Computer Science Program

Srinivasan Seshan, Department Head, Computer Science
Location: GHC 7019

Mark Stehlik, Program Director, Assistant Dean for Outreach
Location: GHC 6205

Amy Weis, Program Coordinator, CS Undergraduate Office
Location: GHC 4115
www.csd.cs.cmu.edu (http://www.csd.cs.cmu.edu)

The B.S. program in Computer Science combines a solid core of Computer Science courses with the ability to gain additional depth through a required minor in a second subject or a concentration in a computing area. In addition, the curriculum provides breadth through 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.

Students apply to, and are directly admitted into, the School of Computer Science. Admitted students may choose to pursue an undergraduate degree 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.

Students in the B.S. program in Computer Science are expected to acquire the following skills upon graduation:

- Identify, use, design, develop and analyze appropriate abstractions and algorithms to solve problems while being able to prove the algorithm's performance and correctness across a variety of metrics (e.g., time, space, parallel vs. sequential implementation, computability).
- Implement solutions to problems in domains such as artificial intelligence, graphics and sound, software engineering, and human-computer interaction, by applying the fundamentals of those areas to create solutions to current problems while being exposed to research developments that will enable them to adapt as the technology changes.
- Reason about and implement programs in various programming languages and paradigms
- Describe, specify, and develop large-scale, open-ended software systems subject to constraints such as performance and/or resource issues
- Communicate technical material effectively to technical and non-technical audiences
- Work both individually and in teams
- Recognize the social impact of computing and the attendant responsibility to consider the legal, moral and ethical implications of computing technologies.

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.

SCS also offers a B.S. degree in Artificial Intelligence, a B.S. degree in Computational Biology, a B.S. degree in Human-Computer Interaction, and a Bachelor's Degree in Computer Science and the Arts (jointly with the College of Fine Arts). More detail about the Artificial Intelligence major, the Computational Biology major and the Computer Science and the Arts program is available in separate sections of the Undergraduate Catalog. 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. Information about additional majors and minors in SCS besides those in Computer Science are listed in a separate section in the Undergraduate Catalog.

Curriculum - B.S. in Computer Science

The following requirements are for students entering Fall 2023.

Computer Science

Computer Science Core (all of the following):		Units
07-128	First Year Immigration Course	3
15-122	Principles of Imperative Computation (students without credit or a waiver for 15-112, Fundamentals of Programming and Computer Science, must take 15-112 before 15-122)	12
15-150	Principles of Functional Programming	12
15-210	Parallel and Sequential Data Structures and Algorithms	12
15-213	Introduction to Computer Systems	12
15-251	Great Ideas in Theoretical Computer Science	12
15-451	Algorithm Design and Analysis	12

One Artificial Intelligence elective (min. 9 units). Students will be able to tackle complex, real-world problems using techniques from Artificial Intelligence, including symbolic and probabilistic reasoning, machine learning, optimization, and perception.

10-315	Introduction to Machine Learning (SCS Majors)	12
11-411	Natural Language Processing	12
11-485	Introduction to Deep Learning	9
15-281	Artificial Intelligence: Representation and Problem Solving	12
15-386	Neural Computation	9
16-384	Robot Kinematics and Dynamics	12
16-385	Computer Vision	12
others as designated by the CS Undergraduate Program		

One Domains elective (min. 9 units). Students will gain expertise in fundamental principles from a larger domain of computer science not already represented by other constrained categories, currently logic and languages, systems, and artificial intelligence (which includes machine learning, language technologies, and robotics). Students will be able to apply theoretical and computational techniques from the Computer Science core to an introductory study of another major subarea of Computer Science.

02-251	Great Ideas in Computational Biology	12
05-391	Designing Human Centered Software	12
11-324	Human Language for Artificial Intelligence	12
15-322	Introduction to Computer Music	9
15-330	Introduction to Computer Security	12
15-362	Computer Graphics	12
15-455	Undergraduate Complexity Theory	9
17-313	Foundations of Software Engineering	12
others as designated by the CS Undergraduate Program		

One Logics/Languages elective (min. 9 units). Students will master techniques for rigorous, formal reasoning about programs or systems, rooted in their logical foundations.

15-311	Logic and Mechanized Reasoning	9
15-312	Foundations of Programming Languages	12
15-316	Software Foundations of Security and Privacy	9
15-317	Constructive Logic	9
15-414	Bug Catching: Automated Program Verification	9
17-355	Program Analysis	12
17-363	Programming Language Pragmatics	12
80-413	Category Theory	9
others as designated by the CS Undergraduate Program		

One Software Systems elective (min. 12 units). Students will: 1. be able to describe how the properties of modern hardware (e.g., processor architecture, networks, storage) influence the design and implementation of software systems, such as through reasoning about concurrency and performance. 2. be able to analyze failures and / or resource limitations of physical systems and plan for their mitigation or management. 3. be able to develop abstractions based on lower-level primitives to manage the failures or other difficulties inherent in working with hardware. 4. demonstrate their learning through significant project / system implementations, requiring both course-specific knowledge as well as general system-building skills (i.e., not just programming, but also design, debugging, testing, etc.). The programming tasks together constitute a significant fraction of the course grade (e.g., 40% or more).

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	Networking and the Internet	12
15-445	Database Systems	12

others as designated by the CS Undergraduate Program

Two School of Computer Science electives:	Units
These electives can be from any SCS department (Computer Science [15-], Computational Biology [02-], Human-Computer Interaction [05-], Machine Learning [10-], Language Technologies [11-], Robotics [16-], or Software & Societal Systems [17-]). They must be 200-level or above and at least 9 units each, with the following exceptions:	18
ADDITIONS: 1) students who take two of the major-intro mini-courses (02-180, 05-180, 07-180, or 16-180) during their first year may combine these two mini-courses together to count as one SCS elective; 2) 15-195 and 15-295 can be combined to count as one SCS elective; 3) 07-400 and 07-599 are countable as SCS electives.	
DELETIONS: 1) the following courses do NOT count as SCS electives: 02-201, 02-223, 02-250, 02-261, 05-200, 11-423, 15-351, 16-211, 16-223, 16-224, 16-397, 16-480, 17-200, 17-333, 17-562; 2) some IDeATe courses and some SCS undergraduate and graduate courses might not be allowed based on course content. Always consult with a CS undergraduate advisor before registration to determine eligibility for this requirement.	

Mathematics

All of the following Mathematics courses:

15-151	Mathematical Foundations for Computer Science (if not offered, substitute 21-127 or 21-128)	12
21-120	Differential and Integral Calculus	10
21-122	Integration and Approximation	10
21-241	Matrices and Linear Transformations	11
or 21-242	Matrix Theory	
21-259	Calculus in Three Dimensions	10
or 21-266	Vector Calculus for Computer Scientists	
or 21-268	Multidimensional Calculus	
or 21-269	Vector Analysis	

Plus one of the following four Probability choices:

15-259	Probability and Computing	12
21-325	Probability	9
36-218	Probability Theory for Computer Scientists	9
36-225-36-226	Introduction to Probability Theory - Introduction to Statistical Inference (must take both courses in this sequence to satisfy requirement)	18

Technical Communication

Learning Objectives:

Audience
Students will be able to analyze the audience or audiences of a work: who the stakeholders are, what those stakeholders are trying to accomplish, and how persuasive communication can help students achieve their goals.

Written Mechanics
Students will practice developing their professional writing voice, paying attention to spelling, grammar, style, and the use of visual information.

Adaptation
Students will be able to adapt expert-level information for a general audience, across textual, graphical, and oral presentation.

Genres/Templates
Students will practice developing information in a variety of communication contexts, connecting areas of content in their own voice, and leading the reader through a cogent sequence of ideas while working within the constraints of a particular genre or template.

Peer Review
Students will practice reading and revising the work of themselves and their peers.

Oral Presentation Mechanics
Students will prepare for the real-time act of presenting to an audience, including delivery, practice, venue setup, and managing presentation support (slides, props, etc.).

Oral Presentation Development
Students will explore formal and informal modes of communication, selecting, organizing and explaining information for a real-time audience, and increasing clarity with use of structure and style.

Group Work
Students will practice skills to help them work as a team, stay on track, and produce a group deliverable.

One Technical Communications course:		Units
07-300	Research and Innovation in Computer Science	9
17-200	Ethics and Policy Issues in Computing	9
76-270	Writing for the Professions	9

Science and Engineering

All candidates for the bachelor's degree in Computer Science must complete a minimum of 36 units offered by the Mellon College of Science and/or the College of Engineering (CIT). These courses offer students an opportunity to explore scientific and engineering domains that can influence their effectiveness as computer scientists upon graduation.

Requirements for this component of the degree are listed under the SCS main page under General Education Requirements (http://coursecatalog.web.cmu.edu/schools-colleges/schoolofcomputerscience/#genedtext).

Humanities and Arts

All candidates for the bachelor's degree in Computer Science must complete a minimum of 63 units offered by the College of Humanities & Social Sciences and/or the College of Fine Arts. Some courses from the Tepper School of Business also qualify for this requirement. These courses offer students breadth in their education and perspectives and provide students with a better appreciation of social, artistic, cultural, political and economic issues that can influence their effectiveness as computer scientists upon graduation.

Requirements for this component of the degree are listed under the SCS main page under General Education Requirements (http://coursecatalog.web.cmu.edu/schools-colleges/schoolofcomputerscience/#genedtext).

Required Minor or Concentration

Students completing the bachelor's degree in Computer Science must complete either a minor outside of SCS or a concentration within SCS. A minor is a sequence of (typically 5-6) courses within a particular department to give students a core of a specific discipline but not an entire major of study. Refer to the sections for other CMU colleges for details about available non-SCS minors. An SCS concentration is a sequence of (typically 4-5) courses within an SCS department to give students further depth in specific areas of research important to SCS. SCS concentrations are available only to SCS students and assume that these students have a significant core knowledge in Computer Science including 15-210, 15-213 and 15-251. See the SCS Concentrations section for a list of available concentrations and their requirements. Completion of an additional major (or dual degree) also satisfies this requirement. Students should consult with their academic advisor to plan for their desired minor or concentration starting in the sophomore year.

Double Counting

In general, courses taken in satisfaction of the minor or additional 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). In general, students may double count at most 5 of the 12 core

Computer Science requirements toward all other declared additional majors and minors. Additional majors and minors have their own double counting rules as well. 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 (1 course)

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

99-101	Core@CMU	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:

Area	Courses	Units
Computer Science (core courses, constrained electives, and SCS electives)	12	**125**
Mathematics	6	**58**
Technical Communication	1	**9**
Science/Engineering	4	**36**
Humanities/Arts	7	**63**
Minor or Concentration Requirement/Free electives	Varies	**63**
Computing @ Carnegie Mellon	1	**3**
First Year Seminar	1	**3**
		360

Sample Course Sequence

The sample given below is for a student who already has credit for introductory programming and one semester of calculus. Students with credit for two semesters of calculus may start with a more advanced math class (e.g. 21-241) in their first year. Students with no credit for introductory programming and/or one semester of calculus will take 15-112 and/or 21-120 in their first semester and shift a few courses to later semesters after consulting with their academic advisor; these students should still be able to complete their degree in four years. It is recommended that students keep their academic load lighter for their Senior Fall semester to account for offsite job interviews or for their Senior Spring semester to account for visits to graduate schools.

Freshman Year:

Fall		Units
07-128	First Year Immigration Course	3
07-131	Great Practical Ideas for Computer Scientists (optional, not required for CS major)	2
15-122	Principles of Imperative Computation	12
15-151	Mathematical Foundations for Computer Science (if not offered, substitute 21-127)	12
21-122	Integration and Approximation	10
76-101	Interpretation and Argument	9
99-101	Core@CMU	3
		51

Spring		Units
15-150	Principles of Functional Programming	12
15-213	Introduction to Computer Systems	12
21-259	Calculus in Three Dimensions	10
xx-180	SCS Major Intro Mini	5
xx-xxx	Humanities and Arts Elective	9
		48

Sophomore Year:

Fall		Units
15-210	Parallel and Sequential Data Structures and Algorithms	12
21-241	Matrices and Linear Transformations	11
xx-xxx	Science/Engineering Course	9
xx-xxx	Humanities and Arts Elective	9
xx-xxx	Minor Requirement / Free Elective	9
		50

Spring		Units
15-251	Great Ideas in Theoretical Computer Science	12
xx-xxx	Computer Science: Domains Elective*	9
xx-xxx	Probability Course*	9
xx-xxx	Science/Engineering Course	9
xx-xxx	Humanities and Arts Elective	9
		48

Junior Year:

Fall		Units
15-451	Algorithm Design and Analysis	12
xx-xxx	Computer Science: Logic/Languages Elective*	9
xx-xxx	Technical Communications Course*	9
xx-xxx	Science/Engineering Course	9
xx-xxx	Minor Requirement / Free Elective	9
		48

Spring		Units
15-xxx	Computer Science: Systems Elective*	12
xx-xxx	Computer Science: Artificial Intelligence Elective*	9
xx-xxx	Science/Engineering Course	9
xx-xxx	Humanities and Arts Elective	9
xx-xxx	Minor Requirement / Free Elective	9
		48

Senior Year:

Fall		Units
xx-xxx	School of Computer Science Elective	9
xx-xxx	Humanities and Arts Elective	9
xx-xxx	Minor Requirement / Free Elective	9
xx-xxx	Minor Requirement / Free Elective	9
		36

Spring		Units
xx-xxx	School of Computer Science Elective	9
xx-xxx	Humanities and Arts Elective	9
xx-xxx	Minor Requirement / Free Elective	9
xx-xxx	Minor Requirement / Free Elective	9
		36

Minimum number of units required for the degree: 360

*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 electives (probability, logic/languages, software systems, artificial intelligence and domains) 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.

Undergraduate Research Thesis

CS majors may use the SCS Honors Research Thesis as part of their degree. The SCS Honors Undergraduate Research Thesis (07-599) 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 interested in research may also consider using Research and Innovation

in Computer Science (07-300, 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 (07-400, 12 units) that allows students to complete a small-scale research study or experiment and present a research poster. Students who use 15-400 to start their senior thesis can use these units toward the required 36 units.

For more information about the SCS Honors Research Thesis, refer to the SCS Honors Research Thesis (http://coursecatalog.web.cmu.edu/schools-colleges/schoolofcomputerscience/#scshonorsresearchthesistext) section for learning objectives, application requirements and expected outcomes.

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 SCS Policies section. 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 all general education requirements (humanities/arts and science/engineering). Dual degree students do not need to complete 07-128, and these students will replace 15-151 with either 21-127 or 21-128. Since the CS major requires at least a minor or concentration 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 all requirements for the CS primary major (except 07-128 which is not required and 15-151 which will be replaced with 21-127 or 21-128). In addition, at most 5 of the 12 computer science requirements can double count with all other declared 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.

Computer Science Additional Major

Students interested in pursuing an additional major in Computer Science should first consult with the Program Coordinator 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 12 computer science courses (not including 15-110 and 15-112 if needed), 5 mathematics courses, 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.

Declaration for the additional major is allowed only after all math requirements are completed or in progress, and at least 9 of the 12 CS requirements (core and electives) are completed or in progress. Due to high demand, 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:

Computer Science requirements (12 courses):

Core courses (all are required):		Units
15-122	Principles of Imperative Computation	12
15-150	Principles of Functional Programming	12
15-210	Parallel and Sequential Data Structures and Algorithms	12
15-213	Introduction to Computer Systems	12
15-251	Great Ideas in Theoretical Computer Science	12
15-451	Algorithm Design and Analysis	12

One Artificial Intelligence elective (minimum 9 units). Students will be able to tackle complex, real-world problems using techniques from Artificial Intelligence, including symbolic and probabilistic reasoning, machine learning, optimization, and perception.		Units
10-315	Introduction to Machine Learning (SCS Majors) (or 10-301)	12
11-411	Natural Language Processing	12
11-485	Introduction to Deep Learning	9
15-281	Artificial Intelligence: Representation and Problem Solving	12
15-386	Neural Computation	9
16-384	Robot Kinematics and Dynamics	12
16-385	Computer Vision	12
others as designated by the CS Undergraduate Program		

One Domains elective (minimum 9 units). Students will gain expertise in fundamental principles from a larger domain of computer science not already represented by other constrained categories, currently logic and languages, systems, and artificial intelligence (which includes machine learning, language technologies, and robotics). Students will be able to apply theoretical and computational techniques from the Computer Science core to an introductory study of another major subarea of Computer Science.		Units
02-251	Great Ideas in Computational Biology	12
05-391	Designing Human Centered Software	12
11-324	Human Language for Artificial Intelligence	12
15-322	Introduction to Computer Music	9
15-330	Introduction to Computer Security	12
15-362	Computer Graphics	12
15-455	Undergraduate Complexity Theory	9
17-313	Foundations of Software Engineering	12
others as designated by the CS Undergraduate Program		

One Logic & Languages elective (minimum 9 units). Students will master techniques for rigorous, formal reasoning about programs or systems, rooted in their logical foundations.		Units
15-311	Logic and Mechanized Reasoning	9
15-312	Foundations of Programming Languages	12
15-316	Software Foundations of Security and Privacy	9
15-317	Constructive Logic	9
15-414	Bug Catching: Automated Program Verification	9
17-355	Program Analysis	12
17-363	Programming Language Pragmatics	12
80-413	Category Theory	9
others as designated by the CS Undergraduate Program		

One Systems elective (minimum 12 units). Students will 1. be able to describe how the properties of modern hardware (e.g., processor architecture, networks, storage) influence the design and implementation of software systems, such as through reasoning about concurrency and performance. 2. be able to analyze failures and / or resource limitations of physical systems and plan for their mitigation or management. 3. be able to develop abstractions based on lower-level primitives to manage the failures or other difficulties inherent in working with hardware. 4. demonstrate their learning through significant project / system implementations, requiring both course-specific knowledge as well as general system-building skills (i.e., not just programming, but also design, debugging, testing, etc.). The programming tasks together constitute a significant fraction of the course grade (e.g., 40% or more).		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	Networking and the Internet	12
15-445	Database Systems	12
others as designated by the CS Undergraduate Program		

Two School of Computer Science electives (minimum 18 units):		
These electives can be from any SCS department (Computer Science [15-xxx], Computational Biology [02-xxx], Human-Computer Interaction [05-xxx], Machine Learning [10-xxx], Language Technologies [11-xxx], Robotics [16-xxx], or Software & Societal Systems [17-xxx]). They must be 200-level or above and at least 9 units each, with the following exceptions:		18
ADDITION: 15-195 and 15-295 can be combined to count as one SCS elective.		
DELETIONS: 1) the following courses do NOT count as SCS electives: 02-201, 02-223, 02-250, 02-261, 05-200, 11-423, 15-351, 16-211, 16-223, 16-224, 16-397, 16-480, 17-200, 17-333, 17-562; 2) some IDeATe courses and some SCS undergraduate and graduate courses might not be allowed based on course content. Always consult with a CS undergraduate advisor before registration to determine eligibility for this requirement.		

Math requirements (minimum 5 courses):

		Units
All of the following courses:		
21-120	Differential and Integral Calculus	10
21-122	Integration and Approximation	10
21-127	Concepts of Mathematics	12
or 21-128	Mathematical Concepts and Proofs	
21-241	Matrices and Linear Transformations	11
or 21-242	Matrix Theory	
21-259	Calculus in Three Dimensions	10
or 21-266	Vector Calculus for Computer Scientists	
or 21-268	Multidimensional Calculus	
or 21-269	Vector Analysis	
Plus one of the following:		
15-259	Probability and Computing	12
21-325	Probability	9
36-218	Probability Theory for Computer Scientists	9
36-226	Introduction to Statistical Inference (for students already taking 36-219 or 36-225)	9

Technical Communication requirement (1 course)

Learning Objectives:

Audience
Students will be able to analyze the audience or audiences of a work: who the stakeholders are, what those stakeholders are trying to accomplish, and how persuasive communication can help students achieve their goals.

Written Mechanics
Students will practice developing their professional writing voice, paying attention to spelling, grammar, style, and the use of visual information.

Adaptation
Students will be able to adapt expert-level information for a general audience, across textual, graphical, and oral presentation.

Genres/Templates
Students will practice developing information in a variety of communication contexts, connecting areas of content in their own voice, and leading the reader through a cogent sequence of ideas while working within the constraints of a particular genre or template.

Peer Review
Students will practice reading and revising the work of themselves and their peers.

Oral Presentation Mechanics
Students will prepare for the real-time act of presenting to an audience, including delivery, practice, venue setup, and managing presentation support (slides, props, etc.).

Oral Presentation Development
Students will explore formal and informal modes of communication, selecting, organizing and explaining information for a real-time audience, and increasing clarity with use of structure and style.

Group Work
Students will practice skills to help them work as a team, stay on track, and produce a group deliverable.

One Technical Communications course:		Units
07-300	Research and Innovation in Computer Science	9
17-200	Ethics and Policy Issues in Computing	9
76-270	Writing for the Professions	9

Double-Counting Restriction

Students pursuing an Additional Major in Computer Science must complete all requirements listed above. In addition, at most 5 of the 12 computer science requirements can be double counted toward all other declared majors and minors. The mathematics and technical communication requirements can be double counted without restriction. Students, especially from interdisciplinary majors or with multiple majors or minors, are urged to consult with the Computer Science Program Director or the Undergraduate Program Coordinator in the CS Undergraduate Office to determine double-counting restrictions specific to their own situations.

Computer Science Minor

Students interested in pursuing a minor in Computer Science should first consult with the Program Coordinator in the CS Undergraduate Office after completion of the prerequisites, 15-122, 15-150 and with at least one of the 200-level required courses in progress. Students are expected to complete all courses for the minor with a C or higher (for a minor average QPA of 2.0 or higher).

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

Prerequisites:		Units
15-112	Fundamentals of Programming and Computer Science (some students may need to take 15-110 prior to 15-112 for additional preparation)	12
21-127	Concepts of Mathematics	12
or 21-128	Mathematical Concepts and Proofs	
Computer Science core courses:		
15-122	Principles of Imperative Computation	12
15-150	Principles of Functional Programming	12
15-210	Parallel and Sequential Data Structures and Algorithms	12
One of the following Computer Science core courses:		
15-213	Introduction to Computer Systems	12
15-251	Great Ideas in Theoretical Computer Science	12
Two additional Computer Science electives, of at least 9 units each:		
CS elective courses must be 15-213 or higher, at least 9-units each. 15-351 cannot be used. One course can be from any other SCS department besides the Computer Science Department, with prior approval.		18
Note: students who have to take 15-213/18-213 or 15-251 as part of another degree program are required to replace that CS core course requirement with another CS elective (15-xxx) as defined above, for a total of 3 additional CS electives.		

Double-Counting Restriction

Students may double-count a maximum of 2 courses for the CS minor (not including the prerequisites) toward all other majors and minors. Students, especially from computing-related majors, interdisciplinary majors or with multiple majors or minors, are urged to consult with the Computer Science Program Director or the Undergraduate Program Coordinator in the CS Undergraduate Office to review double-counting restrictions specific to their own situations.