School of Computer Science

Andrew Moore, Dean
Guy Blelloch, Associate Dean for Undergraduate Education
Thomas Cortina, Assistant Dean for Undergraduate Education
Undergraduate Office: GHC 4115
https://www.csd.cs.cmu.edu/academics/undergraduate/overview

Carnegie Mellon founded one of the first Computer Science departments in the world in 1965. Today, the Computer Science Department forms the centerpiece of the School of Computer Science, and is joined by the Human-Computer Interaction Institute, the Institute for Software Research, the Computational Biology Department, the Language Technologies Institute, the Machine Learning Department, and the Robotics Institute. Together, these units make the School of Computer Science a world leader in research and education.

The B.S. program in Computer Science combines a solid core of Computer Science courses with the ability to gain substantial depth in another area through a required minor in a second subject. In addition, the curriculum provides numerous choices for science, engineering, humanities and fine arts courses. As computing is a discipline with strong links to many fields, this provides students with unparalleled flexibility to pursue allied (or non-allied) interests. The curriculum’s mathematics and probability component ensures that students have the formal tools to remain current as technologies and systems change, rather than be limited by a narrow focus on programming alone. At the same time, students gain insight into the practical issues of building and maintaining systems by participating in intensive project-oriented courses. Due to the tremendous number of ongoing research projects within the School, many students obtain part-time or summer jobs, or receive independent study credit, working on research while pursuing their undergraduate degree. Students seeking a research/graduate school career may pursue an intensive course of research, equivalent to four classroom courses, culminating in the preparation of a senior research thesis.

Students apply to, and are directly admitted into, the undergraduate program in Computer Science and, upon successful completion, are awarded a Bachelor of Science in Computer Science. Suitably prepared students from other Carnegie Mellon colleges are eligible to apply for internal transfer to the School of Computer Science and will be considered for transfer if grades in core CS requirements are sufficiently high and space is available. Computation-oriented programs are also available within the Departments of Biology, Chemistry, Physics, Electrical and Computer Engineering, Information Systems, Philosophy, Psychology, and Design. We also offer a B.S. degree in Computational Biology (jointly with the Mellon College of Science), a B.S. degree in Statistics and Machine Learning (jointly with the Dietrich College of Humanities and Social Sciences) and a Bachelor’s Degree in Computer Science and the Arts (jointly with the College of Fine Arts). SCS offers additional majors in Computer Science (for non-CS majors), Human-Computer Interaction, and Robotics, and minors in Computational Biology, Computer Science (for non-CS majors), Human-Computer Interaction, Language Technologies, Machine Learning, Neural Computation, Robotics, and Software Engineering.

Curriculum - B.S. in Computer Science

Computer Science (students entering Fall 2016)

Computer Science Core (all of the following): Units
15-128 Freshman Immigration Course 1
15-122 Principles of Imperative Computation 10 (students with no prior programming experience take 15-112 before 15-122)
15-150 Principles of Functional Programming 10
15-151 Mathematical Foundations for Computer Science 10 (if not offered, substitute 21-127)
15-210 Parallel and Sequential Data Structures and Algorithms 12
15-213 Introduction to Computer Systems 12
15-251 Great Theoretical Ideas in Computer Science 12
15-451 Algorithm Design and Analysis 12
One Communications course: Units
76-270 Writing for the Professions 9
15-300 Research and Innovation in Computer Science 9
15-221 Technical Communication for Computer Scientists 9
One Algorithms/Complexity elective (min. 9 units): 15-354 Computational Discrete Mathematics 12
15-355 Modern Computer Algebra 9
15-453 Formal Languages, Automata, and Computability 9
15-455 Undergraduate Complexity Theory 9
15-456 Computational Geometry 9
21-301 Combinatorics 9
21-484 Graph Theory 9
others as designated by the CS Undergraduate Program
One Logics/Languages elective (min. 9 units): 15-312 Foundations of Programming Languages 12
15-317 Constructive Logic 9
15-414 Bug Catching: Automated Program Verification and Testing 9
15-424 Foundations of Cyber-Physical Systems 12
21-300 Basic Logic 9
80-310 Formal Logic 9
80-311 Undecidability and Incompleteness 9
others as designated by the CS Undergraduate Program
One Software Systems elective (min. 12 units): 15-410 Operating System Design and Implementation 15
15-411 Compiler Design 15
15-418 Parallel Computer Architecture and Programming 12
15-440 Distributed Systems 12
15-441 Computer Networks 12
others as designated by the CS Undergraduate Program
One Applications elective, representing important branches of computer science (min. 9 units): 02-510 Computational Genomics 12
05-391 Designing Human Centered Software 12
10-401 Introduction to Machine Learning (Undergrad) (or 10-601 Introduction to Machine Learning) 12
11-411 Natural Language Processing 12
15-313 Foundations of Software Engineering 12
15-323 Computer Music Systems and Information Processing (or 15-322 Introduction to Computer Music) 9
15-381 Artificial Intelligence: Representation and Problem Solving 9
15-415 Database Applications 12
15-462 Computer Graphics 12
16-384 Robot Kinematics and Dynamics 12
16-385 Computer Vision 9
others as designated by the CS Undergraduate Program
Two Computer Science electives: Units
These electives can be from any SCS department; 200-level or above, at least 9 units each (see exceptions below): 21-201, 21-223, 21-250, 21-261, 08-200, 08-532, 15-351, 16-223. Consult with a CS undergraduate advisor before registration to determine eligibility for this requirement.

Mathematics

21-120 Differential and Integral Calculus 10
21-122 Integration and Approximation 10
Plus one of the following Matrix Algebra courses: 21-241 Matrices and Linear Transformations 10
21-242 Matrix Theory 10
Science and Engineering

Four courses in science and engineering are required, of which at least one must have a laboratory component and at least two must be from the same department. At present, courses meeting the lab requirement are:

02-261 Quantitative Cell and Molecular Biology Laboratory 9
03-124 Modern Biology Laboratory 9
09-101 Introduction to Experimental Chemistry 3
(Also 09-105 satisfies the lab requirement.)
09-221 Laboratory I: Introduction to Chemical Analysis 12
27-100 Engineering the Materials of the Future 12
33-104 Experimental Physics 9
42-203 Biomedical Engineering Laboratory 9
85-310 Research Methods in Cognitive Psychology 9
85-314 Cognitive Neuroscience Research Methods 9

The following courses from the Dietrich College of Humanities and Social Sciences and University Wide Studies can be used to satisfy the Science and Engineering requirement:

85-219 Biological Foundations of Behavior 9
85-310 Research Methods in Cognitive Psychology 9
85-314 Cognitive Neuroscience Research Methods 9
99-238 Materials, Energy and Environment 9

The following courses from the Computational Biology Department can be used to satisfy the Science and Engineering requirement and can be paired with a Biology (03-) or Mathematics (09-) course for two courses from one department:

02-223 Personalized Medicine: Understanding Your Own Genome 9
02-250 Introduction to Computational Biology (or 02-251 + 02-252) 12
02-261 Quantitative Cell and Molecular Biology Laboratory 9

The following MCS and CIT courses cannot be used to satisfy the Engineering and Natural Sciences requirement:

03-511 Computational Molecular Biology and Genomics 9
03-512 Computational Methods for Biological Modeling and Simulation 9
06-262 Mathematical Methods of Chemical Engineering 12
09-103 Atoms, Molecules and Chemical Change 9
09-231 Mathematical Methods for Chemists 9
12-271 Introduction to Computer Application in Civil & Environmental Engineering 9
18-090 Twisted Signals: Multimedia Processing for the Arts 10

Category 3:

Complete three courses, one each from Category 1, Category 2, and Category 3:

Humanities and Arts

All candidates for the bachelor's degree must complete a minimum of 63 units offered by the College of Humanities & Social Sciences and/or the College of Fine Arts as prescribed below:

A. Writing Requirement (9 units)
Complete the following course:
76-101 Interpretation and Argument 9

B. Breadth Requirement (minimum 27 units: 9 units each)
Complete three courses, one each from Category 1, Category 2, and Category 3:

Category 1: Cognition, Choice and Behavior - this requirement explores the processes of thinking, decision making, and behavior in the context of the individual.
70-311 Organizational Behavior 9
80-130 Introduction to Ethics 9
80-150 Nature of Reason 9
80-180 Nature of Language 9
80-221 Philosophy of Social Science 9
80-230 Ethical Theory 9
80-241 Ethical Judgments in Professional Life 9
80-242 Conflict and Dispute Resolution 9
80-270 Philosophy of Mind 9
80-271 Philosophy and Psychology 9
80-275 Metaphysics 9
80-281 Language and Thought 9
85-102 Introduction to Psychology 9
85-211 Cognitive Psychology 9
85-221 Principles of Child Development 9
85-241 Social Psychology 9
85-251 Personality 9
85-261 Abnormal Psychology 9
88-120 Reason, Passion and Cognition 9
88-260 Organizations 9

Category 2: Economic, Political and Social Institutions - this requirement explores the processes by which institutions organize individual preferences and actions into collective outcomes.
19-432 Special Topics: Bitcoin and Cryptocurrencies 6
27-410 Computational Techniques in Engineering 12
33-100 Basic Experimental Physics 6
33-115 Physics for Future Presidents 9
33-124 Introduction to Astronomy 9
33-232 Mathematical Methods of Physics 10
39-100 Special Topics: WHAT IS ENGINEERING? 9
39-200 Business for Engineers 9
42-201 Professional Issues in Biomedical Engineering 3

In addition, all Electrical and Computer Engineering graduate courses [18-6xx, 18-7xx, 18-8xx, 18-9xx] cannot be used for this requirement. Consult with a CS undergraduate advisor about any course to be used for the Science and Engineering requirement before registration.
Category 3: Cultural Analysis - this requirement seeks to recognize cultures that have shaped and continue to shape the human experience; courses in this category are usually either broad in place, time, or cultural diversity.

Survey of Western Music History 9
Modern Visual Culture 1789-1960 9
Managing Across Cultures 9
Books You Should Have Read By Now 9
Comedy 9
Introduction to African American Literature 9
Introduction to Film Studies 9
Introduction to Gender Studies 9
Global Histories 9
Introduction to Anthropology 9
Flesh and Spirit: Early Modern Europe, 1400-1750 9
Development of European Culture 9
Between Revolutions: The Development of Modern Latin America 9
African History: Earliest Times to 1780 9
Origins of the Arab-Israeli Conflict, 1880-1948 9
Arab-Israeli Conflict and Peace Process since 1948 9
Development of American Culture 9
African American History: Africa to the Civil War 9
African American History: Reconstruction to the Present 9
Irish History (must be paired with another humanities course for 9 units or more) 6

The Last Emperors: Chinese History and Society, 1600-1900 9
Modern China: From the Birth of Mao ... to Now 9
Russian History: From the First to the Last Tsar 9
Europe and the World since 1800 9
Photography, the First 100 Years, 1839-1939 9
Roots of Rock & Roll 9
Early Christianity 9
The Arts in Pittsburgh 9
Music and Society in 19th and 20th Century Europe and the U.S. 9

Introduction to Philosophy 9
Ancient Philosophy 9
Modern Philosophy 9
Continental Philosophy 9
Analytic Philosophy 9
Pragmatism 9
Empiricism and Rationalism 9
Philosophy of Religion 9
Introduction to Japanese Language and Culture 9

Introduction to Russian Culture 9
Introduction to French Culture 9
The Francophone World 9
The Emergence of the German Speaking World 9
Introduction to Chinese Language and Culture 9
Spain: Language and Culture 9
Latin America: Language and Culture 9
U.S. Latinos: Language and Culture 9
Introduction to Hispanic Literary and Cultural Studies 9

C. Humanities and Arts Electives (minimum 27 units)

Complete 3 non-technical courses of at least 9 units each from any of the departments in the College of Humanities & Social Sciences or the College of Fine Arts. Some of the courses taught in these units are considered technical courses and may not be used to satisfy this requirement. Students may combine humanities/arts courses with lower units together to form a single course of 9 units or more. Additionally, a select set of courses from Business Administration and from Environmental and Public Policy can also count for this requirement. The complete list of additions and deletions can be found at http://www.csd.cs.cmu.edu/content/bscs-humanities-and-arts-requirements. Consult with a CS undergraduate advisor for additional information.

Students are encouraged, but not required, to take courses from different departments to gain additional breadth and to create new opportunities for engagement with the university community.

Required Minor

A sequence of courses prescribed by the requirements of the particular department. Completion of an additional major (or dual degree) also satisfies this requirement. In general, courses taken in satisfaction of the minor or second major may also count toward any general education category in the CS major (i.e. courses outside of the Computer Science and Mathematics requirements). Double counting toward Computer Science and Mathematics courses in the CS major is strictly limited and depends on the chosen minor (or additional major). Consult with a CS undergraduate advisor and an advisor from the department of the minor (or additional major) for specific restrictions on double counting.

Computing @ Carnegie Mellon

The following course is required of all students to familiarize them with the campus computing environment:

99-10x Computing @ Carnegie Mellon 3

Free Electives

A free elective is any Carnegie Mellon course. However, a maximum of nine (9) units of Physical Education and/or Military Science (ROTC) and/or Student-Led (StuCo) courses may be used toward fulfilling graduation requirements.

Summary of Degree Requirements:

<table>
<thead>
<tr>
<th>Area</th>
<th>Courses</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Science</td>
<td>14</td>
<td>135</td>
</tr>
<tr>
<td>Mathematics</td>
<td>5</td>
<td>49</td>
</tr>
<tr>
<td>Science/Engineering</td>
<td>4</td>
<td>36</td>
</tr>
<tr>
<td>Humanities/Arts</td>
<td>7</td>
<td>63</td>
</tr>
<tr>
<td>Minor Requirement/Free electives</td>
<td>8</td>
<td>74</td>
</tr>
<tr>
<td>Computing @ Carnegie Mellon</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>360</strong></td>
<td><strong>360</strong></td>
</tr>
</tbody>
</table>

Sample Course Sequence

Freshman Year:

Fall

15-122 Principles of Imperative Computation 10
15-128 Freshman Immigration Course 1
15-131 Great Practical Ideas for Computer Scientists 2
15-151 Mathematical Foundations for Computer Science (if not offered, substitute 21-127) 10
Minimum number of units required for the degree:
The flexibility in the curriculum allows many different schedules, of which
the above is only one possibility. Some elective courses are offered only
once per year (Fall or Spring). Constrained CS electives (algorithms/
complexity, logic/languages, systems and applications) may be taken
in any order and in any semester if prerequisites are met and seats are
available. Constrained electives are shown in the specific semesters in
the schedule above as an example only. Students should consult with
their academic advisor to determine the best elective options depending
on course availability, their academic interests and their career goals.
Additionally, the School of Computer Science offers an Additional Major in
Human–Computer Interaction and an Additional Major in Robotics, as well as
numerous computing-oriented minors available to majors and non-majors
alike.

Undergraduate Research Thesis
Students considering going on to graduate school in Computer Science
should take a wide variety of Computer Science and Mathematics courses,
as well as consider getting involved in independent research as early as
possible. This would be no later than the junior year and can begin even
earlier. Students interested in graduate school are strongly encouraged to
participate in the Undergraduate Research Thesis program. Additionally,
graduate CS courses can be taken with permission of the instructor and in
consultation with an academic advisor.

The goal of the Undergraduate Research Thesis Program is to introduce
students to the breadth of tasks involved in independent research, including
library work, problem formulation, experimentation, analysis, writing and
speaking. In particular, students write a survey paper summarizing prior
results in their desired area of research, present a public poster session in
December of their senior year describing their current progress, present
their final results in an oral summary in the year-end university-wide
Undergraduate Research Symposium (Meeting of the Minds) and submit
a written thesis at the end of their senior year. Students work closely with
faculty advisors to plan and carry out their research. The SCS Honors
Undergraduate Research Thesis (15-599) typically starts in the fall semester
of the senior year, and spans the entire senior year. Students receive
a total of 36 units of academic credit for the thesis work, 18 units per
semester. Up to 18 units can be counted toward CS elective requirements
(9 per semester for 2 semesters maximum). Students should prepare their
research prospectus (i.e. proposal of work) during the spring semester of
their junior year, and students in this program are advised to plan their
dates carefully to ensure there is ample time to perform the required
research for the thesis during the senior year.

Students interested in research are urged to consult with their CS
undergraduate advisor and Assistant Dean no later than the end of the
sophomore year in order to plan their workload effectively. Students may
also consider using Research and Innovation in Computer Science (15-300, 9 units) as their technical communications requirement in their junior year
since this course will introduce students to various research projects going
on in the School of Computer Science that may lead to a senior thesis. This
course leads to a subsequent Research Practicum in Computer Science
(15-400, 12 units) that begins to build the foundation for a senior thesis by
starting preliminary work toward the thesis.

Computer Science Additional Majors
and Minors
The School of Computer Science (SCS) offers an Additional Major in
Computer Science, Human-Computer Interaction, and Robotics. SCS offers
majors in Computational Biology (jointly with the Mellon College of Science)
and Statistics and Machine Learning (jointly with the Dietrich College
of Humanities and Social Sciences). SCS also offers Minors in Computer
Science, Computational Biology, Human-Computer Interaction, Language
Technologies, Machine Learning, Neural Computation, Robotics, and
Software Engineering.

To see information for the additional majors and minors other than
Computer Science, see Additional Majors and Minors in SCS (http://
coursescatalog.web.cmu.edu/schoolofcomputerscience/addmajorsminors).

Computer Science Minor
Students interested in pursuing a minor in Computer Science should first
consult with an advisor in the CS Undergraduate Office after completion of
the prerequisites, 15-122, 15-150 and at least one of the 200-level required
courses. Students are expected to complete all courses for the minor with
an average QPA of 2.5 or higher.

The following courses are required for the Minor in Computer Science:

<table>
<thead>
<tr>
<th>Course</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>15-112 Fundamentals of Programming and Computer Science</td>
<td>12</td>
</tr>
</tbody>
</table>

(some students may need to take 15-110 prior to 15-112 for additional preparation)
Computer Science Additional Major

Students interested in pursuing an additional major in Computer Science should first consult with an advisor in the CS Undergraduate Office. Students are expected to complete the requirements for the CS minor first before continuing on to the additional major. Completion of the CS additional major requires 5 mathematics courses, 12 computer science courses (not including 15-110 and 15-112 if needed), and 1 technical communication course. Students are expected to complete all courses for the additional major with an average QPA of 3.0 or higher.

Declarative for the additional major is allowed only after all math requirements are completed and at least 75% (9 of 12) of the CS requirements (core and electives) are completed. Seats in upper-level CS courses are not guaranteed for additional majors so students should plan to be flexible in selecting constrained and general electives. Acceptance to complete a Computer Science additional major is not guaranteed and depends on student performance and seat availability.

The following courses are required for the Additional Major in Computer Science:

<table>
<thead>
<tr>
<th>Math Requirements:</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>21-120 Differential and Integral Calculus</td>
<td>10</td>
</tr>
<tr>
<td>21-122 Integration and Approximation</td>
<td>10</td>
</tr>
<tr>
<td>21-127 Concepts of Mathematics</td>
<td>10</td>
</tr>
</tbody>
</table>

All of the following courses:

| 21-241 Matrices and Linear Transformations | 10 |
| 21-242 Matrix Theory | 10 |

All of the following required Computer Science courses:

| 15-122 Principles of Imperative Computation (co-requisite: 21-127; students with no prior programming experience must take 15-112 before 15-122; some students may need to take 15-110 prior to 15-112 for additional preparation.) | 10 |
| 15-150 Principles of Functional Programming | 10 |
| 15-210 Parallel and Sequential Data Structures and Algorithms | 12 |
| 15-213 Introduction to Computer Systems | 12 |
| 15-210 Principles of Functional Programming | 10 |
| 15-462 Computational Discrete Mathematics | 12 |
| 15-451 Modern Computer Algebra | 9 |
| 15-456 Formal Languages, Automata, and Computability | 9 |
| 15-455 Undergraduate Complexity Theory | 9 |
| 15-456 Computational Geometry | 9 |
| 21-301 Combinatorics | 9 |
| 21-484 Graph Theory | 9 |

Note: Students who take 15-213/18-213 or 15-251 as part of another major are required to replace this requirement in the CS minor with another CS elective (15-xxx) as defined above, for a total of 3 additional CS electives.

Double-Counting Restriction

Students pursuing a Minor in Computer Science must complete at least 6 courses from their home department, of at least 9 units each, none of which are used toward the Computer Science minor. Additionally, students may double-count a maximum of 3 courses for the CS minor toward all other majors and minors. Students, especially from interdisciplinary majors or with multiple majors or minors, are urged to consult with the Assistant Dean or Undergraduate Program Coordinator in the CS Undergraduate Office to determine double-counting restrictions specific to their own situations.
Undergraduate Program Coordinator in the CS Undergraduate Office to determine double-counting restrictions specific to their own situations.

Dual Degree in Computer Science

Students wishing to pursue a Dual Degree in Computer Science are required to apply in the same way as students wishing to transfer into the Computer Science major. Details are given in the Policies section below. Besides the student’s primary degree requirements, a student accepted for Dual Degree in CS is required to complete at least 450 units in total and meet all requirements for the CS major including general education requirements (humanities/arts and science/engineering). Since the CS major requires at least a minor in another area, the student’s primary major will substitute for this requirement. Note that the primary major must be completed prior to or at the same time as the dual degree in CS to satisfy the minor requirement; a dual degree in CS cannot be certified if the primary degree is not completed. Students should consult with the Assistant Dean in the CS Undergraduate Office and/or their CS academic advisor to review all requirements, once approved.

Double-Counting Restriction

Students pursuing a Dual Degree in Computer Science must complete at least 6 courses from their home department, of at least 9 units each, none of which are used toward the technical requirements (computer science core and electives, and mathematics) of the Computer Science major. Students, especially from interdisciplinary majors or with multiple majors or minors, are urged to consult with the Assistant Dean or Undergraduate Program Coordinator in the CS Undergraduate Office to determine double-counting restrictions specific to their own situations.

Additional Majors and Minors in the School of Computer Science

Besides Computer Science, the School of Computer Science offers additional majors in:
- Human-Computer Interaction
- Robotics

Besides Computer Science, the School of Computer Science also offers minors in:
- Computational Biology
- Human-Computer Interaction
- Language Technologies
- Machine Learning
- Neural Computation
- Robotics
- Software Engineering

To see information for the additional majors and minors other than Computer Science, see Additional Majors and Minors in SCS (http://coursecatalog.web.cmu.edu/schoolofcomputerscience/addmajorsminors).

SCS Policies & Procedures

School of Computer Science (SCS) Academic Standards and Actions

Grading Practices

Grades given to record academic performance in SCS are detailed under Grading Practices at http://coursecatalog.web.cmu.edu/servicesandoptions/undergraduateacademicregulations/

Dean’s List

SCS recognizes each semester those undergraduates who have earned outstanding academic records by naming them to the Dean’s List. The criterion for such recognition is a quality point average of at least 3.75 while completing a minimum of 36 factorable units and earning no incomplete grades.

Academic Actions

In the first year, quality point averages below 1.75 in either semester invoke an academic action. For all subsequent semesters an academic action will be taken if the semester quality point average or the cumulative quality point average (excluding the first year) is below 2.00.

Probation: The action of probation will be taken in the following cases based on QPA:
1. One semester of the first year is below 1.75 QPA;
2. The semester QPA of a student in good standing beyond the first year falls below 2.00.

The term of probation is one semester as a full-time student. First year students are no longer on probation at the end of the second semester if the second semester’s QPA is 1.75 or above. Students in the third or subsequent semester of study are no longer on probation at the end of one semester if the semester QPA and cumulative QPA (excluding the first year) are 2.00 or above.

Probation Continued: A student who has had one semester on probation and is not yet meeting minimum requirements but whose record indicates that the standards are likely to be met at the end of the next semester of study is occasionally continued on probation. This action is normally taken only when a student’s semester QPA is above 2.0 but their cumulative QPA is not yet above 2.0.

Suspension: A student who does not meet minimum standards based on QPA at the end of one semester of probation will be suspended:
- A first year student will be suspended if the QPA from each semester is below 1.75.
- A student on probation in the third or subsequent semester of study will be suspended if the semester QPA is below 2.0.

The minimum period of suspension is one academic year (two non-summer semesters). At the end of that period a student may return to school (on probation) by:
1. completing a Return from Leave form from the HUB,
2. submitting an additional written statement to the SCS Assistant Dean for Undergraduate Education, minimum one page, that outlines what the student did while on leave to address the issues that led to the suspension and that would indicate future success on return, and
3. written approval from the student’s academic advisor and the Assistant Dean for Undergraduate Education, in consultation with the Office of Student Affairs and the Office of International Education as appropriate.

Students who have been suspended or have withdrawn are required to absent themselves from the campus (including residence halls and Greek houses) within a maximum of two days after the action and to remain off the campus for the duration of the time specified. This action includes debarment from part-time or summer courses at the university for the duration of the period of the action. Although suspended students may not hold student jobs, students on academic suspension may, under certain circumstances, have a non-student job with the university. Students on disciplinary or administrative suspension may not.

Drop: This is a permanent severance. Students who have been suspended and who fail to meet minimum standards in the semester that they return to school will be dropped.

Students who have been dropped are required to absent themselves from the campus (including residence halls and Greek houses) within a maximum of two days after the action.

Other Actions: In addition to academic actions based on QPA, the Associate Dean for Undergraduate Education may place students on probation, or subsequent suspension, if they do not demonstrate reasonable progress through the core CS curriculum (e.g., not completing a CS core class after 3 attempts, or not completing the required CS 100-level core courses by the end of the sophomore year). Students are encouraged to consult with their academic advisor about any concerns with regard to lack of progress in the CS major.

The relation indicated above between probation, suspension and drop is nominal. In unusual circumstances, SCS College Council may suspend or drop a student without prior probation.

Return from Leave of Absence

SCS undergraduate students returning from a leave of absence are required to submit a Return from Leave of Absence form to the CS Undergraduate Office for approval by the student’s academic advisor and the SCS Assistant Dean for Undergraduate Education. In addition, the student must also supply a letter that explains the reason for the leave, the actions that were performed during the leave to prepare the student for a successful return, and a description of the on-campus resources, if required, that would be used by the student in order to increase the likelihood of success. Students returning from a leave are also encouraged to provide two letters of support from people close to the student (e.g. family, friends, clergy, teachers, coaches, others as appropriate). Requests to return are reviewed by the student’s academic advisor, the Assistant Dean and the Student Affairs liaison to determine eligibility and any resources that need to be put into
Transfer into SCS / CS Dual-degree

Undergraduate students admitted to colleges at CMU other than SCS and wishing to transfer into SCS or wishing to pursue a dual degree in Computer Science should consult with the Assistant Dean for Undergraduate Education during their first year. Students must complete in 21-127, 15-122, 15-150, 15-210, 15-213, 15-251 with an overall QPA over these six courses of 3.6 or higher in order to apply for transfer or dual degree. Students may apply for transfer by the mid-semester break in the semester when the last of the six required courses will be completed. In the case of courses in progress, the midsemester grades will be used in the QPA calculation. The decision to allow transfer or dual degree will be made by committee based on the student’s academic performance (in the specified courses and in their courses overall if necessary), additional involvement in SCS and other computing-related activities, and availability of space in the student’s class level. Students should consult the CS Undergraduate Program office for complete information concerning minimum requirements, instructions and deadlines.

Procedure for transfer of students from another university into SCS: A student should first apply through the Office of Admission. If the Office of Admission believes the applicant is acceptable, the student’s record is sent to SCS for evaluation. Admission is based on seat availability, overall academic performance from the student’s current institution, and the application material. It is important to note that extremely few external transfers are admitted to the SCS program at Carnegie Mellon University.

Graduation Requirements

1. A requirement for graduation is the completion of the program specified for a degree with a cumulative quality point average of 2.00 or higher for all courses taken after the first year.
2. Students must be recommended for a degree by the faculty of SCS.
3. A candidate for the bachelor’s degree must complete at the University a minimum of four quarters of full-time study, or the equivalent of part-time study, comprising at least 180 units of course work.
4. Students will be required to have met all financial obligations to the university before being awarded a degree.

Modification of Graduation Requirements: A student may seek permission to modify graduation requirements by petition to the SCS College Council.

Faculty

UMUT ACAR, Associate Professor, Computer Science Department – Ph.D., Carnegie Mellon University; Carnegie Mellon, 2012–.
ANIL ADA, Assistant Teaching Professor, Carnegie Mellon University – Ph.D., McGill University; Carnegie Mellon, 2014–.
YUVRaj AGARWAL, Assistant Professor, Institute for Software Research – Ph.D., University of California, San Diego; Carnegie Mellon, 2013–.
JONATHAN ALDRICH, Assistant Professor, Computer Science Department – Ph.D., University of Washington; Carnegie Mellon, 2003–.
VINCENT ALEVEN, Associate Professor, Human-Computer Interaction Institute – Ph.D., University of Pittsburgh; Carnegie Mellon, 2000–.
OMEAD AMIDI, Senior Systems Scientist, Robotics Institute – Ph.D., Carnegie Mellon University; Carnegie Mellon, 1994–.
DAVID ANDERSEN, Associate Professor, Computer Science Department – Ph.D., Massachusetts Institute Of Technology; Carnegie Mellon, 2005–.
JOHN ANDERSON, R.K. Mellon University Professor – Ph.D., Stanford University; Carnegie Mellon, 1978–.
DIMITRIOS APOSTOLOPOULOS, Senior Systems Scientist, Robotics Institute – Ph.D., Carnegie Mellon University; Carnegie Mellon, 1989–.
CHRISTOPHER ATKESON, Professor, Robotics Institute – Ph.D., Massachusetts Institute Of Technology; Carnegie Mellon, 2000–.
MATT BAGNELL, Associate Professor, Robotics Institute – Ph.D., Carnegie Mellon University; Carnegie Mellon, 2004–.
MARIA FLORINA BALCAN, Associate Professor, Machine Learning Department – Ph.D., Carnegie Mellon University; Carnegie Mellon, 2014–.
STEVENIE BALZER, Systems Scientist, Carnegie Mellon University – Ph.D., ETH Zurich; Carnegie Mellon, 2016–.
ZIV BAR-JOSEPH, Professor, Computational Biology Department – Ph.D., Massachusetts Institute Of Technology; Carnegie Mellon, 2003–.
MATTHEW BASS, Assistant Teaching Professor, Institute for Software Research – M.S., Carnegie Mellon University; Carnegie Mellon, 2012–.
LUJO BAUER, Associate Professor, Institute for Software Research – Ph.D., Princeton University; Carnegie Mellon, 2016–.
MIKE BECKMANN, Assistant Professor, Computer Science Department – Ph.D., Massachusetts Institute Of Technology; Carnegie Mellon, 2017–.
TAYLOR BERG-KIRKPATRICK, Assistant Professor, Computer Science Department – Ph.D., University of California at Berkeley; Carnegie Mellon, 2016–.
KAREN BERNTSEN, Associate Teaching Professor, Human Computer Interaction Institute – Ph.D., Duquesne University; Carnegie Mellon, 2005–.
JEFFREY BIGHAM, Associate Professor,Human-Computer Interaction Institute – Ph.D., University of Washington; Carnegie Mellon, 2013–.
ALAN BLACK, Professor, Language Technologies Institute – Ph.D., University Of Edinburgh; Carnegie Mellon, 1999–.
GUY BLELLOCH, Associate Dean for Undergraduate Education and Professor, Computer Science Department – Ph.D., Massachusetts Institute Of Technology; Carnegie Mellon, 1988–.
AVI RUBIN BLUM, Professor, Computer Science Department – Ph.D., Massachusetts Institute Of Technology; Carnegie Mellon, 1991–.
LENORE BLUM, Distinguished Career Professor, Computer Science Department – Ph.D., Massachusetts Institute Of Technology; Carnegie Mellon, 1999–.
MANUEL BLUM, University Professor, Computer Science Department – Ph.D., Massachusetts Institute Of Technology; Carnegie Mellon, 1999–.
DAVID BOURNE, Principal Systems Scientist, Robotics Institute – M.S., University Of Pennsylvania; Carnegie Mellon, 1980–.
DANIEL BOYARSKI, Professor – M.F.A., Indiana University; Carnegie Mellon, 1982–.
TRAVIS BREAUX, Associate Professor, Institute for Software Research – Ph.D., North Carolina State University; Carnegie Mellon, 2004–.
STEPHEN BROOKES, Professor, Computer Science Department – Ph.D., University College; Carnegie Mellon, 1981–.
RALF BROWN, Principal Systems Scientist, Language Technologies Institute – Ph.D., Carnegie Mellon University; Carnegie Mellon, 1993–.
EMMA BRUNSKILL, Assistant Professor, Computer Science Department – Ph.D., Massachusetts Institute Of Technology; Carnegie Mellon, 2011–.
RANDAL BRYANT, University Professor, Computer Science Department – Ph.D., Massachusetts Institute Of Technology; Carnegie Mellon, 1984–.
JAMES CALLAN, Professor, Language Technologies Institute – Ph.D., University Of Massachusetts; Carnegie Mellon, 1999–.
JAVIER CAMARAMORENO, Systems Scientist, Institute for Software Research – Ph.D., University Of Malaga; Carnegie Mellon, 2015–.
JAIME CARBONELL, University Professor and Director, Language Technologies Institute – Ph.D., Yale University; Carnegie Mellon, 1979–.
KATHLEEN CARLEY, Professor, Institute for Software Research – Ph.D., Harvard University; Carnegie Mellon, 1984–.
JACOBO CARRASQUEL, Associate Teaching Professor, Computer Science Department – M.S., Carnegie Mellon University; Carnegie Mellon, 1980–.
JUSTINE CASSELL, Professor, Human-Computer Interaction Institute – Ph.D., University Of Chicago; Carnegie Mellon, 2010–.
ILIANO CERVESATO, Teaching Professor, Computer Science Department – Ph.D., University Of Toronto; Carnegie Mellon, 2016–.
HOWARD CHOSET, Professor, Robotics Institute – Ph.D., California Institute Of Technology; Carnegie Mellon, 1996–.
MICHAEL CHRISTEL, Teaching Professor, Entertainment Technology Center – Ph.D., Georgia Institute Of Technology; Carnegie Mellon, 1987–.
EDMUND CLARKE, University Professor, Emeritus, Computer Science Department – Ph.D., Carnegie Mellon, 1983–.
WILLIAM COHEN, Professor, Machine Learning Department – Ph.D., Rutgers University; Carnegie Mellon, 2003–.
PHILLIP COMPEAU, Assistant Teaching Professor, Computational Biology Department – Ph.D., University Of California, San Diego; Carnegie Mellon, 2015–.
ALBERT CORBETT, Associate Research Professor Emeritus, Human-Computer Interaction Institute – Ph.D., University Of Oregon; Carnegie Mellon, 1963–.
STELIAN COROS, Assistant Professor, Robotics Institute – Ph.D., University Of British Columbia; Carnegie Mellon, 2015–.
THOMAS CORTINA, Assistant Dean for Undergraduate Education and Teaching Professor, Computer Science Department – Ph.D., Polytechnic University; Carnegie Mellon, 2009–.
KEENAN CRANE, Assistant Professor, Robotics Institute – Ph.D., California Institute Of Technology; Carnegie Mellon, 2015–.
LORRIE CRANOR, Professor, Institute for Software Research – Ph.D., Washington University; Carnegie Mellon, 2008–.
KARL CRAY, Associate Professor, Computer Science Department – Ph.D., Cornell University; Carnegie Mellon, 1998–.
LAURA DABISH, Associate Professor, Human Computer Interaction Institute – Ph.D., Carnegie Mellon University; Carnegie Mellon, 1998–.
ROGER DANNENBERG, Professor, Computer Science Department – Ph.D., Carnegie Mellon University; Carnegie Mellon, 2007–.
FERNANDO DE LA TORRE FRADE, Associate Research Professor, Robotics Institute – Ph.D., La Salle School of Engineering; Carnegie Mellon, 2002–.
ANIND DEY, Professor and Director, Human-Computer Interaction Institute – Ph.D., Georgia Institute Of Technology; Carnegie Mellon, 2001–.
GIANNI DI CARO, Assistant Teaching Professor, Computer Science Department – Ph.D., Universite Libre De Bruxelles; Carnegie Mellon, 2016–.
JOHN DOLAN, Principal Systems Scientist, Robotics Institute – Ph.D., Carnegie Mellon University; Carnegie Mellon, 1991–.

STEVEN DOW, Assistant Professor, Human-Computer Interaction Institute – Ph.D., Georgia Institute of Technology; Carnegie Mellon, 2011–.

ARTUR DUBRAWSKI, Senior Systems Scientist, Robotics Institute – Ph.D., Institute of Fundamental Technological Research; Carnegie Mellon, 2003–.

CHRISTOPHER DYER, Assistant Professor, Language Technologies Institute – Ph.D., University of Maryland; Carnegie Mellon, 2012–.

DAVID ECKHARDT, Teaching Professor, Computer Science Department – Ph.D., Carnegie Mellon University; Carnegie Mellon, 2003–.

WILLIAM EDDY, Professor – Ph.D., Yale University; Carnegie Mellon, 1976–.


MARTIAL HEBERT, Professor and Director, Robotics Institute – Ph.D., Paris-XI; Carnegie Mellon, 1984–.

JAMES HERBSLEB, Professor, Institute for Software Research – Ph.D., University Of Nebraska; Carnegie Mellon, 2002–.

JESSICA HODGINS, Professor, Robotics Institute – Ph.D., Carnegie Mellon University; Carnegie Mellon, 2001–.

JAN HOFFMANN, Assistant Professor, Computer Science Department – Ph.D., Ludwig-Maximilians-Universität and TU Munich; Carnegie Mellon, 2015–.

RALPH HOLZIS, Research Professor, Robotics Institute – Ph.D., University Of Colorado; Carnegie Mellon, 1993–.

JASON HONG, Associate Professor, Human-Computer Interaction Institute – Ph.D., University Of California at Berkeley; Carnegie Mellon, 2004–.

EDUARDO HOVY, Research Professor, Language Technologies Institute – Ph.D., Yale University; Carnegie Mellon, 2012–.

DANIEL HUBER, Senior Systems Scientist, Robotics Institute – Ph.D., Carnegie Mellon University; Carnegie Mellon, 2002–.

SCOTT HUSON, Professor, Human-Computer Interaction Institute – Ph.D., University Of Colorado; Carnegie Mellon, 2002–.

FARNAH JAHANIAN, Provost, Carnegie Mellon University, and Professor, Computer Science Department – Ph.D., University of Texas at Austin; Carnegie Mellon, 2014–.

ANGEL JORDAN, University Professor Emeritus, Robotics Institute – Ph.D., Stanford University; Carnegie Mellon, 1985–.

MAYANK GOEL, Assistant Professor – Ph.D., University of California at Los Angeles; Carnegie Mellon, 1985–.

JOHN MACKEY, Teaching Professor, Computer Science Department and Pennsylvania State University; Carnegie Mellon, 2016–.
SIDHARTHA SRINIVASA, Associate Professor, Robotics Institute – Ph.D., Carnegie Mellon University; Carnegie Mellon, 2011–.

JOHN STAMPER, Assistant Professor, Human-Computer Interaction Institute – Ph.D., University of North Carolina At Charlotte; Carnegie Mellon, 2009–.


PETER STEENKISTE, Professor, Computer Science Department – Ph.D., Stanford University; Carnegie Mellon, 1987–.

MARK STEHLIK, Teaching Professor, Computer Science Department – B.S., Pace University; Carnegie Mellon, 1981–.

AARON STEINFELD, Associate Research Professor, Robotics Institute – Ph.D., University of Michigan; Carnegie Mellon, 2001–.

ANTHONY STENTZ, Research Professor, Robotics Institute – Ph.D., Carnegie Mellon University; Carnegie Mellon, 1989–.

GEORGE STETTEN, Adjunct Research Professor, Robotics Institute – Ph.D., University of North Carolina; Carnegie Mellon, 1999–.


KLAUS SUTNER, Teaching Professor, Computer Science – Ph.D., University of Munich; Carnegie Mellon, 1995–.

KATIA SYCARA, Research Professor, Robotics Institute – Ph.D., Georgia Institute of Technology; Carnegie Mellon, 1987–.

SUJATA TELANG, Associate Teaching Professor, Institute for Software Research – Ph.D., Carnegie Mellon University; Carnegie Mellon, 2004–.

ANTHONY TOMASIC, Senior Systems Scientist, Institute for Software Research – Ph.D., Princeton University; Carnegie Mellon, 2003–.

DAVID TOURETZKY, Research Professor, Computer Science Department – Ph.D., Carnegie Mellon University; Carnegie Mellon, 1984–.

BOGDAN VASILESCU, Assistant Professor, Institute for Software Research – Ph.D., Eindhoven University of Technology; Carnegie Mellon, 2016–.

MANUELA VELOSO, University Professor, Computer Science, and Head, Machine Learning Department – Ph.D., Carnegie Mellon University; Carnegie Mellon, 1992–.

JOHN VU, Distinguished Career Professor, Computational Biology Department – M.S., Carnegie Mellon University; Carnegie Mellon, 2011–.

HOWARD WACTLAR, Research Professor, Computer Science Department – M.S., University of Maryland; Carnegie Mellon, 1967–.

ALEXANDER WAIBEL, Professor, Language Technologies Institute – Ph.D., Carnegie Mellon University; Carnegie Mellon, 1988–.

LARRY WASSERMAN, Professor, Machine Learning Department – Ph.D., University of Toronto; Carnegie Mellon, 1988–.

LEE WEISS, Research Professor, Robotics Institute – Ph.D., Carnegie Mellon University; Carnegie Mellon, 1983–.


DAVID WETTERGREEN, Research Professor, Robotics Institute – Ph.D., Carnegie Mellon University; Carnegie Mellon, 2000–.

WILLIAM RED WHITTAKER, Fredkin University Research Professor, Robotics Institute – Ph.D., Carnegie Mellon University; Carnegie Mellon, 1973–.

WEI WU, Associate Research Professor, Computational Biology Department – Ph.D., Rutgers University; Carnegie Mellon, 2011–.

POE ERIC XING, Professor, Machine Learning Department – Ph.D., University of California At Berkeley; Carnegie Mellon, 2004–.

MIN XU, Assistant Research Professor, Computational Biology Department – Ph.D., University of Southern California; Carnegie Mellon, 2016–.

JEAN YANG, Assistant Professor, Computer Science Department – Ph.D., Massachusetts Institute of Technology; Carnegie Mellon, 2016–.

YIMING YANG, Professor, Language Technologies Institute – Ph.D., Kyoto University; Carnegie Mellon, 1996–.

JOHN ZIMMERMAN, Associate Professor, Human-Computer Interaction Institute – M.Des., Carnegie Mellon University; Carnegie Mellon, 2002–.