<td>Programming Abstractions</td>
</tr>
<tr>
<td>5</td>
<td>CS 103</td>
<td>Mathematical Foundations of Computing</td>
</tr>
<tr>
<td>5</td>
<td>CS 107</td>
<td>Computer Organization and Systems</td>
</tr>
<tr>
<td></td>
<td>or CS 107E</td>
<td>Computer Systems from the Ground Up</td>
</tr>
<tr>
<td>5</td>
<td>CS 109</td>
<td>Introduction to Probability for Computer Scientists</td>
</tr>
</tbody>
</table>

Electives (choose two courses from different areas):

- Artificial Intelligence—
  - CS 124 From Languages to Information (4)
  - CS 221 Artificial Intelligence: Principles and Techniques (4)
  - CS 229 Machine Learning (3-4)

- Human-Computer Interaction—
  - CS 147 Introduction to Human-Computer Interaction Design (4)

- Software—
  - CS 108 Object-Oriented Systems Design (4)
  - CS 110 Principles of Computer Systems (5)

- Systems—
  - CS 140 Operating Systems and Systems Programming (4)
  - or CS 140E Operating systems design and implementation (4)
  - CS 143 Compilers (4)
  - CS 144 Introduction to Computer Networking (4)
  - CS 145 Data Management and Data Systems (4)
  - CS 148 Introduction to Computer Graphics and Imaging (4)

- Theory—
  - CS 154 Introduction to Automata and Complexity (4)
  - Theory (4)
  - CS 157 Computational Logic (3)
  - CS 161 Design and Analysis of Algorithms (5)

Note: for students with no programming background and who begin with CS 106A, the minor consists of seven courses.

Master of Science in Computer Science

In general, the M.S. degree in Computer Science is intended as a terminal professional degree and does not lead to the Ph.D. degree. Most students planning to obtain the Ph.D. degree should apply directly for admission to the Ph.D. program. Some students, however, may wish to complete the master's program before deciding whether to pursue the Ph.D. To give such students a greater opportunity to become familiar with research, the department has a program leading to a master's degree with distinction in research. This program is described in more detail below.

Admission

Applications to the M.S. program and all supporting documents must be submitted and received online by the published deadline. Information on admission requirements (http://cs.stanford.edu/admissions) is available on the department's website; see also the department's deadlines page (https://cs.stanford.edu/admissions/deadlines). Exceptions are made for applicants who are already students at Stanford and are applying to the coterminal program (https://cs.stanford.edu/admissions/current-stanford-students/coterminal-program).

University Coterminal Requirements

Coterminal master's degree candidates are expected to complete all master's degree requirements as described in this bulletin. University requirements for the coterminal master's degree are described in the “Coterminal Master's Program (http://exploredegrees.stanford.edu/cotermdegrees)” section. University requirements for the master's degree are described in the “Graduate Degrees (http://exploredegrees.stanford.edu/graduatedegrees/#masterstext)” section of this bulletin.

After accepting admission to this coterminal master's degree program, students may request transfer of courses from the undergraduate to the graduate career to satisfy requirements for the master's degree. Transfer of courses to the graduate career requires review and approval of both the undergraduate and graduate programs on a case by case basis.

In this master's program, courses taken during or after the first quarter of the sophomore year are eligible for consideration for transfer to the graduate career; the timing of the first graduate quarter is not a factor. No courses taken prior to the first quarter of the sophomore year may be used to meet master's degree requirements.

Course transfers are not possible after the bachelor's degree has been conferred.

The University requires that the graduate adviser be assigned in the student's first graduate quarter even though the undergraduate career may still be open. The University also requires that the Master's Degree Program Proposal be completed by the student and approved by the department by the end of the student's first graduate quarter.

Requirements

A candidate is required to complete a program of 45 units. At least 36 of these must be graded units, passed with a grade point average (GPA) of 3.0 (B) or better. The 45 units may include no more than 10 units of courses from those listed below in Requirement 1. Thus, students needing to take more than two of the courses listed in Requirement 1 actually complete more than 45 units of course work in the program. Only well-prepared students may expect to finish the program in one year; most students complete the program in six quarters. Students hoping to complete the program with 45 units should already have a substantial background in computer science, including course work or experience equivalent to all of Requirement 1 and some prior course work related to their specialization area.

Requirement 1: Foundations—

Students must complete the following courses, or waive out of them by providing evidence to their advisers that similar or more advanced courses have been taken, either at Stanford or another institution (total units used to satisfy foundations requirement may not exceed 10):

- Logic, Automata, and Computability
  - CS 103 Mathematical Foundations of Computing
- Probability
  - CS 109 Introduction to Probability for Computer Scientists
Requirement 2: Significant Software Implementation—
Students must complete at least one course designated as having a significant software implementation component. The list of such courses includes:

- CS 140 Operating Systems and Systems Programming 3-4
- or CS 140E Operating systems design and implementation 3-4
- CS 143 Compilers 3-4
- CS 144 Introduction to Computer Networking 3-4
- CS 145 Data Management and Data Systems 3-4
- CS 148 Introduction to Computer Graphics and Imaging 3-4
- CS 190 Software Design Studio 3
- CS 210B Software Project Experience with Corporate Partners 3-4
- CS 221 Artificial Intelligence: Principles and Techniques 3-4
- CS 227B General Game Playing 3
- CS 243 Program Analysis and Optimizations 3-4
- CS 248 Interactive Computer Graphics 3-4
- CS 341 Project in Mining Massive Data Sets 3
- CS 346 (Offered occasionally) 3-5

Requirement 3: Breadth—
Students must complete at least two courses from the approved Breadth areas A, B, C and D (different Breadth Areas apply to each specialization – see the specialization descriptions under ‘Requirement 4: Specialization’ for specifics). Breadth courses may not be waived, must be taken for at least 3 units each, and must be completed for a letter grade. Each of the two Breadth courses must be from different Areas:

Breadth Area A: Mathematical and Theoretical Foundations
- CS 154, CS 157, CS 168, CS 254, CS 258 (Not Given This Year), CS 261, CS 265, CS 361; EE 364B; PHIL 251

Breadth Area B: Computer Systems
- CS 143, CS 144, CS 242, CS 243, CS 244, CS 244B, CS 316, CS 358; EE 180, EE 282, EE 284

Breadth Area C: Applications
- CS 145, CS 147, CS 148, CS 155, CS 221, CS 223A, CS 223N, CS 224U, CS 224W, CS 227B, CS 228, CS 229, CS 229T, CS 231A, CS 245, CS 246, CS 247 (any suffix), CS 248, CS 251, CS 255, CS 273A, CS 273B, CS 279, CS 348B, CS 348C, CS 355, CS 356, CS 373, CS 448B

Breadth Area D: Computing and Society
- CS 181, CS 182, CS 384; ENGR 131 (Not Given This Year), ENGR 248; ME 177; MS&E 193, MS&E 234

Requirement 4: Specialization—
All courses taken for this requirement must be taken on a letter grade basis for three or more units.

- A program of 21 units must be completed. A maximum of 6 units of independent study (CS 393, CS 395, CS 399) may be counted toward the specialization.

Specialization Areas—
Nine approved specialization areas which may be used to satisfy Requirement 4 are listed following. Students may propose to the M.S. program committee other coherent programs that meet their goals and satisfy the basic requirements.

Courses marked with an asterisk (*) require consent of the faculty adviser. Courses marked with a double asterisk (**) may be waived by students with equivalent course work and with the approval of their adviser.

1. Artificial Intelligence—
A.
- CS 221 Artificial Intelligence: Principles and Techniques **
B. Select at least four of the following:
- CS 223A Introduction to Robotics
- CS 224N Natural Language Processing with Deep Learning
- CS 224S Spoken Language Processing
- CS 224U Natural Language Understanding
- CS 224W Machine Learning with Graphs
- CS 228 Probabilistic Graphical Models: Principles and Techniques
- CS 229 Machine Learning
- CS 231A Computer Vision: From 3D Reconstruction to Recognition
- CS 231N Convolutional Neural Networks for Visual Recognition
- CS 234 Reinforcement Learning
- CS 238 Decision Making under Uncertainty
C. A total of at least 21 units from categories A, B, and the following:
- CS 205L Continuous Mathematical Methods with an Emphasis on Machine Learning
- CS 217 Hardware Accelerators for Machine Learning
- CS 225A Experimental Robotics
- CS 227B General Game Playing
- CS 229T Statistical Learning Theory
- CS 230 Deep Learning
- CS 232 Digital Image Processing
- CS 233 Geometric and Topological Data Analysis
- CS 235 Computational Methods for Biomedical Image Analysis and Interpretation
- CS 236 Deep Generative Models
- CS 237A Principles of Robot Autonomy I
- CS 237B Principles of Robot Autonomy II
- CS 239 Advanced Topics in Sequential Decision Making
- CS 246 Mining Massive Data Sets
- CS 257 Logic and Artificial Intelligence
- CS 270 Modeling Biomedical Systems: Ontology, Terminology, Problem Solving
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 273A</td>
<td>The Human Genome Source Code</td>
</tr>
<tr>
<td>CS 273B</td>
<td>Deep Learning in Genomics and Biomedicine</td>
</tr>
<tr>
<td>CS 274</td>
<td>Representations and Algorithms for Computational Molecular Biology</td>
</tr>
<tr>
<td>CS 275</td>
<td>Translational Bioinformatics</td>
</tr>
<tr>
<td>CS 276</td>
<td>Information Retrieval and Web Search</td>
</tr>
<tr>
<td>CS 279</td>
<td>Computational Biology: Structure and Organization of Biomolecules and Cells</td>
</tr>
<tr>
<td>CS 294A</td>
<td>Research Project in Artificial Intelligence *</td>
</tr>
<tr>
<td>CS 323</td>
<td>Automated Reasoning: Theory and Applications (*)</td>
</tr>
<tr>
<td>CS 325</td>
<td>Topics in Advanced Robotic Manipulation (<em>Topics in</em> course. Offered occasionally.)</td>
</tr>
<tr>
<td>CS 326</td>
<td>Advanced Robotic Manipulation (Not given this year)</td>
</tr>
<tr>
<td>CS 328</td>
<td>Topics in Computer Vision</td>
</tr>
<tr>
<td>CS 329</td>
<td>Topics in Artificial Intelligence</td>
</tr>
<tr>
<td>CS 330</td>
<td>Deep Multi-task and Meta Learning</td>
</tr>
<tr>
<td>CS 331B</td>
<td>Representation Learning in Computer Vision</td>
</tr>
<tr>
<td>CS 332</td>
<td>Advanced Survey of Reinforcement Learning</td>
</tr>
<tr>
<td>CS 333</td>
<td>Algorithms for Interactive Robotics</td>
</tr>
<tr>
<td>CS 334A</td>
<td>Convex Optimization I</td>
</tr>
<tr>
<td>or EE 364A</td>
<td>Convex Optimization I</td>
</tr>
<tr>
<td>CS 336</td>
<td>Robot Perception and Decision-Making: Optimal and Learning-based Approaches</td>
</tr>
<tr>
<td>CS 341</td>
<td>Project in Mining Massive Data Sets</td>
</tr>
<tr>
<td>CS 345</td>
<td>(Offered occasionally)</td>
</tr>
<tr>
<td>CS 368</td>
<td>Algorithmic Techniques for Big Data</td>
</tr>
<tr>
<td>CS 369L</td>
<td>Algorithmic Perspective on Machine Learning</td>
</tr>
<tr>
<td>CS 369M</td>
<td>Metric Embeddings and Algorithmic Applications</td>
</tr>
<tr>
<td>CS 371</td>
<td>Computational Biology in Four Dimensions</td>
</tr>
<tr>
<td>CS 373</td>
<td>Statistical and Machine Learning Methods for Genomics</td>
</tr>
<tr>
<td>CS 375</td>
<td>Large-Scale Neural Network Modeling for Neuroscience</td>
</tr>
<tr>
<td>CS 377</td>
<td>Topics in Human-Computer Interaction (CS 377 with any suffix)</td>
</tr>
<tr>
<td>CS 379</td>
<td>Interdisciplinary Topics (CS 379 with any suffix)</td>
</tr>
<tr>
<td>CS 393</td>
<td>Computer Laboratory *</td>
</tr>
<tr>
<td>CS 395</td>
<td>Independent Database Project *</td>
</tr>
<tr>
<td>CS 398</td>
<td>Computational Education</td>
</tr>
<tr>
<td>CS 399</td>
<td>Independent Project *</td>
</tr>
<tr>
<td>CS 428</td>
<td>Computation and Cognition: The Probabilistic Approach</td>
</tr>
<tr>
<td>APPPHYS 293</td>
<td>Theoretical Neuroscience</td>
</tr>
<tr>
<td>EE 263</td>
<td>Introduction to Linear Dynamical Systems</td>
</tr>
<tr>
<td>EE 276</td>
<td>Information Theory</td>
</tr>
<tr>
<td>EE 278</td>
<td>Introduction to Statistical Signal Processing</td>
</tr>
<tr>
<td>EE 364B</td>
<td>Convex Optimization II</td>
</tr>
<tr>
<td>EE 377</td>
<td>Information Theory and Statistics</td>
</tr>
<tr>
<td>EE 378B</td>
<td>Inference, Estimation, and Information Processing</td>
</tr>
</tbody>
</table>

**Artificial Intelligence Breadth Courses**

- Two courses required. Choose from Breadth Areas A, B and D under Requirement 3.

**2. Biocomputation**

A. Select at least four of the following:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 235</td>
<td>Computational Methods for Biomedical Image Analysis and Interpretation</td>
</tr>
<tr>
<td>CS 270</td>
<td>Modeling Biomedical Systems: Ontology, Terminology, Problem Solving</td>
</tr>
<tr>
<td>CS 272</td>
<td>Introduction to Biomedical Informatics Research Methodology</td>
</tr>
<tr>
<td>CS 273A</td>
<td>The Human Genome Source Code</td>
</tr>
<tr>
<td>CS 274</td>
<td>Representations and Algorithms for Computational Molecular Biology</td>
</tr>
<tr>
<td>CS 279</td>
<td>Computational Biology: Structure and Organization of Biomolecules and Cells</td>
</tr>
</tbody>
</table>

B. A total of at least 21 units from category (A) and the following:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 228</td>
<td>Probabilistic Graphical Models: Principles and Techniques</td>
</tr>
<tr>
<td>CS 229</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>CS 230</td>
<td>Deep Learning</td>
</tr>
<tr>
<td>CS 231N</td>
<td>Convolutional Neural Networks for Visual Recognition</td>
</tr>
<tr>
<td>CS 233</td>
<td>Geometric and Topological Data Analysis</td>
</tr>
<tr>
<td>CS 236</td>
<td>Deep Generative Models</td>
</tr>
<tr>
<td>CS 245</td>
<td>Principles of Data-Intensive Systems</td>
</tr>
<tr>
<td>CS 246</td>
<td>Mining Massive Data Sets</td>
</tr>
<tr>
<td>CS 261</td>
<td>Optimization and Algorithmic Paradigms</td>
</tr>
<tr>
<td>CS 264</td>
<td>Beyond Worst-Case Analysis</td>
</tr>
<tr>
<td>CS 265</td>
<td>Randomized Algorithms and Probabilistic Analysis</td>
</tr>
<tr>
<td>CS 268</td>
<td>Geometric Algorithms</td>
</tr>
<tr>
<td>CS 273B</td>
<td>Deep Learning in Genomics and Biomedicine</td>
</tr>
<tr>
<td>CS 275</td>
<td>Translational Bioinformatics</td>
</tr>
<tr>
<td>CS 325</td>
<td>(Offered occasionally)</td>
</tr>
<tr>
<td>CS 341</td>
<td>Project in Mining Massive Data Sets</td>
</tr>
<tr>
<td>CS 345</td>
<td>(Offered occasionally)</td>
</tr>
</tbody>
</table>
Biocomputation Breadth Courses
- Two courses required. Choose from Breadth Areas A, B and D under Requirement 3.

3. Computer and Network Security—

A.

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 140</td>
<td>Operating Systems and Systems Programming **</td>
</tr>
<tr>
<td>or CS 140E</td>
<td>Operating systems design and implementation</td>
</tr>
<tr>
<td>CS 144</td>
<td>Introduction to Computer Networking **</td>
</tr>
<tr>
<td>CS 155</td>
<td>Computer and Network Security</td>
</tr>
<tr>
<td>CS 244</td>
<td>Advanced Topics in Networking</td>
</tr>
<tr>
<td>CS 255</td>
<td>Introduction to Cryptography</td>
</tr>
</tbody>
</table>

B. Select at least three of the following:

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 142</td>
<td>Web Applications</td>
</tr>
<tr>
<td>CS 190</td>
<td>Software Design Studio</td>
</tr>
<tr>
<td>CS 240</td>
<td>Advanced Topics in Operating Systems</td>
</tr>
<tr>
<td>CS 244B</td>
<td>Distributed Systems</td>
</tr>
<tr>
<td>CS 261</td>
<td>Optimization and Algorithmic Paradigms</td>
</tr>
<tr>
<td>CS 265</td>
<td>Randomized Algorithms and Probabilistic Analysis</td>
</tr>
<tr>
<td>CS 340</td>
<td>Topics in Computer Systems</td>
</tr>
<tr>
<td>CS 344</td>
<td>Topics in Computer Networks (CS 344 with any suffix)</td>
</tr>
<tr>
<td>CS 355</td>
<td>Advanced Topics in Cryptography (Not given this year)</td>
</tr>
<tr>
<td>CS 356</td>
<td>Topics in Computer and Network Security</td>
</tr>
</tbody>
</table>

C. A total of at least 21 units from categories (A), (B), and the following:

- Broader CS
- Art Studio
- Communication
- Empirical Methods

4. Human-Computer Interaction—

A.

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 147</td>
<td>Introduction to Human-Computer Interaction Design **</td>
</tr>
<tr>
<td>CS 247</td>
<td>(Any suffix) *</td>
</tr>
</tbody>
</table>

B. Select any three of the following:

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 142</td>
<td>Web Applications</td>
</tr>
<tr>
<td>CS 146</td>
<td>Introduction to Game Design and Development</td>
</tr>
<tr>
<td>CS 148</td>
<td>Introduction to Computer Graphics and Imaging</td>
</tr>
<tr>
<td>CS 194H</td>
<td>User Interface Design Project</td>
</tr>
<tr>
<td>CS 206</td>
<td>Exploring Computational Journalism</td>
</tr>
<tr>
<td>CS 210A</td>
<td>Software Project Experience with Corporate Partners</td>
</tr>
<tr>
<td>CS 248</td>
<td>Interactive Computer Graphics</td>
</tr>
<tr>
<td>CS 278</td>
<td>Social Computing</td>
</tr>
<tr>
<td>CS 347</td>
<td>Human-Computer Interaction Research</td>
</tr>
<tr>
<td>CS 377</td>
<td>Topics in Human-Computer Interaction (CS 377 with any suffix)</td>
</tr>
<tr>
<td>CS 448B</td>
<td>Data Visualization</td>
</tr>
<tr>
<td>ME 216M</td>
<td>Introduction to the Design of Smart Products</td>
</tr>
</tbody>
</table>

C. A total of at least 21 units from categories (A), (B), and the following:

- Broader CS
- Art Studio
- Communication
- Empirical Methods
1. Computer Science
   - COMM 314: Ethnographic Methods
   - EDUC 200B: Introduction to Qualitative Research Methods
   - MS&E 125: Introduction to Applied Statistics
   - PSYCH 251: Experimental Methods
   - PSYCH 252: Statistical Methods for Behavioral and Social Sciences
   - PSYCH 253: Advanced Statistical Modeling
   - STATS 203: Introduction to Regression Models and Analysis of Variance

2. e. Learning Design & Tech
   - EDUC 236: Beyond Bits and Atoms: Designing Technological Tools
   - EDUC 239: Educating Young STEM Thinkers
   - EDUC 281: Technology for Learners
   - EDUC 342: Child Development and New Technologies

3. f. Management Science & Engr
   - MS&E 185: Global Work
   - MS&E 231: Introduction to Computational Social Science
   - MS&E 334: Topics in Social Data

4. g. Mechanical Engr
   - ME 203: Design and Manufacturing
   - ME 210: Introduction to Mechatronics
   - ME 216A: Advanced Product Design: Needfinding

5. h. Music
   - MUSIC 220A: Fundamentals of Computer-Generated Sound
   - MUSIC 220B: Compositional Algorithms, Psychoacoustics, and Computational Music
   - MUSIC 220C: Research Seminar in Computer-Generated Music
   - MUSIC 250A: Physical Interaction Design for Music
   - MUSIC 256A: Music, Computing, Design I: The Art of Design

6. i. Psych
   - PSYCH 204: Computation and Cognition: The Probabilistic Approach
   - PSYCH 209: Neural Network Models of Cognition

7. j. Sym Sys
   - SYMSYS 245: Cognition in Interaction Design

8. Additional courses

9. Note that if CS145 was waived in area (A), students should take an additional course from either area (B) or (C) in its place.

Human-Computer Interaction Breadth Courses
- Two courses required. Choose from Breadth Areas A, B and D under Requirement 3.

5. Information Management and Analytics—
   A. CS 145: Data Management and Data Systems **  3-4
   B. Select at least four of the following:
      - CS 224N: Natural Language Processing with Deep Learning
      - CS 224W: Machine Learning with Graphs
      - CS 229: Machine Learning

   • Note that if CS145 was waived in area (A), students should take an additional course from either area (B) or (C) in its place.

Information Management and Analytics Breadth Courses
- Two courses required. Choose from Breadth Areas A, B and D under Requirement 3.

1 Any d.school course (http://dschool.stanford.edu) or any HCI course (http://hci.stanford.edu/courses); such courses must be numbered 100 or above and be taken for at least 3 units to count for this requirement.
6. Real-World Computing—

A. Select at least three of the following:

- **CS 148** Introduction to Computer Graphics and Imaging
- **CS 223A** Introduction to Robotics
- **CS 231A** Computer Vision: From 3D Reconstruction to Recognition
- **CS 248** Interactive Computer Graphics

B. Select at least three of the following:

- **CS 205L** Continuous Mathematical Methods with an Emphasis on Machine Learning
- **CS 233** Geometric and Topological Data Analysis
- **CS 268** Geometric Algorithms
- **CS 348A** Computer Graphics: Geometric Modeling & Processing
- **CS 348B** Computer Graphics: Image Synthesis Techniques
- **CS 348C** Computer Graphics: Animation and Simulation
- **CS 348E** Character Animation: Modeling, Simulation, and Control of Human Motion
- **CS 348K** Visual Computing Systems
- **CME 302** Numerical Linear Algebra
- **CME 306** Numerical Solution of Partial Differential Equations

C. A total of at least 21 units from categories (A), (B), and the following:

- **CS 146** Introduction to Game Design and Development
- **CS 225A** Experimental Robotics
- **CS 228** Probabilistic Graphical Models: Principles and Techniques
- **CS 229** Machine Learning
- **CS 230** Deep Learning
- **CS 232** Digital Image Processing or **EE 368** Digital Image Processing
- **CS 247** (Any suffix)
- **CS 270** Modeling Biomedical Systems: Ontology, Terminology, Problem Solving
- **CS 272** Introduction to Biomedical Informatics Research Methodology
- **CS 273A** The Human Genome Source Code
- **CS 274** Representations and Algorithms for Computational Molecular Biology
- **CS 294A** Research Project in Artificial Intelligence *
- **CS 326** Topics in Advanced Robotic Manipulation
- **CS 327A** Advanced Robotic Manipulation (Not given this year)
- **CS 328** Topics in Computer Vision
- **CS 331B** Representation Learning in Computer Vision
- **CS 333** Algorithms for Interactive Robotics
- **CS 393** Computer Laboratory *
- **CS 395** Independent Database Project *
- **CS 399** Independent Project *
- **CS 448** Topics in Computer Graphics (CS 448 with any suffix)
- **EE 267** Virtual Reality

Real-World Computing Breadth Courses

- Two courses required. Choose from Breadth Areas A, B and D under Requirement 3.

7. Software Theory—

A.

- **CS 243** Program Analysis and Optimizations

B. Select at least one of the following:

- **CS 244** Advanced Topics in Networking
- **CS 245** Principles of Data-Intensive Systems
- **CS 341** Project in Mining Massive Data Sets
- **CS 345** (Offered occasionally)

C. Select at least two courses from the following:

- **CS 242** Programming Languages
- **CS 255** Introduction to Cryptography
- **CS 261** Optimization and Algorithmic Paradigms
- **CS 264** Beyond Worst-Case Analysis
- **CS 265** Randomized Algorithms and Probabilistic Analysis
- **CS 268** Geometric Algorithms
- **CS 355** Advanced Topics in Cryptography (Not given this year)

D. A total of at least 21 units from (A), (B), (C), or the following:

- **CS 151** Logic Programming
- **CS 250** Algebraic Error Correcting Codes
- **CS 251** Cryptocurrencies and blockchain technologies
- **CS 252** Analysis of Boolean Functions
- **CS 294S** Research Project in Software Systems and Security (Not given this year) *
- **CS 350** Secure Compilation
- **CS 356** Topics in Computer and Network Security
- **CS 393** Computer Laboratory *
- **CS 395** Independent Database Project *
- **CS 399** Independent Project *

Software Theory Breadth Courses

- Two courses required. Choose from Breadth Areas A, C and D under Requirement 3.

8. Systems—

A.

- **CS 140** Operating Systems and Systems Programming *
- or **CS 140E** Operating systems design and implementation
- **CS 144** Introduction to Computer Networking *
- **CS 240** Advanced Topics in Operating Systems

B. Select at least four of the following:

- **CS 190** Software Design Studio
- **CS 242** Programming Languages
- **CS 243** Program Analysis and Optimizations
- **CS 244** Advanced Topics in Networking
- **CS 245** Principles of Data-Intensive Systems
- **CS 248** Interactive Computer Graphics
- **CS 348B** Computer Graphics: Image Synthesis Techniques
- **EE 271** Introduction to VLSI Systems
- **EE 282** Computer Systems Architecture

C. A total of at least 21 units from categories (A), (B), and the following:
### Systems Breadth Courses

- **Two courses required. Choose from Breadth Areas A, C and D under Requirement 3.**

#### 9. Theoretical Computer Science—

**A.**

- CS 154: Introduction to Automata and Complexity Theory **
- CS 261: Optimization and Algorithmic Paradigms

**B.** A total of at least 21 units from category (A) and the following:

- CS 151: Logic Programming
- CS 166: Data Structures
- CS 168: The Modern Algorithmic Toolbox
- CS 228: Probabilistic Graphical Models: Principles and Techniques
- CS 233: Geometric and Topological Data Analysis
- CS 236: Deep Generative Models
- CS 246: Mining Massive Data Sets
- CS 250: Algebraic Error Correcting Codes
- CS 251: Cryptocurrencies and blockchain technologies
- CS 252: Analysis of Boolean Functions
- CS 254: Computational Complexity
- CS 254B: Computational Complexity II
- CS 255: Introduction to Cryptography
- CS 257: Logic and Artificial Intelligence
- CS 264: Beyond Worst-Case Analysis
- CS 265: Randomized Algorithms and Probabilistic Analysis
- CS 268: Geometric Algorithms
- CS 269G: Almost Linear Time Graph Algorithms
- CS 269I: (Not Given This Year)
- CS 269O: Introduction to Optimization Theory
- CS 334A: Convex Optimization I
- CS 334B: Convex Optimization II
- CS 334G: Advanced Topics in Convex Optimization
- CS 335: Advanced Topics in Cryptography (Not given this year)
- CS 338: Topics in Programming Language Theory
- CS 339: Topics in the Theory of Computation
- CS 368: Algorithmic Techniques for Big Data
- CS 369: Topics in Analysis of Algorithms
- CS 393: Computer Laboratory
- CS 399: Independent Project

**EE 267: Virtual Reality**

**EE 273: Digital Systems Engineering**

**EE 382C: Interconnection Networks**

**EE 384A: Internet Routing Protocols and Standards**

**EE 384C: Wireless Local and Wide Area Networks**

**EE 384S: Performance Engineering of Computer Systems & Networks**

- **MS&E 310: Linear Programming**
- **MS&E 315: Advanced Optimization Theory**
- **MS&E 319: Matching Theory**

* Multiple CS 359, CS 369, and/or CS 468 courses may be taken as long as they are each on different topics, denoted by different letter suffixes for the courses.

### Theoretical Computer Science Breadth Courses

- **Two courses required. Choose from Breadth Areas B, C and D under Requirement 3.**

* With consent of faculty adviser.

** Students with equivalent course work may waive with approval of their adviser.

### Requirement 5

Additional elective units must be technical courses (numbered 100 or above) related to the degree program and approved by the adviser and MS program administrator. All CS courses numbered above 110.
(with the exception of CS 196 and CS 198) taken for 3 or more units are pre-approved as elective courses. Additionally, up to a maximum of 3 units of 500-level CS seminars, CS 300, EE 380, EE 385A, or other 1-2 unit seminars offered in the School of Engineering may be counted as electives. Elective courses may be taken on a satisfactory/no credit basis provided that a minimum of 36 graded units is presented within the 45-unit program.

**Master of Science with Distinction in Research**

A student who wishes to pursue the M.S. in CS with distinction in research must first identify a faculty adviser who agrees to supervise and support the research work. The research adviser must be a member of the Academic Council and must hold an appointment in Computer Science. The student and principal adviser must also identify another faculty member, who need not be in the Department of Computer Science, to serve as a secondary adviser and reader for the research report. In addition, the student must complete the following requirements beyond those for the regular M.S. in CS degree:

1. **Research Experience**—The program must include significant research experience at the level of a half-time commitment over the course of three academic quarters. In any given quarter, the half-time research commitment may be satisfied by a 50 percent appointment to a departmentally supported research assistantship, 6 units of independent study (CS 393, CS 395, or CS 399), or a prorated combination of the two (such as a 25 percent research assistantship supplemented by 3 units of independent study). This research must be carried out under the direction of the primary or secondary adviser.

2. **Supervised Writing and Research**—In addition to the research experience outlined in the previous requirement, students must enroll in at least 3 units of independent research (CS 393, CS 395, or CS 399) under the direction of their primary or secondary adviser. These units should be closely related to the research described in the first requirement, but focused more directly on the preparation of the research report described in the next section. The writing and research units described in parts (1) and (2) may be counted toward the 45 units required for the degree.

3. All independent study units (CS 393, CS 395, CS 399) must be taken for letter grades and a GPA of 3.0 (B) or better must be maintained.

4. **Research Report**—Students must complete a significant report describing their research and its conclusions. The research report represents work that is publishable in a journal or at a high-quality conference, although it is presumably longer and more expansive in scope than a typical conference paper. A copy of the research report must be submitted to the student services office in the department three weeks before the beginning of the examination period in the student’s final quarter. Both the primary and secondary adviser must approve the research report before the distinction-in-research designation can be conferred.

**Joint M.S. and MBA Degree**

The joint MS in Computer Science/MBA degree links two of Stanford University’s world-class programs. This joint degree offers students an opportunity to develop advanced technical and managerial skills for a broader perspective on both existing technologies and new technology ventures.

Admission to the joint MSCS/MBA program requires that students apply and be accepted independently to both the Computer Science Department in the School of Engineering and the Graduate School of Business. Students may apply concurrently, or elect to begin their course of study in CS and apply to the GSB during their first year.

Additional information on the MS in Computer Science/MBA Joint Degree Program and its requirements is available on the department’s web site (https://cs.stanford.edu/academics/joint-degree-programs/joint-cs-mmba-degree).

**Joint M.S. and Law Degree**

Law students interested in pursuing an M.S. in Computer Science must apply for admission to the Computer Science Department either (i) concurrently with applying to the Law School; or (ii) after being admitted to the Law School, but no later than the earlier of: (a) the end of the second year of Law School; or (b) the Computer Science Department’s admission deadline for the year following that second year of Law School.

In addition to being admitted separately to the Law School and the Computer Science Department, students must secure permission from both academic units to pursue degrees in those units as part of a joint degree program.

J.D./M.S. students may elect to begin their course of study in either the Law School or the Computer Science Department. Faculty advisors from each academic unit participate in the planning and supervising of the student’s joint program. Students must be enrolled full-time in the Law School for the first year of law studies. Otherwise, enrollment may be in the graduate school or the Law School, and students may choose courses from either program regardless of where enrolled. Students must satisfy the requirements for both the J.D. degree as specified by the Law School and the M.S. degree as specified in this Bulletin.

The Law School approves courses from the Department of Computer Science that may count toward the J.D. degree, and the Computer Science Department approves courses from the Law School that may count toward the M.S. degree in Computer Science. In either case, approval may consist of a list applicable to all joint-degree students or may be tailored to each individual student program. No more than 45 units of approved courses may be counted toward both degrees. No more than 36 units of courses that originate outside the Law School may count toward the Law degree. To the extent that courses under this joint degree program originate outside of the Law School but count toward the Law degree, the Law School credits permitted under Section 17(1) of the Law School Regulations shall be reduced on a unit-per-unit basis, but not below zero. The maximum number of Law School credits that may be counted toward the M.S. in Computer Science is the greater of: (i) 12 units; or (ii) the maximum number of units from courses outside of the department that M.S. candidates in Computer Science are permitted to count toward the M.S. in the case of a particular student’s individual program. Tuition and financial aid arrangements are normally through the school in which the student is then enrolled.

**Teaching and Research Assistantships in Computer Science**

Graduate student assistantships are available. Half-time assistants receive a tuition scholarship for 8, 9, or 10 units per quarter during the academic year, and in addition receive a monthly stipend.

Duties for half-time assistants during the academic year involve approximately 20 hours of work per week. Course assistants (CAs) help an instructor teach a course by conducting discussion sections, consulting with students, and grading examinations. Research assistants (RAs) help faculty and senior staff members with research in computer science. Many MS students are hired to staff teaching and research assistantships. However, MS students should not plan on being appointed to an assistantship.

Students with fellowships may have the opportunity to supplement their stipends by serving as graduate student assistants.
Doctor of Philosophy in Computer Science

The University’s basic requirements for the Ph.D. degree are outlined in the “Graduate Degrees” section of this bulletin. Department requirements are stated below.

Requirements

Applications to the Ph.D. program and all supporting documents must be submitted and received online by the published deadline. See the department’s website for admissions requirements and the application deadline (https://cs.stanford.edu/admissions/general-information). Changes or updates to the admission process are posted in September.

The following are general department requirements. Contact the Computer Science Ph.D. administrator for details.

1. A student should plan and complete a coherent program of study covering the basic areas of computer science and related disciplines. The student’s advisor has primary responsibility for the adequacy of the program, which is subject to review by the Student Services Office.

2. The first year of the Ph.D. program is spent working with 1-3 different professors on a rotating basis. The intent is to allow the first-year Ph.D. student to work with a variety of professors before aligning with a permanent program advisor. Students who don’t need the full year to find a professor to align with will have the option of aligning within the first or second quarter.

3. The CS 300 (http://exploreCourses.stanford.edu/search?view=catalog&filter-coursestatus-Active=on&page=0&catalog=&academicYear=&q=CS +300&collapse=) Departmental Lecture Series seminar gives faculty the opportunity to explain their research to first-year CS Ph.D. students. First-year CS Ph.D. students are required to attend 2/3 of the classes to receive credit.

4. A student must complete 135 course units for graduation. Computer Science Ph.D. students take 8-10 units per quarter. Credit for coursework done elsewhere (up to the maximum of 45 course units) may be applied to graduation requirements. Students must also take at least three units of coursework from four different faculty members. There are NO courses specifically required by the CS Ph.D. program except for the 1 unit CS 300 (https://exploreCourses.stanford.edu/search?view=catalog&filter-coursestatus-Active=on&page=0&catalog=&academicYear=&q=CS +300&collapse=) Departmental Lecture Series and CS 499 Advanced Reading and Research or its equivalent. At least one course must be taken for a letter grade. A 3.0 GPA must be maintained.

5. Each student, to remain in the Ph.D. program, must satisfy the breadth requirement covering introductory-level graduate material in major areas of computer science. A student must fulfill two breadth-area requirements in each of three general areas by the end of the second year in the program. If students have fulfilled the six breadth-area requirements, and taken courses from at least four different faculty who are members of the Academic Council, they are eligible to apply for candidacy prior to the second year in the program. An up-to-date list of courses that satisfy the breadth requirements (http://cs.stanford.edu/education/phd) can be found on the department’s website. The student must completely satisfy the breadth requirement by the end of the second year in the program and must pass a qualifying exam in the general area of their expected dissertation by the end of the third year in the program.

6. University policy requires that all doctoral students declare candidacy by the end of the sixth quarter in residence, excluding summers. However, after aligning with a permanent adviser, passing six breadth requirements, and taking classes with four different faculty, a student is eligible to file for candidacy prior to the sixth quarter. The candidacy form serves as a “contract” between the department and the student. The department acknowledges that the student is a bona fide candidate for the Ph.D. and agrees that the program submitted by the student is sufficient to warrant granting the Ph.D. upon completion. Candidacy expires five years from the date of submission of the candidacy form, rounded to the end of the quarter. In special cases, the department may extend a student’s candidacy, but is under no obligation to do so.

7. Each student is required to pass a qualifying exam in their area by the end of their third year in the program. A student may only take the qualifying exam twice. If the student fails the qualifying exam a second time, the Ph.D. program committee is convened to discuss the student’s lack of reasonable academic progress. Failing the exam a second time is cause for dismissal from the Computer Science Ph.D. program and the committee meets to discuss the final outcome for the student.

8. As part of the training for the Ph.D., the student is also required to complete at least four units (a unit is ten hours per week for one quarter) as a course assistant or instructor for courses in Computer Science numbered 100 or above.

9. The student must present an oral thesis proposal and submit the form to their full reading committee by Spring Quarter of the fourth year. The Thesis Proposal Form (https://cs.stanford.edu/degrees/PHD/ThesisProposalForm.pdf) must be filled out, signed and approved by all members of the committee and submitted to the CS Ph.D. student services in Gates 196. The goal of the thesis proposal is to enable students to get better formative feedback from their reading committee on what directions to take to successfully complete a quality dissertation. The thesis proposal should allow plenty of time for discussion with the reading committee about the direction of the thesis research.

10. The Oral Thesis Proposal must be submitted before the end of the fourth year.

11. The most important requirement is the dissertation. After passing the required qualifying examination, each student must secure the agreement of a member of the department faculty to act as the dissertation advisor. The dissertation advisor is often the student’s program advisor.

12. The student must pass a University oral examination in the form of a defense of the dissertation. This is typically held after all or a substantial portion of the dissertation research has been completed.

13. The student is expected to demonstrate the ability to present scholarly material orally in the dissertation defense.

14. The dissertation must be accepted by a reading committee composed of the principal dissertation advisor, a second member from within the department, and a third member chosen from within or outside of the University. The department requires at least two committee members to be affiliated with the Computer Science department. The principal advisor and at least one of the other committee members must be Academic Council members.

Guidelines for Reasonable Progress

- By the end of the first academic year, you should align with a permanent adviser. Students are welcome to switch advisers, but a student should not have significant periods of time (after the first year) with no adviser.

- A student must make satisfactory progress in his or her research, as determined by his or her adviser.

- By Spring Quarter of the second year, a student should complete all six breadth area requirements, two breadth area requirements in each of three areas, and file for candidacy.

- By Spring Quarter of the third year, a student should pass a Qualifying Examination (https://cs.stanford.edu/academics/phd/qualifying-exams) in the area of his or her intended dissertation.
Ph.D. Minor in Computer Science

For a minor in Computer Science, a candidate must complete 20 units of Computer Science coursework numbered 200 or above, except for the 100-level courses listed on the Ph.D. Minor Worksheet (https://cs.stanford.edu/sites/default/files/PhDMinorWorksheet_2.pdf). At least three of the courses must be master’s core courses to provide breadth and one course numbered 300 or above to provide depth. One of the courses taken must include a significant programming project to demonstrate programming efficiency. Courses must be taken for a letter grade and passed with a grade of ‘B’ or better. Applications for a minor in Computer Science are submitted at the same time as admission to candidacy.

Graduate Advising Expectations

The Department of Computer Science is committed to providing academic advising in support of graduate student scholarly and professional development. The most effective, this advising relationship entails collaborative and sustained engagement by both the adviser and the advisee. As a best practice, advising expectations should be periodically discussed and reviewed to ensure mutual understanding. Both the adviser and the advisee are expected to maintain professionalism and integrity.

Faculty advisers guide students in key areas such as selecting courses, designing and conducting research, developing of teaching pedagogy, navigating policies and degree requirements, and exploring academic opportunities and professional pathways.

Graduate students are active contributors to the advising relationship, proactively seeking academic and professional guidance and taking responsibility for informing themselves of policies and degree requirements for their graduate program.

For a statement of Computer Science policy on graduate advising, see the Computer Science Graduate Advising (https://cs.stanford.edu/academics/phd/phd-advising) link. For a statement of University policy on graduate advising, see the "Graduate Advising (http://exploredegrees.stanford.edu/graduatedegrees/ #advisingandcredentialtext)" section of this bulletin.


Chair: John Mitchell (http://theory.stanford.edu/people/jcm/home.html)
Associate Chair for Education: Mehran Sahami (http://robotics.stanford.edu/users/sahami/bio.html)
Director of Ph.D. Program: John Ousterhout (https://web.stanford.edu/~ouster/cgi-bin/home.php)
Director of M.S. Program: Omer Reingold (https://omereingold.wordpress.com)
Director of B.S. Program: Gerald Cain


Professor (Teaching): Mehran Sahami

Courtesy Professors: Russ Altman, Stephen Boyd, Jacob Fox, Patrick Hayden, Michael Levitt, Roy Pea, Russell Poldrak, Daniel Rubin, Marco Pavone

Courtesy Associate Professors: Ashish Goel, Allison Okamura, Chris Potts, Ge Wang

Courtesy Assistant Professors: John Duchi, Sean Follmer, Surya Ganguli, Sharad Goel, Thomas Icard, Ramesh Johari, Mykel Kochenderfer, Stephen Montgomery, Aaron Sidford, Camille Utterback, Gordon Wetzstein, Dan Yamins, James Zou

Senior Lecturers: Gerald Cain, Nicholas J. Parlante, Keith Schwarz, Julie Zelenski

Lecturers: Jay Borenstein, Chris Gregg, Cynthia Lee, Julie Stanford, Nick Troccoli, David Varodayan, Christina Wodtke, Lisa Yan, Patrick Young

Adjunct Professors: Pei Cao, Stuart Card, Daphne Koller, Bill MacCartney, P. Pandurang Nayak, Andrew Ng, Sebastian Thrun

Visiting Assistant Professors: Lucjan Hanzlik, HoJoon Lee, Hamed Nemati, Marco Patrignani, Giancarlo Pellegrino

Secondary Appointment in CS: Anshul Kundaje

*recalled to active duty.

The Bing Overseas Studies Program manages Stanford study abroad programs for Stanford undergraduates. Students should consult their department or program's student services office for applicability of Overseas Studies courses to a major or minor program.

The Bing Overseas Studies course search site displays courses, locations, and quarters relevant to specific majors.

For course descriptions and additional offerings, see the listings in the Stanford Bulletin's ExploreCourses or Bing Overseas Studies (http://bosp.stanford.edu).

<table>
<thead>
<tr>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPOXFRO 62</td>
</tr>
<tr>
<td>OSPOXFRO 63</td>
</tr>
</tbody>
</table>