

**Henry M. Walker***Samuel R and Marie-Louise Rosenthal Professor
of Natural Science and Mathematics
Professor of Computer Science*Department of Computer Science
Noyce Science Center
Grinnell College
1116 Eighth Avenue
Grinnell, Iowa 50112 USA

641-269-4208

fax 641-269-4285, 641-269-4984

walker@cs.grinnell.edu

<http://www.cs.grinnell.edu/~walker/>

December 18, 2013

To: Dean Lopatto and the Curriculum Committee

From: Department of Computer Science

Re: Changes in the Computing Curriculum

The Department of Computer Science proposes substantial changes in its course offerings and Computer Science Major, motivated by five major factors:

- Emerging recommendations for undergraduate computing science programs from international professional computing organizations,
- Acknowledgment of strengths of the current curriculum, combined with an interest in providing solid exposure to a broader range of topics within the discipline of computing,
- New approaches in pedagogy regarding the incorporation of software projects within the undergraduate curriculum,
- A desire for more flexibility in the major, an approach supported by students and alums, and
- A desire to support college goals to respond to societal and community needs.

Altogether, this proposal makes substantial adjustments in both content and pedagogy, while working within constraints of staffing and Grinnell's traditions of options for students within a liberal arts environment.

Since this proposal includes many changes to courses and the Computer Science Major, the narrative is organized into the following main sections:

1. Proposal Overview
2. Background
3. Reduced core with increased elective offerings
4. Software development, problem solving, student projects
5. Revised requirements for a Computer Science Major
6. Revised statement regarding honors in Computer Science
7. Transitions for current Computer Science Majors
8. Conclusions and national/international recognition
9. References

This document describes the various elements of this proposal, including an identification of courses, implications for the Computer Science Major, and appropriate background information. Details of new or revised courses are available separately in attachments to the paper copy of this proposal and at on-campus computers at

<http://www.cs.grinnell.edu/~walker/restricted-documents/2013-cs-curriculum-proposal/>

1. Proposal Overview

This proposal seeks to make the following changes in courses:

Courses dropped

- CSC 302, *Programming Language Concepts*
- CSC 362, *Compilers*
- CSC 323, *Software Design*
- CSC 325, *Databases and Web Application Design*

Courses regularized

- CSC 262, *Computer Vision* (4 credits)
§ offered previously for 4 credits in Fall 2011
- CSC 281, *Life Beyond Grinnell — Learning from CS Alumni* (1 credit)
§ offered previously for 1 credit in Spring and Fall 2013
- CSC 282, *Thinking in C and Unix* (1 credit)
§ offered previously for 1 credit in Spring 2013

Courses significantly revised (each course goes from 4 credits to 2 credits)

- CSC 232, *Human-Computer Interaction* (2 credits)
§ offered previously for 4 credits
- CSC 216, *Computer Networks* (2 credits)
§ offered previously for 4 credits as CSC 364

New courses added (all offered for 2 credits)

- CSC 214, *Computer and Network Security*
- CSC 312, *Programming Language Implementation*
- CSC 321, *Software Development Principles and Practices*
- CSC 322, *Team Software Development for Community Organizations*

With these changes, requirements for the Computer Science Major would become:

Multi-paradigm, Introductory Sequence (12 credits required)

- § CSC 151, *Functional Problem Solving*
- § CSC 161, *Imperative Problem Solving and Data Structures*
- § CSC 207, *Algorithms and OO Problem Solving* (updated theme)

Required Upper-Level Theory Courses (8 credits required)

- § CSC 301, *Analysis of Algorithms*
- § CSC 341, *Automata, Formal Languages, & Computational Complexity*

Systems (4 credits required; 8 credits strongly recommended)

- § CSC 211, *Computer Organization and Architecture* (4 credits)
- § CSC 213, *Operating Systems and Parallel Algorithms* (4 credits)

Software Development (4 credits required)

- § CSC 321, *Software Development Principles and Practices* (2 credits)
- § CSC 322, *Team Software Development for Community Organizations* (2 credits)

Electives (4 credits in CS at the 200-level or higher)

- § CSC 281, CSC 282, guided reading, independent study and MAPs may not be used to fulfill this requirement.
- § If repeated, up to 2 credits of CSC 322 may contribute towards satisfying this requirement.

Mathematical Foundations

- § MAT/CSC 208, *Discrete Mathematics* or MAT 218, *Combinatorics*
- § Math elective: a course numbered above MAT 131

2. Background

Within the discipline of computer science, the major professional societies (the Association for Computing Machinery (ACM) and the Computer Society of the Institute of Electrical and Electronics Engineers (IEEE-CS)) combine to produce extensive recommendations for undergraduate curricula on a regular basis. Thus, the ACM and IEEE-CS produced an extensive (240-page) volume of recommendations in 2001 [3], and an interim document was published in 2008 [1]. Since these recommendations are intended to apply to all types of schools from technical institutes to liberal arts colleges, each individual school must interpret the national guidelines for its own audience and mission. As a result, the Liberal Arts Computer Science Consortium (a group of faculty from liberal arts colleges) also produces recommendations tailored toward liberal arts colleges. The most recent of these liberal arts recommendations was published in 2007 [4].

Grinnell's current computer science curriculum and major draw strongly upon these national recommendations, and Grinnell's current offerings cover the vast majority of the recommended content. Pragmatically, Grinnell's limit of 32 credits in a major does not allow the department to require both CSC 211 and CSC 213, but students could gain solid coverage of most topics required by national recommendations by taking the 32-credit major plus both CSC 211 and CSC 213.

In the past two years, the ACM and IEEE-CS again have combined to develop updated and revised curricular recommendations. To date, drafts have appeared in November 2012 and February 2013 [2], and a "Pre-release" of the final recommendations appeared in October 2013. As with past recommendations, the emerging recommendations provide extensive details on content, including a specification of about 300 hours of classroom coverage in 171 areas (called knowledge units) and approximately 1052 learning outcomes.

As would be expected in a rapidly-evolving technical field, the emerging curricular recommendations update content in some subdisciplines, de-emphasize content in other subdisciplines and significantly expand recommended coverage in still other subdisciplines. Beyond modest refinements, substantial changes were found in several general areas.

- New recommendations highlight the need for students to be comfortable with multiple views of problem solving.
- Fundamental topics related to problem solving and programming languages remain in the 2013 recommendations, but some advanced topics related to language concepts are de-emphasized.
- The areas of networking, computer security, and parallel computing are greatly expanded.
- Discussion of social and professional issues is expanded and more intentional.
- Some recommendations encourage familiarity with a relatively wide range of topics.

3. Reduced Core with Increased Elective Offerings

At a fundamental level, the discipline of computer science draws upon three basic methodologies or processes:

- theory, from mathematics, begins with definitions and axioms, utilizes formal proofs to obtain results.
- abstraction, from the experimental sciences, utilizes hypothesis formation, modeling, experimentation and analysis to explore and test hypotheses.
- design, from engineering, creates and utilizes formal problem-strategies to address complex problems.

A strong computer science major should combine these basic processes with an extensive collection of topics and learning outcomes that are recommended by national curricular guidelines (e.g., [2]). The current Computer Science Major has organized these themes by specifying five specific introductory or core courses (CSC 151, CSC 161, CSC 207, CSC 301, CSC 341) and requiring one course from each three content areas (systems, languages, large team-based project). Although these requirements did a good job of providing contact with the three basic processes and with vital topics, the current major does not allow much flexibility.

Also, the emerging curricular recommendations [2] identify several areas as having increased significance to computing today, including networks, computer security, social and ethical issues, and distributed systems. Although some students will be interested in all of these emerging areas, others will not.

In order to provide students with the option of learning important content in various areas, the proposed curriculum drops some 4-credit courses (e.g., CSC 302 and CSC 362), condenses other 4-credit courses to 2-credits (e.g., CSC 232 and CSC 364—reabeled 216), and offers some new 2-credit courses (e.g., CSC 214 and CSC 312).

With the dropping of CSC 302 and CSC 362, student will not normally delve as deeply into topics related to programming languages—a particular strength of the current curriculum. However, the introductory sequence (CSC 151, CSC 161, and CSC 207) provides important introductions to some language concepts, and the new 2-credit CSC 312 will build upon this base.

In the current curriculum, CSC 364, *Networks*, has been offered for 4 credits, but did not satisfy any requirements for the major. Enrollments have been unimpressive. In the revised curriculum, this course will be restructured for 2 credits, coupled with a new topic, computer security, as two 2-credit courses, moved to the 200-level (214), and allowed to count toward the major within the 4-credit elective requirement. This change adds a topic (security) to the curriculum and brings both networks and security into a more central role within the curriculum.

Similarly, some content within CSC 232, *Human-Computer Interaction*, will move to new project-based courses, but otherwise CSC 232 will become a 2-credit course that also would fulfill part of the elective requirement for the major.

Altogether, the new curriculum brings several new topics into the curriculum and provides students a wider range of options as they plan their undergraduate program.

Details of each of these revised or new courses are given in the appendices and at on-campus computers at

4. Software Development, Problem Solving, Student Projects

National curricular recommendations (e.g., [1, 2, 3, 4]) consistently emphasize the importance of team-based projects within the undergraduate program. Such projects reinforce fundamental concepts and also provide important insights regarding problem solving in the context of large-scale software development. Special challenges arise when developers interact with clients and other team members. Simply stated, problem-solving approaches involving one developer working for herself/himself rarely scale up nicely to allow problem solving for large, complex problems.

Time constraints complicate the incorporation of large-scale projects within a liberal arts context. To be a successful educational experience, student experience with a large project should include several key elements:

- Formal course work should include problem-solving methodologies, development principles, and other foundational material.
- Students should interact with one or more clients.
- Students should work within a team (likely 3-5 students).
- The project should be large enough to require careful planning, organization, and management.
- Software developed should have sufficient scale (at least several thousand lines of code), so that students can experience the pragmatic issues of large-scale software development.

Over the years, Grinnell's computer science faculty have experimented in finding ways to incorporate all these key elements within a single 4-credit course. These courses have had mixed success, but challenges and difficulties regularly arise in both CSC 323 and CSC 325. After reviewing emerging curricular approaches and pedagogy at a few other schools, this seems an appropriate time to consider a different approach, involving two 2-credit courses.

- CSC 321, *Software Development Principles and Practices*, will encapsulate many foundational principles and methodologies for large-scale problem solving.
- CSC 322, *Team-based Development for Community Organizations*, will allow students to gain experience with a practical, multi-semester project to help a community-based client.

Students would take CSC 321 either before CSC 322 or concurrently with CSC 322, to gain appropriate background.

For CSC 322, a faculty member will enter into an understanding with a community-based organization to try to address an identified need in the community. Typically, over a 2-3 year period, the faculty member would guide teams of students over several semesters to address the problem and move toward helpful solutions. As with any development project or academic course experience, no guarantees could be made concerning the success or outcome of the effort, but the client, the faculty member, and the students involved would strive to address the need in a meaningful way.

Since CSC 322 would involve on-going projects over several semesters, students would be allowed to enroll in the course multiple times. At first, students might take a modest, participant role, but students might move over time toward leadership positions.

With this new approach to software development and projects within the curriculum, both CSC 323 and CSC 325 would no longer be offered. Instead, the department envisions offering CSC 321 one semester each year and CSC 322 every semester.

Additional details for both CSC 321 and CSC 322 may be found in the appendix and at on-campus computers at

<http://www.cs.grinnell.edu/~walker/restricted-documents/2013-cs-curriculum-proposal/>

5. Revised Requirements for a Computer Science Major

Emerging curricular recommendations from national professional computing organizations [2] specify topic coverage extending over 300 hours within the classroom. Organizing those topics into coherent courses requires more time, and placing this material within a context of a liberal arts education involves still more time. At Grinnell, many classes meet an hour per day for three days a week (sometimes four days a week), so covering the recommended topics within well-crafted courses would likely fill more than eight 4-credit Grinnell courses. Overall, a 32-credit Grinnell major cannot meet all of the national curricular recommendations for computing [2].

Of course, not all computer science majors will be interested in careers within the computing field. Also, some students will specialize within computing. Some students may be well served by taking a coherent collection of courses that provides a solid background in problem solving and some areas of computing, but does not fulfill the extensive range of national recommendations. Past requirements for a Computer Science Major have taken the diverse needs of students into account, but the emerging recommendations [2] highlight this need for a major that serves multiple populations.

With this reality, the computer science faculty propose a revised Computer Science Major that covers much foundational work, but also allows students to tailor some course work toward their specific interests and needs. As described in Section 1, the new Computer Science Major will require the introductory sequence (CSC 151, CSC 161, CSC 207), two foundational courses (CSC 301, CSC 341), a significant software-development experience (CSC 321, CSC 322), essential mathematical foundations (two courses), and at least one systems course (CSC 211 or CSC 213). However, in contrast to past Grinnell requirements, students also would be able to gain breadth by choosing four credits from a range of eligible electives. Overall, these requirements for all majors seem an excellent balance among core topics, problem-solving experiences, and electives.

As a separate matter, students interested in careers in computing will be advised that some additional topics are needed to meet national recommendations. In particular, a more complete major would add these courses to the basic, 32-credit major:

- Both CSC 211, *Computer Organization and Architecture*, and CSC 213, *Operating Systems and Parallel Algorithms* (4 credits beyond the 32-credit major)
- CSC 214, *Computer and Network Security* (2 credits)
- CSC 312, *Programming Language Implementation* (2 credits)
- CSC 216, *Computer Networks* (2 credits)

Since the 32-credit major allows 4 credits of electives, this expanded listing would yield a 38-credit major for students to meet about 96% of the learning outcomes identified by national recommendations.

As an analogy, Grinnell's Chemistry Major specifies 32 credits for any student. However, for a student's program to be approved by the American Chemical Society, some additional courses are expected. The above listing would provide similar guidance to computing students who want to move beyond the basic, 32-credit Computer Science Major.

6. Revised Statement regarding Honors in Computer Science

For many years, the Computer Science Department has provided a detailed list of specific requirements regarding expectations for honors in computer science. As the major becomes more flexible, however, any specific list becomes unwieldy. For this reason, the Computer Science Department expects to replace its current detailed list with the following general statement.

To be considered for honors in computer science, graduating seniors must not only meet the College's general requirements for honors but also demonstrate exceptional commitment to the discipline and its values, as evidenced by significant engagement in the department and excellence in computing-related work, both in the classroom and beyond.

Although statements regarding honors are not subject to approval by the Science Division or the Curriculum Committee, this statement is included here for completeness.

7. Transition for Current Computer Science Majors

The current Computer Science Major specifies course work in identified categories:

- A. Introductory sequence (12 credits)
- B. Core courses (8 credits)
- C. Systems content (4 credits from two 4-credit options)
- D. Languages content (4 credits from two 4-credit options)
- E. Large, team-based project (4 credits from two 4-credit options)
- F. Supporting mathematics (12 credits, including MAT 131, CSC/MAT 208 or MAT 218), and 1 elective above MAT 131)

The proposed Computer Science Major has several similar requirements, but includes some alternative courses and some flexibility:

- §§§ Requirements A, B, and C identical to the current major
- D. Electives (4 credits from several 2- and 4-credit options)
 - E. Large, team-based project (two 2-credit courses for 4 credits)
 - F. Supporting mathematics (8 credits, including CSC/MAT 208 or MAT 218 and 1 elective above MAT 131)

In the transition from the current computer science major to the new computer science major, the following observations and rules would apply.

- Both current and new majors will need to satisfy requirements A, B, and C without change.
- Courses counting toward the languages content requirement of the current major could be used to satisfy the electives requirement of the new major.
- Current majors without courses to satisfy the languages requirement would be allowed to graduate under the rules for the new major.
- The large, team-based project requirement (requirement F) would be met by either of the current course options (CSC 323 or CSC 325) or by both of the revised, 2-credit project courses (both CSC 321 and CSC 322). Both current and new majors could fulfill either the current or new requirements by four credits of either the current or new project-based courses.
- MAT 131 is a specified pre-requisite for CSC/MAT 208 or MAT 218 and for the required mathematics elective above MAT 131. For students starting in MAT 124 or MAT 131, both current and new requirements yield the same course requirements. However, for students with advanced placement, the current requirements may not be clear regarding whether another mathematics course is required. The new requirements clarify that taking courses after MAT 131 assure the requirement level of mathematics competency without taking MAT 131 itself.

In summary, the transition from current to new course offerings and requirements seems straightforward, since various current courses satisfy new requirements, and the new major is somewhat more flexible than the current major.

8. Conclusions and National/International Recognition

This proposal represents a 2-year effort to update the content and pedagogy of Grinnell's computer science curriculum and major.

- Changes in this proposal reflect ongoing interest by the computer science faculty for continual improvement in courses and the curriculum, based on exit interviews with graduates, conversations with alumni, evaluations from courses, feedback from students, and faculty observations. Many specific proposals were advanced in a three-day computer science curriculum workshop in July 2013, with extensive follow-up conversations and collaborations.
- The computer science faculty interacted with the ACM/IEEE-CS Task Force in its preparation of the emerging 2013 recommendations for undergraduate computer science [2].
 - § The current draft recommendations [2, pp. 283-291] explicitly recognize Grinnell's introductory sequence (CSC 151, CSC 161, and CSC 207) as "exemplar" courses for beginning students
 - § Grinnell's proposed upper-level offerings reflect the full range of topics identified as vital by the emerging national recommendations [2].
 - § Grinnell's proposed 32-credit computer science major covers many fundamental topics within the discipline, while allowing diverse students flexibility to meet their educational goals. With careful advising, students specifically interested in careers in computing will be able to meet topic coverage from the national recommendations [2].
- Grinnell's curriculum fits well within a liberal arts environment, balancing desired breadth among disciplines and appropriate depth within a major.
- Grinnell's new project-based courses will provide important experience in problem solving related to large, team projects, and the projects will support the College's interest in social engagement and community action.
- The final 2013 recommendations [2] of the ACM/IEEE-CS Task Force identify five examples of how schools can update their curricula to address new developments of content and pedagogy.
 - § With Grinnell's ongoing dialog with the Task Force, the "Pre-release" final Computing Curricula 2013 document recognizes Grinnell's full curriculum.
 - § Grinnell's experience shows how a school can utilize constructive assessment and review in conjunction with emerging curricular recommendations to develop a modern computing curriculum while remaining true to its mission as a liberal arts college.
 - § Each of the other four exemplar curricula come from schools represented by Task Force members: two programs from Union City College as a community college, one from Stanford University as a research university, and one from Williams College as another liberal arts college.
 - § Dialog between Grinnell's computer science faculty and the Task Force influenced Grinnell's curricular proposal, but also had a clear impact on the Task Force. As a result, the Task Force has indicated that each of Grinnell's computer science faculty will be recognized individually in the final report. Overall, Grinnell is mentioned on 23 pages of the "Pre-release document".

In summary, this proposal updates Grinnell's curriculum and major to reflect recent developments in the field, incorporates cutting-edge pedagogy, provides students with additional flexibility to meet their educational objectives, and highlights Grinnell as a national and international leader in undergraduate computing education.

9. References

1. ACM/IEEE-CS Interim Review Task Force, *Computer Science Curriculum 2008: An Interim Revision of CS 2001*. ACM and the IEEE Press, New York, NY, December 2008.
2. ACM/IEEE-CS Joint Task Force. *Computer Science Curricula 2013: Ironman Draft*. February 2013. Retrieved September 18, 2013 from <http://ai.stanford.edu/users/sahami/CS2013/>.
3. ACM/IEEE-CS Task Force on the Curriculum. *Computing Curricula 2001*. ACM and the IEEE Press, New York, NY, December 2001.
4. Liberal Arts Computer Science Consortium. “A model curriculum for a liberal arts degree in computer science”, *Journal on Educational Resources in Computing (JERIC)*, 7(2), June 2007, Article 2.
- .
- .