Connection-Oriented Computer Science Education

Kim Kihlstrom
Westmont College

Computers play an important role in every area of our society and are integral in every academic discipline. Today's computer science students need a background that will prepare them for the expanding range of computing opportunities. The opportunities for computer professionals are varied and increasing in diversity. However, undergraduate computer science programs tend to be narrowly focused on programming and related technical skills. Female students in particular tend to be highly interested in exploring connections between computer science and other fields.

How can we leverage these observations at a liberal arts college, where interdisciplinary connections are highly desirable, and where the student population is often heavily female? We have implemented a radically different form of computer science major that fosters interdisciplinary connections. We present our curriculum here as an example of a truly liberal arts approach to computer science, one that we believe will encourage participation by a wide range of students, and particularly women.

1 Motivation

Computers have become ubiquitous and permeate every facet of our society. They are increasingly a part of every academic discipline, and more and more fields of study have a basis in computer science. For example, bioinformatics is an important and relatively new area of study and research. Cognitive science, computer graphics, management information systems, and human-computer interaction are just a few more of the growing number of fields that are integrally related to computer science.

It is important for a computer science program to reflect the changing nature of technology and the way in which computer science informs and interacts with other fields. "The increased diversity of computing degree programs presents important challenges and opportunities... The new diversity of computing programs gives institutions an unprecedented opportunity to focus their degree programs to meet the needs of their students, communities, and other constituents in the most effective way." [Joint Task Force, 2005].

In order to meet the changing needs of students, schools must continue to evaluate and evolve their computer science programs. "One of the great potentials of the new diversity of computing degree programs is that it permits academia to bring its computing degree programs in line with the diverse needs that exist among students and in local communities. In the past, many institutions had little choice but to have a CS program on the technical side and perhaps an IS program on the business side. Now, a college might offer a portfolio of degree programs to serve various student needs more appropriately." [Joint Task Force, 2005].

"At most colleges, the CS program is (or should be) facing serious choices about two closely related issues: (1) the appropriateness of a narrow CS identity... and (2) how to best respond to the emergence of newer computing disciplines..." [Joint Task Force, 2005]. It is desirable to encourage students to explore connections between computer science and other fields, and to be aware of newer disciplines that are related to computing. "Schools have a message to communicate about the future of work: All jobs, including those in the arts, medicine, law, design, literature, and the helping professions, will involve more and more computing. Conversely, technological careers will increasingly draw on the humanities, social science, and 'people skills.' " [AAUW, 2000].

We have experienced declining enrollments in computer science programs nationwide. Diversifying our programs can lead to more success in recruiting and retaining students. "At many institutions, student retention is an important concern. Colleges and universities routinely report that 50% or more of those students who initially choose CS study soon decide to abandon it. It seems plausible to expect that the better the match between student interests and abilities and available degree programs, the better the retention level that can be achieved." [Joint Task Force, 2005].

Broadening our computer science degree offerings may also serve to strengthen our programs. "When an academic unit offers a family of quality computing degree programs, its faculty will naturally come to reflect a variety of perspectives on computing issues and challenges. This kind of situation can set the stage for a useful cross-fertilization of ideas among the disciplines which might, in turn, support creativity and innovation in both teaching and research." [Joint Task Force, 2005].

Diversifying our computer science programs can lead to increased diversity in our student population. "To the extent that a given institution's student population presents a range of interests and sets of abilities, a broader range of computing degree choices may permit the institution to do a better job of serving that range of students needs." [Joint Task Force, 2005].

In particular, female students are more likely to be attracted to interdisciplinary connections. "For most women students, the technical aspect of computing are interesting, but the study of computer science is made meaningful by its connections to other fields." [Margolis, 2002]. "Connecting computing to other fields and working within its human and social contexts make the study of computer science compelling and meaningful for [women]." [Margolis, 2002]. "Traditional computer science curriculum and programming assignments often lack the larger interdisciplinary framework that women find important." [Margolis, 2002].

"Apart from retention, computing educators (and others) have long been concerned with the relatively narrow profile of students who are attracted to the computing disciplines. A broader portfolio may prove to attract a wider population." [Joint Task Force, 2005]. It is important to "cast the net" as widely as possible to attract a diverse student population.

2 Computer Science and the Liberal Arts

A liberal arts college offers the perfect setting in which to offer an interdisciplinary degree in computer science. Some of the key questions a liberal arts education seeks to explore are, "what does it mean to be human?" and "what sets us free, or *liberates*us, to reach our full human potential?" A liberal arts program seeks to foster breadth of understanding, intellectual skills, and wholeness of knowledge, in part by pursuing interdisciplinary connections.

Prior work has been done in an effort to combine the strengths of a liberal arts environment with those of computer science education [Gibbs, 1986, LACS, 2004, Walker, 1996]. "Liberal arts programs in computer science generally emphasize multiple perspectives of problem solving (from computer science and other disciplines), theoretical results and their applications, breadth of study, and skills in communication... liberal arts programs embrace the premise that graduates working in areas related to their majors will find that their careers develop in unexpected ways -often involving new areas of knowledge and application." [LACS, 2004].

The broad skills that are emphasized in a liberal arts program serve computer science students well. "Three general-purpose capabilities that are among those fundamental to a liberal arts education are the ability to organize and synthesize ideas, the ability to reason in a logical manner and solve problems, and the ability to communicate ideas to others... Since liberal arts curricula traditionally emphasize principles, foundations, concepts, and theory, computer science programs in a liberal art setting usually highlight algorithms, discrete structures, programming languages, complexity, and the theory of computation." [LACS, 2004].

On a practical note, liberal arts colleges tend to have higher percentage of women than universities [Gose, 1997]; thus, a liberal arts program has a greater potential for attracting women to computer science.

3 Curricular Background

Given the desire to form interdisciplinary connections at our institution, combined with the goal of attracting a broader range of students to our program, we began to examine our curriculum with these factors in mind.

The previous instantiation of our computer science major was one of the largest majors at our institution in number of credit hours. The high number of units required for the major is not ideal in a liberal arts setting, where the general education requirements typically consume about one-third to one-half of the available credit hours. We desired to provide our computer science students with more options than were possible under the old major, to explore course work in other fields of interest, and to engage in interdisciplinary study.

In fact, many of our recent computer science students had desired to complete a double major. While some students were able to do so, others were forced to pursue a major in one field and a minor in the other because of scheduling difficulties and the high number of units required for the computer science major. A number of our students have pursued double majors in CS and art, CS and business, CS and engineering-physics, and CS and mathematics.

To address these needs and desires, we restructured our computer science major in several ways. First, we determined to offer a Bachelor of Arts degree in addition to a Bachelor of Science degree, since the smaller number of units required by the B.A. degree makes a double major more feasible. We also reduced the total number of units required for the B.S. degree.

Second, to encourage and facilitate inter-disciplinary study, we began offering computer science majors the option of completing an emphasis in a second field of interest. For example, a student can complete a B.A. in computer science with an emphasis in art, or a B.S. in computer science with an emphasis in bioinformatics. We provided a number of example programs that serve to aid students in planning coherent programs, although the structure allows for students to plan, in consultation with faculty advisors, an individualized program that adheres to general requirements.

Third, we wanted to streamline the various options to allow for ease in advising and checking of degree requirements. Rather than creating separate rigid tracks, one for each possible field that could be combined with computer science, we envisioned a core set of courses that would be required for all computer science majors, and then a set of four options to complete the degree, which are structured in terms of *general*requirements. The four options lead to B.A. and B.S. degrees in general computer science, as well as B.A. and B.S. degrees in computer science with an emphasis in a second field.

We have found that this restructuring greatly benefits our students, and that the way in which we have organized the new requirements has been helpful to faculty and administration. No new courses were required to implement this curriculum, either in computer science or in other fields, as we worked within the current courses offered by the various departments. We also worked closely with faculty in a number of the proposed fields of emphasis.

We believe that our approach reflects the observation made in Computing Curriculum 2005, "When we look at high-quality programs, we see coherent programs that are driven and developed from within. Faculty and local administrators contribute because they have looked beyond the boundaries of conventional subject-matter areas, recognized that their students and their community need something new and different, and innovated to solve what they see as a legitimate, substantive problem." [Joint Task Force, 2005].

Other institutions, *e.g.*, Queensland [QUT, 2006] and Georgia Tech [Georgia Tech, 2005], have also begun offering cross-disciplinary degrees, and we expect that this trend will continue.

4 Curriculum

Our computer science major [Westmont, 2007a] is designed to provide rigor while allowing for flexibility in preparing students for the diverse opportunities that are available.

All students who desire to complete a major in computer science must complete 32 semester units of core courses, designed to provide breadth of understanding in the field. The core courses emphasize the common framework and key concepts that are essential to computer science, and provide the fundamental underpinnings that allow for further study in computer science and interdisciplinary programs. They include both lower-division (numbered below 100) and upper division courses, traditional computer science courses, a senior capstone course, a course in ethics, and a course in discrete mathematics.

Core Courses Required for All CS Programs

- CS 10 Introduction to Computer Science I (4)
- CS 30 Introduction to Computer Science II (4)
- CS 120 Data Structures and Algorithms (4)
- CS 130 Software Development (4)
- CS 192 Project (2)
- CS 195 Senior Seminar (4)
- CS 198 Research (2)
- Math 15 Discrete Mathematics (4)
- Philosophy 104 Ethics (4)

Although ethical and social issues are woven throughout the core computer science courses, we also require the students to spend a focused time looking at these issues by requiring one course in ethics, taught by the philosophy department.

In addition to the core courses, one of the following four programs must be completed, which provide depth in computer science and, optionally, another field of interest. Each option offers a balance between lower-division and upper-division courses. The four completion options lead to B.A or B.S. degrees in general computer science, or B.A. or B.S. degrees combining computer science with another field. Each option provides flexibility within a general framework, allowing for a diversity of interests and goals within a computer science major. Each student is required to work closely with a computer science faculty advisor in planning the degree program. In the case of an emphasis in a second field, the student is required to work closely with a second faculty advisor in the other field to ensure that a coherent program is developed.

B.A. in General Computer Science

- CS 45 Computer Organization and Architecture
- CS 105 Programming Languages

- Additional courses in CS or math to bring the total to 44 units.

B.S. in General Computer Science

- CS 45 Computer Organization and Architecture
- CS 105 Programming Languages
- Math 9 Calculus I
- Additional courses in CS or math to bring the total to 56 units. Recommended courses for graduate school preparation:
  - CS 135 Formal Languages
  - CS 150 Topics: Information Theory, Coding Theory, and Cryptography
  - Math 10 Calculus II

B.A. in Computer Science With Emphasis in a Second Field

- Three additional courses in CS or math, one of which must be a CS upper-division course
- Three courses from a second field of interest, two of which must be upper-division courses

B.S. in Computer Science With Emphasis in a Second Scientific Field

- Three additional courses in CS or math, one of which must be a CS upper-division course
- Three courses from a second field of interest within the sciences, two of which must be upper-division courses

A sample four-year schedule leading to a B.S. in computer science with emphasis in a second scientific field is shown in Table 1. The remaining two to three time slots per semester can be used for general education requirements and electives.

5 Example Programs

A number of example B.A. and B.S. programs with an emphasis in a second field are shown below [Westmont, 2007b]. This list is not meant to be exhaustive or prescriptive. Other programs in these and other fields are possible, and students are encouraged to work with the faculty advisors from both fields in planning a coherent program. All course requirements listed are in addition to core courses.

Fall		Spring
First Year	Introduction to Computer Science I	Introduction to Computer Science II
	Discrete Math	Lower Division Science
Second Year	Computer Science or Math	Computer Science or Math
	Ethics	
Third Year	Software Development	Project
	Upper Division Science	Upper Division Science
Fourth Year	Research	Senior Seminar
	Data Structures and Algorithms	Upper Division Computer Science

Table 1: Sample Four-Year Schedule for B.S. in Computer Science With Emphasis in a Second Scientific Field

B.A. in Computer Science With Emphasis in Business and Management Information Systems

- CS 125 Database Design
- CS 140 Networks
- Math 9 Calculus I
- Math 5 Statistics
- Economics and Business 3 Principles of Accounting
- Economics and Business 17 Quantitative Economics
- Economics and Business 138 Managerial Economics
- Economics and Business 192 Change and Innovation

B.A. in Computer Science With Emphasis in Technology, Arts, and Media

- CS 5 Fundamentals of Computing
- CS 140 Networks
- CS 150 Topics: Graphics Programming
- Art 65 Computer Graphics I
- Communications 6 Messages, Meaning, and Culture
- Two of the following:
  - Communications 125 Mass Communication
  - English 101 Film Studies
  - English 111 Screenwriting
  - Art 165 Computer Graphics II
  - Art 167 Publication Design

B.A. in Computer Science With Emphasis in Art and Graphics

- CS 5 Fundamentals of Computing
- CS 140 Networks
- CS 150 Topics: Graphics Programming
- Math 20 Linear Algebra
- Art 65 Computer Graphics I
- Art 165 Computer Graphics II
- Art 167 Publication Design

B.S. in Computer Science With Emphasis in Computer Engineering

- CS 45 Computer Organization and Architecture
- CS 140 Networks
- CS 145 Operating Systems
- CS 150 Topics: Distributed Systems
- Math 9, 10 Calculus I, II
- Physics 23 General Physics
- Physics 142, 143 Circuits and Electronics with Lab

B.S. in Computer Science With Emphasis in Computational Mathematics

- CS 135 Formal Languages
- CS 150 Topics: Information Theory, Coding Theory, and Cryptography
- Math 9, 10 Calculus I, II
- Math 20 Linear Algebra
- Math 121 Numerical Analysis
- Math 123 Number Theory *or* Math 110 Modern Algebra
- Math 130 Probability and Statistics

B.A. in Computer Science With Emphasis in Human-Computer Interaction

- CS 150 Topics: Graphics Programming (4)
- Math 5 Statistics *or* Math 130 Probability and Statistics
- Math 20 Linear Algebra
- Art 10 Design I *or* Art 65 Computer Graphics I
- Psychology 13 Experimental Psychology
- Psychology 120 Cognitive Psychology
- Psychology 124, 124L Sensation and Perception with Lab

B.A. in Computer Science With Emphasis in Cognitive Science and Neuroscience

- CS 116 Artificial Intelligence
- CS 150 Topics: Machine Learning
- Math 5 Statistics *or* Math 130 Probability and Statistics
- Psychology 13 Experimental Psychology
- Psychology 120 Cognitive Psychology *or* Psychology 124, 124L Sensation and Perception with Lab
- Psychology 125 Physiological Psychology *or* Biology 162 Neuroscience

B.A. in Computer Science With Emphasis in Bioinformatics

- CS 125 Database Design
- CS 150 Topics: Information Theory, Coding Theory, and Cryptography
- Math 5 Statistics *or* Math 130 Probability and Statistics
- Biology 5, 6 General Biology I, II
- Biology 114 Genetics
- Biology 162 Neuroscience

B.A. in Computer Science With Emphasis in Artificial Intelligence and Philosophy

- CS 116 Artificial Intelligence
- CS 135 Formal Languages
- CS 150 Topics: Machine Learning
- Philosophy 12 Critical Reasoning and Logic
- Two of the following:
  - Philosophy 104 Ethics
  - Philosophy 135 Philosophy of Language
  - Philosophy 170 Theory of Knowledge
  - Philosophy 175 Metaphysics

B.A. in Computer Science With Emphasis in Analytical and Computational Chemistry

- CS 45 Computer Organization and Architecture
- Math 5 Statistics *or* Math 130 Probability and Statistics
- Physics 142, 143 Circuits and Electronics with Lab
- Chemistry 5, 6 General Chemistry I, II
- Chemistry 121, 122 Analytical Chemistry I, II

B.A. in Computer Science With Emphasis in Music

- CS 5 Fundamentals of Computing
- CS 105 Programming Languages
- CS 116 Artificial Intelligence
- Music 10, 12, 110, 112 Principles of Music I, II, III, IV
- Music 11, 13, 111, 113 Musicianship Lab I, II, III, IV
- Applied Music 30/130 Private Composition or other applied music course
- Physics 7 Physics of Music

B.A. in Computer Science With Emphasis in Secondary Education

- CS 5 Fundamentals of Computing
- CS 140 Networks
- Math 9, 10 Calculus I, II
- Art 65 Computer Graphics I
- Art 165 Computer Graphics II *or* Art 167 Publication Design
- Education 101 Explorations in Teaching: Culturally Diverse Secondary Schools
- Education 161 Computers for the Classroom Teacher: Secondary

6 Results

We now present quantitative and qualitative results from our 2007 graduating class, which is shown in Figure 1. We highlight four of the eight graduates.

Kristin (Figure 2) completed a major in computer science with an emphasis in kinesiology. She is now in physical therapy school at the University of Nebraska, pursuing research that will make use of her computer science skills.

Annie (Figure 3) started her undergraduate program as an elementary education major. She switched to a computer science major beginning in her sophomore year. She is now teaching introduction to computers, multimedia, web design, and java programming at a charter high school in Colorado.

JB (Figure 4) intends to pursue intellectual property law. His major was computer science with an emphasis in philosophy, and in particular the pre-law track within the philosophy department.

Mike (Figure 5) completed a double major in computer science and Spanish. He is currently doing software development in the Santa Barbara area, where there is a high percentage of Spanish-speaking residents.

Figure 1: 2007 Graduates.

Figure 2: Kristen
Computer Science and Kinesiology

Figure 3: Annie
Computer Science and Education

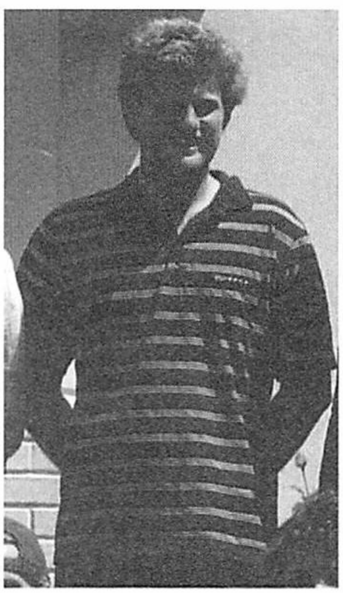

Figure 4: JB
Computer Science and Pre-law

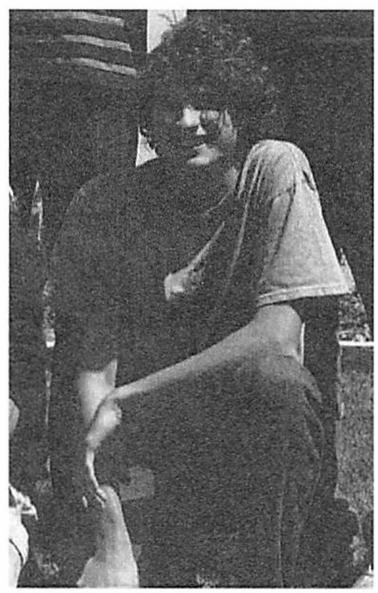

Figure 5: Mike
Computer Science and Spanish

It is important to note that none of these four students would have been likely to complete a computer science major had it not been for our new curriculum. Their other interests were strong enough that they would have eliminated the computer science component had it come to a choice between the two majors. However, because they were able to pursue the combination program, they did so. All of them were happy in the end that they had continued their studies in computer science, and all of them are still making use of their computer science major. As this group constituted half of our graduating class, and two-thirds of our female graduates, it had a significant impact on the size and diversity of the class. We expect to see similar results in the future.

7 Conclusions

We have presented a new liberal arts computer science curriculum that allows for interdisciplinary connections between computer science and other fields. Liberal arts schools provide the ideal setting in which to explore interdisciplinary topics. In an era when computers are ubiquitous and computer scientists are needed in a broad range of settings and fields, such interdisciplinary connections are extremely relevant. The additional options provided through our major serve to attract a more diverse pool of students. Women especially tend to be interested in these connections and are well-served by programs such as the one we present.

References

[AAUW, 2000] AAUW Educational Foundation Commission on Technology, Gender, and Teacher Education (2000). *Tech-Savvy:EducatingGirlsintheNewComputerAge.*

[Georgia Tech, 2005] Georgia Institute of Technology (2005). *B.S.inComputationalMedia.* http://lcc.gatech.edu/compumedia/Main.html.

[Gibbs, 1986] Gibbs, N. E. and Tucker, A. B. (1986). A model curriculum for a liberal arts degree in computer science. *CommunicationsoftheACM*, 29(3):202–210.

[Gose, 1997] Gose, B. (1997). Liberal-arts colleges ask: Where have the men gone? *ChronicleofHigherEducation*, A35:3–5.

[Joint Task Force, 2005] Joint Task Force for Computing Curricula 2005 (2005). *ComputingCurricula2005.* http://www.acm.org/education/curric _ vols/CC2005-March06Final.pdf.

[LACS, 2004] Liberal Arts Computer Science Consortium (2004). *A2004ModelCurriculumforaLiberalArtsDegreeinComputerScience,Draft.* http://www.lacs.edu/model-curriculum.pdf.

[Margolis, 2002] Margolis, J. and Fisher, A. (2002). *UnlockingtheClubhouse:WomeninComputing.* MIT Press, Cambridge, MA.

[QUT, 2006] Queensland University of Technology (2006). *BachelorofInformationTechnology–IT22:ITComplementaryStudies.* http://www.fit.qut.edu.au/future/undergraduate/complementary.jsp.

[Walker, 1996] Walker, H. M. and Schneider, G. M. (1996). A revised model curriculum for a liberal arts degree in computer science. *CommunicationsoftheACM*, 39(12):85–95.

[Westmont, 2007a] Westmont College (2007a). *CollegeCatalog*. http://cgi2.westmont.edu/publications/catalog_07_08/?page_id=18.

[Westmont, 2007b] Westmont College (2007b). *ProgramInComputerScience*. http://www.westmont.edu/_academics/pages/departments/computer_science/csprogram/.

A Course Descriptions

We now provide descriptions for the computer science courses; space limitations preclude descriptions of the courses from other departments. Courses typically meet for three and a quarter hours per week over the course of a fifteen-week semester; factoring in holidays, this amounts to approximately 45 total hours per course.

Our CS 5 course is a "CS0" course, designed to get students interested in pursuing further computer science courses, and meeting a general education requirement. Our CS 10 course is a "CS1" course and is taught using a functional language (Scheme). Our course CS 30 is a "CS2" data structures course and is taught using C++. Students thus experience a multi-paradigm approach during their first year. We also allow for multiple entries into the curriculum, as students may or may not take CS 5.

A.1 Lower-Division Course Descriptions

CS 5 Fundamentals of Computing Introduction to foundations of computing, problem solving, algorithmic thinking, and abstraction. Overview of hardware and software. Social and ethical issues.

CS 10 Introduction to Computer Science I Basics of programming including language features, disciplined programming style, and documentation. Problem solving, algorithm design, and the software development process.

CS 30 Introduction to Computer Science II Prerequisite: CS 10. Introduction to object-oriented programming. Abstract data types including lists, stacks, queues, and trees. Sorting and searching algorithms. Big-O notation. Software testing and program verification.

CS 45 Computer Organization and Architecture Prerequisite: CS 30. Computer structure, digital logic, data representation, computer arithmetic, software vs. hardware tradeoffs, addressing techniques, instruction sets, cache, virtual memory, pipelining.

A.2 Upper-Division Course Descriptions

CS 105 Programming Languages Prerequisite: CS 30. Language processors, data, binding time, operations, sequence control, referencing environments, scope of a variable, storage management, operating environment, syntax, translation.

CS 116 Artificial Intelligence Prerequisite: CS 30. Computational and philosophical principles of intelligence, methods for knowledge representation, automated reasoning, and learning.

CS 120 Data Structures and Algorithms Prerequisite: CS 30. Advanced data structures (balanced trees, heaps, graphs), hashing. Analysis of algorithms.

CS 125 Database Design Prerequisite: CS 30. Database system architecture, relational and object-oriented databases, SQL, normal forms and database design, query processing and optimization, handling transactions, concurrency control, crash recovery, data warehousing, and data mining.

CS 130 Software Development Prerequisite: CS 30. Software life-cycle. Fundamental concepts of software design. Supporting modern language features. Verification and validation techniques. The course is organized around a major group software project.

CS 135 Formal Languages and Automata Prerequisite: CS 30. Regular languages, finite automata. Context-free languages, pushdown automata, Turing machines, halting problem, computability.

CS 140 Networks Prerequisite: CS 45. Packet switching, network topologies, delay, throughput, and protocol layers. Inter-networking including IP, TCP, and UDP. Client-server paradigm. Web technologies including HTTP, CGI, and Java.

CS 145 Operating Systems Prerequisite: CS 45. Sequential processes, concurrent processes, scheduling algorithms, segmentation, paging, virtual systems, store management, networking, parallel processing, security.

CS 150 Topics in Computer Science Prerequisite: CS 30. Special courses on selected topics in computer science. Current offerings include Graphics Programming; Distributed Systems; Machine Learning; History and Ethics in Computing; and Information Theory, Coding Theory, and Cryptography.

CS 190 Internship Prerequisite: CS 30. Field experience arranged in conjunction with the department and supervised by professional computer scientists.

CS 192 Project Prerequisite: CS 130. Participation in a multi-person computer science project.

CS 195 Senior Seminar Prerequisite: computer science majors and minors with senior standing and CS 130. In this capstone class, students reflect on computer science as a discipline, the connections with other disciplines, the impacts of technology upon society, and ethical considerations introduced by computers. In the process of completing a major project, students consider marketing, design, implementation, testing, and maintenance. In this class, seniors complete and present their online portfolio. As the culmination of their program, students explore the transition to graduate school or the commercial sector.

CS 198 Research Prerequisite: CS 30. Work with faculty on original research.