Template for a
Review of the Mathematics and Computer Science Department at Some University

By First Reviewer, First Reviewer's School
Henry M. Walker, Grinnell College
Third Reviewer, Third Reviewer's School

October 31, 2008

This report provides a template for an external review of a combined Department of Mathematics and Computer Science. This document contains some text in each section to illustrate the use of recommendation macros and to suggest possible discussion topics. Of course, sections may be inserted or deleted to address specific constituencies within a department.

A typical review report may start immediately with an introductory section or with some thank yous:

As we begin this report, the external reviewers want to express our great thanks and appreciation to all members of the University community for their hospitality, insights, commentary and analysis, and conversations. The reviewers experienced a fine sense of community, sharing, and openness throughout our visit, and we greatly appreciate the insights and perspectives of all we interviewed. Special thanks are due to ...

1 Introduction

As with any report, the Introduction should identify the main themes and set the tone for what follows. Often, the introduction starts with the Mission Statement of the Department of Mathematics and Computer Science at Some University:

> A Mission Statement goes here

This statement often provides a context for the elements of the review that follows. The introduction also commonly comments upon the audiences served by the department.

A typical introduction clarifies the structure of what follows.

Section 2 provides an overview that highlights many strengths of the department. Subsequent sections include additional observations, analysis, and suggestions for the future.

Section 3 makes observations and recommendations for the department as a whole.

Section 4 sometimes provides a review of how the department may fit within a broad environment of the entire university.

Section 5 may present observations and recommendations regarding mathematics; and Section 6 examines the computer science program.

Section 7 considers observations and suggestions regarding the relationship of the department with various administrative departments, such as the Admission Office, Information Technology Services, and the university's tutoring system.

A brief summary or coda concludes the main body of the report.

For reference, the report also contains several appendices, such as the following.

A. A List of References

B. Articles on mathematical problem solving, the role of computer science within the liberal arts, or other background documents. (A statement on *Computer Science and the Liberal Arts* appears here as an example.)

C. A complete list of the recommendations made throughout this report.

2 Positives Provide a Solid Foundation

Almost any Department of Mathematics and Computer Science has several strengths. Since much of a review document provides feedback on current programs, recommends changes, and identifies possible new directions, the report might seem rather negative.

$\Longrightarrow$ **Recommendation 2.1:** An early section should highlight strong elements within the department under review.

- The $\backslash newrecommendation$ macro sets off a recommendation within the report's text and saves the recommendation for inclusion in the last appendix.
- The $\backslash newrecommendation$ macro numbers recommendations consecutively within a section (assuming the $\backslash recounter$ variable is reset to 0 at the start of each section, as shown throughout this document.
- The $\backslash newrecsublist$ macro inserts a bullet item under a $\backslash newrecommendation$ in the text and also in the appendix of recommendations.

In the sections that follow, the report will make many suggestions. These suggestions should be considered in the context of a department which is basically strong.

3 Comments Regarding the Entire Department

Since a department functions as an overall academic/administrative unit, a review team normally identifies some observations that relate to the department as a whole. In many reviews, these general comments may be divided into a few subsections.

3A Goals

Departmental goals provide some high-level perspectives and often provide a fruitful framework for examining major elements within the overall department.

⟹ **Recommendation 3.1:** The functioning of a department may support the high-level goals of a department, but sometimes practices seem counter productive.

3B General Education, Facilities, Assessment Plans, Etc.

A new subsection might be identified for each high-level topic within a department.

4 The Department within the University Environment

A campus visit often involves meetings with majors, non-majors, and faculty from other departments.

⟹ **Recommendation 4.1:** The view of the department from the outside often has a direct impact on enrollments, advising, resources, facilities, and the level of cooperation present from other departments.

4A Student Perspectives, Community, and Audience

Majors and non-majors can give vital perspectives on how students view the department.

- What reputation does the department have within the university?
- Why do students take courses in the department?
- Why do students decide to major in the department?
- What about the department turns off which students?

$\Longrightarrow$ **Recommendation 4.2:** The department should review its programs, courses, and pedagogy to ensure that all activities connect with many students.

4B Public Relations

Even if a department has truly wonderful programs, few students will enroll or major if they do not know about these programs.

$\Longrightarrow$ **Recommendation 4.3:** The department and university should review the image of the department on campus.

- Review should include Web pages, the university catalog, admission materials, etc.
- The department should identify opportunities to promote its programs, such as posters, admission events, family weekends, etc.

4C Outreach to Other Departments

$\Longrightarrow$ **Recommendation 4.4:** The department should look for ways to increase communication with other departments.

- Mathematics departments should understand the needs and interests of client departments.
- Mathematics faculty can use the Curricular Foundations Project [3] of the MAA as an excuse to renew conversations.

5 The Mathematics Program

A discussion of the mathematics program often considers both introductory-level and upper-level courses. Specifics depend greatly upon the details of the department and university.

5A Introductory-level Mathematics

Distribution requirements may force students to take an initial mathematics course, but students likely will take further courses only if these first experiences go well.

⟹ **Recommendation 5.1:** Beginning courses provide a significant recruiting opportunity.

- Non-majors' courses can excite students and encourage further exploration.
- Courses for non-majors and for majors should highlight applications as well as theory, so students appreciate the relevance of mathematics as well as its beauty.
- A curricular review should consider both content and pedagogy.

5B Upper-level Mathematics

In addition to preparing students for work in client departments, upper-level courses should give majors a good background in a range of mathematical areas.

⟹ **Recommendation 5.2:** The CUPM Curriculum Guide [2] of the MAA can provide useful guidance in considering the breadth and depth of a majors' program.

6 The Computer Science Program

Computer science is misunderstood on many campuses. In such cases, this report can help educate administrators as well as guide the department itself. Thus, a first subsection often presents a general introduction to computer science within the liberal arts. Later subsections consider various components of the curriculum.

6A Computer Science and the Liberal Arts*

Historically, computer science has built upon many disciplines. Over time, the discipline also has developed its own principles, theoretical foundations, and problem-solving perspectives. Basic building blocks include formal deduction and theoretical understandings from mathematics, the development and refinement of hypotheses through

* This subsection is a slightly edited version of material written by Henry M. Walker for a review of Willamette University, 28 November 2007, and is used by permission.

the scientific method, and the refinement of formal problem-solving strategies that build on insights from engineering. [4, 6, 7]

Although computing encompasses numerous topics on many levels, the discipline of computer science generally is considered to have a clear intellectual and academic focus. In 1986, Gibbs and Tucker defined computer science as “the study of algorithms and data structures: their creation, analysis, and realization” [4]. Recently, the Liberal Arts Computer Science Consortium has noted that

> 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. In addition to the material content of computer science, the algorithmic approach is a very general and powerful method of organizing, synthesizing, and analyzing information. 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. The design, expression, and analysis of algorithms and data structures draw upon and contribute significantly to the development of all three capabilities. [5]

With the explosion of ideas and developments within the field of computing, liberal arts programs must take particular care to focus on principles, foundational concepts, and theory. At the same time, these ideas are informed by experimentation and practice, just as the scientific method uses laboratory work to test and refine hypotheses. In computer science and the other sciences, laboratory exercises help solidify ideas and push understandings to deeper and more mature levels.

Appendix B gives further perspective on the role of computer science in the liberal arts.

6B introductory Computer Science

When a university has distribution requirements, it is vital that computer science fills some niche.

A. At some schools, computer science satisfies a requirement in mathematical or logical thinking and problem solving.

B. At other schools, computer science satisfies a lab-science requirement.

As with mathematics, introductory computer science offers an opportunity for recruiting students into the discipline.

6C The Computer Science Curriculum

The ACM and IEEE-CS publish guidelines for undergraduate computer science [1], and the Liberal Arts Computer Science Consortium identifies recommendations for programs in the liberal arts [5].

6D Recruitment, Visibility, Public Relations

Since popular culture promotes many misconceptions about computing, computing programs must make a special effort to clarify the nature of computer science and to reach out to prospective students.

$\Longrightarrow$ **Recommendation 6.1:** Computing departments should mount a consistent and extensive public relations effort to recruit students and to counter negative societal stereotypes. One or more of the following themes connect with many students.

- Problem solving and intellectual challenge
- Ability of computing applications to help people
- Potential for fulfilling and exciting careers

7 Relationships with Administrative Departments

Since each university has special opportunities and constraints, reviewers must listen carefully to the circumstances at a school.

$\Longrightarrow$ **Recommendation 7.1:** External reviewers might schedule meetings with a range of administrative offices and departments. Meetings with some of the following may turn out not to be necessary, but a reviewer likely will not know that until after the review.

- the Office of Admission
- the Department of Information Technology Services
- an office of tutoring or academic advising

8 Coda

A final section highlights a few key points and again thanks all parties involved with the review.

A Bibliography

1. ACM/IEEE-CS Task Force on the Curriculum, *Computing Curricula 2001*, ACM and the IEEE Press, 2002.
2. Committee on the Undergraduate Program in Mathematics (CUPM) of the Mathematical Association of America, *CUPM Curriculum Guide 2004: Undergraduate Programs and Courses in the Mathematical Sciences*, Mathematical Association of America, 2004.
3. Ganter, Susan L, and William Barker for the Subcommittee on Curriculum Renewal Across the First Two Years (CRAFTY), *The Curriculum Foundations Project: Voices of the Partner Disciplines*, Mathematical Association of America, 2004.
4. Gibbs, N. and Tucker, A. A model curriculum for a liberal arts degree in computer science. *Communications of the ACM* 29, 3 (Mar. 1986), 202-210.
5. The Liberal Arts Computer Science Consortium, "A 2007 model curriculum for a liberal arts degree in computer science", *Journal on Educational Resources in Computing (JERIC)*, Volume 7, Issue 2, June 2007, article 2.
6. Henry M. Walker and G. Michael Schneider, "A Revised Model Curriculum for a Liberal Arts Degree in Computer Science", *Communications of the ACM*, December 1996, pp. 85-95.
7. Henry M. Walker, *Computer Science and the Liberal Arts*, communication to John Fink, Kalamazoo College. (Slightly edited, updated, and used by permission)

B Computer Science and the Liberal Arts[7]

Over the past 50 years, computer science has evolved from an esoteric research field based on the solution of highly specialized problems to a broad discipline emphasizing theory, fundamental principles, and techniques of problem solving. With this rapid development of a new discipline, the vast majority of liberal arts colleges have found that computer science fits naturally within the undergraduate curriculum. For example, in speaking for the Liberal Arts Computer Science Consortium in 1986, Gibbs and Tucker noted that many liberal arts colleges already were offering a B.A.-degree program in computer science. These institutions

> offer this program both because they view computer science as an essential discipline within their general academic mission, and because they see the major as preparation for a variety of career paths and graduate programs (inside and outside the computer science community). 6, p. 204]

While this statement depends upon an extensive analysis well beyond the size of this letter, some vital components include the breadth of the discipline, the multiple approaches to problem solving, and the development of core principles and theory. Each of these areas connects directly with central themes within the liberal arts. Further, widely accepted recommendations for an undergraduate computer science curriculum [4, 9] emphasize such themes over the vocationalism often found outside the liberal arts tradition.

Perhaps the most important starting point for any analysis of the discipline is to emphasize that *computer science is not the same as programming*. In this regard, the study of English or a foreign language provides a useful analogy. In language study, the first several courses include discussions of vocabulary and syntax, as well as material on culture, literature, and history. Mechanics of the language play a relatively minor role within a foreign language major, but students must become proficient in the language to understand relevant ideas and to communicate these ideas effectively. Within computer science, programming provides the grammar and syntax for similar communication. While programming is part of many beginning courses, well-designed courses move beyond the syntax to introduce broader ideas and concepts.

While any listing of topics within computer science will be incomplete or uneven, the primary computing societies recognize 14 areas as being vital for undergraduate computer science education [1]:

- Algorithms and Complexity
- Architecture and Organization
- Computational Science
- Discrete Structures
- Graphics and Visual Computing
- Human-Computer Interaction
- Information Management
- Intelligent Systems
- Net-Centric Computing
- Operating Systems
- Programming Fundamentals
- Programming Languages [e.g., precise expressions of problem-solving paradigms]
- Social and Professional Issues
- Software Engineering [e.g., approaches to complex problems]

While students may write programs during the study of many of these areas, such work provides a means to communicate ideas effectively and precisely.

Consistent with a liberal arts perspective, a computer science curriculum should approach each of these topics using three basic methodologies or processes: theory, abstraction, and design. The following excerpts come from *Computing Curricula 1991* [1]:

> One such process, called *theory*, is akin to that found in mathematics, and is used in the development of coherent mathematical theories. It has the following major elements:
>
> - Definitions and axioms
> - Theorems
> - Proofs
> - Interpretation of results
>
> This process is used in developing and understanding the underlying mathematical principles that apply to the discipline of computing. [p. 9-10]
>
> ...
>
> The second process, called *abstraction*, is rooted in the experimental sciences, and has the following elements:
>
> - Data collection and hypothesis formation
> - Modeling and prediction
> - Design of an experiment

> - Analysis of results
>
> When persons engage in abstraction, they are modeling potential algorithms, data structures, architectures, and so forth. They are testing hypotheses about these models, alternative design decisions, or the underlying theory itself. [p. 10]
>
> ...
>
> The third process, called *design*, is rooted in engineering and is used in the development of a system or device to solve a given problem. It has the following parts:
>
> - Requirements
> - Specifications
> - Design and Implementation
> - Testing and Analysis
>
> When computing professionals are engaged in design, they involve themselves with the conceptualization and realization of systems in the context of real-world constraints. [pp. 10-11]

The first two of these perspectives (mathematical theory, scientific method) are widely accepted as fundamental elements of any liberal arts college. The third expands a student's view of problem solving even further.

Throughout the various subject areas and problem-solving methodologies, a liberal arts-oriented undergraduate computer science curriculum emphasizes theory and core principles. In contrast to the perceptions of some incoming students, progress can be made on difficult problems only through careful analysis based on fundamental ideas. Further, while details of specific computer systems change rapidly, computer science embraces an underlying core of material which transcends such fleeting matters. A computer science curriculum within a liberal arts environment emphasizes such principles and thus is completely consistent with the spirit and value of a liberal arts education.

C Recommendations

Recommendation 2.1: An early section should highlight strong elements within the department under review.

- The $\backslash newrecommendation$ macro sets off a recommendation within the report's text and saves the recommendation for inclusion in the last appendix.
- The $\backslash newrecommendation$ macro numbers recommendations consecutively within a section (assuming the $\backslash recounter$ variable is reset to 0 at the start of each section, as shown throughout this document.
- The $\backslash newrecsublist$ macro inserts a bullet item under a $\backslash newrecommendation$ in the text and also in the appendix of recommendations.

Recommendation 3.1: The functioning of a department may support the high-level goals of a department, but sometimes practices seem counter productive.

Recommendation 4.1: The view of the department from the outside often has a direct impact on enrollments, advising, resources, facilities, and the level of cooperation present from other departments.

Recommendation 4.2: The department should review its programs, courses, and pedagogy to ensure that all activities connect with many students.

Recommendation 4.3: The department and university should review the image of the department on campus.

- Review should include Web pages, the university catalog, admission materials, etc.
- The department should identify opportunities to promote its programs, such as posters, admission events, family weekends, etc.

Recommendation 4.4: The department should look for ways to increase communication with other departments.

- Mathematics departments should understand the needs and interests of client departments.
- Mathematics faculty can use the Curricular Foundations Project [3] of the MAA as an excuse to renew conversations.

Recommendation 5.1: Beginning courses provide a significant recruiting opportunity.

- Non-majors' courses can excite students and encourage further exploration.
- Courses for non-majors and for majors should highlight applications as well as theory, so students appreciate the relevance of mathematics as well as its beauty.

- A curricular review should consider both content and pedagogy.

Recommendation 5.2: The CUPM Curriculum Guide [2] of the MAA can provide useful guidance in considering the breadth and depth of a majors' program.

Recommendation 6.1: Computing departments should mount a consistent and extensive public relations effort to recruit students and to counter negative societal stereotypes. One or more of the following themes connect with many students.

- Problem solving and intellectual challenge
- Ability of computing applications to help people
- Potential for fulfilling and exciting careers

Recommendation 7.1: External reviewers might schedule meetings with a range of administrative offices and departments. Meetings with some of the following may turn out not to be necessary, but a reviewer likely will not know that until after the review.

- the Office of Admission
- the Department of Information Technology Services
- an office of tutoring or academic advising