Annual Assessment and Program Review Report

Westmont College Computer Science Program

September 15, 2008

Contents

I. Mission Statement and Learning Outcomes

I.1. Computer Science Program Mission Statement

Core Values: Who We Are
Like many excellent and rigorous computer science programs around the world, we emphasize the fundamental and theoretical foundations of computation. At the same time, we ground the formal concepts in current advances in technology. Many aspects of the field are constantly changing, and staying current with new developments is a significant challenge. We believe that the best way to enable graduates to efficiently stay on top of the field is to lay a solid foundation of the fundamentals on which constant changes are ultimately based. We believe that the best way for us as faculty to remain current in the field is to conduct research and contribute to the field of computer science in our respective areas of emphasis. The formal foundation we lay serves our students well in their continued education in graduate school. For those who choose to enter the workforce immediately, this foundation equips them to be among the more versatile thinkers, learners and developers of their peers.

A number of schools provide similar rigor in their CS programs, yet two core values help set Westmont apart from other similar programs. First, the faculty commitment to research is coupled with an active intention and practice of including undergraduates in their research. These research opportunities for students greatly enhance the depth of their education and significantly improve their options for graduate school. The second distinguishing characteristic of CS at Westmont comes from the small intimate nature of the program. The faculty enjoy the opportunity and privilege of working one on one with students in course work to ensure that concepts are acquired. Thus, students are not left behind when difficult concepts are introduced as may happen in programs at larger universities. Similarly, our size allows us to support students when they want to combine CS with another major or to create an individual major.

Context: Where We Are
The excellence and rigor with which we pursue the fundamentals and the distinctive characteristics of our program are set against and within a rich multi-faceted context consisting of the liberal arts, Christian faith, social and community life and service, and simply a fantastic physical surrounding.

CS and the Liberal Arts
Studying CS at a liberal arts institution such as Westmont is particularly enriching because of the strong influences from the complete range of disciplines. A computer scientist's ability to solve problems is substantially enhanced by exposure and training

in analytical techniques practiced in history or literary criticism. Our understanding of diverse computer languages is broadened and deepened by exposure to multiple natural languages and the study of linguistics. The creativity necessary to construct complex algorithms that solve difficult problems is in part developed through exposure to the fine arts. More importantly, the ability to communicate with peers, advisors, managers, and customers is critical to a successful career in computer science; the liberal arts emphasis on communication enables our graduates to serve as leaders in a field that has traditionally suffered from low communication skills. In essence, a liberal arts education addresses the development of the whole person and we believe that building a rigorous mastery of computer science only makes sense within such a context.

CS and Christianity

Computer science at Westmont lives and breathes in the context of a Christian faith perspective. What this means for CS is that studies of computation inform our growing faith, and that faith guides our study of computer science. There are a number of ways that this interaction can take shape, but two of the most significant would be in the areas of service and leadership. As computers have become ubiquitous beyond any other technological device, our students have the opportunity to exercise Christian service through their chosen vocations, regardless of specialty. We seek to develop a heart of service in ourselves and our students. Just as importantly, computer scientists have the awesome responsibility of guiding a discipline that holds tremendous promise as well as danger for humanity. Fully embracing our faith, we provide and develop leadership by pursuing scholarship that engages issues that will impact our society now and in the future.

CS and Social Contexts

At Westmont, we have achieved and maintained a social community context that defies the norms and stereotypes of computer science. We recognize that in order to be effective in their future careers – either in graduate school and beyond or in the commercial sector – our students must be effective communicators and problem solvers in group contexts. As a faculty, we have taken this requirement and turned it into significant element of our nature. Most of our courses involve group projects of one kind or another; we actively address interpersonal issues and group dynamics inherent in the processes of software development and problem solving. Beyond the classroom, we foster community strength and interpersonal skills through regular social gatherings both on and off campus. In every situation, we seek to develop an enjoyment of each other as social beings created in God's image.

I.2. Student Learning Outcomes

1. **Core Knowledge.** Know the core ideas and methods in the field of computer science.
2. **Communication.** Be able to communicate ideas in writing or orally, following standard conventions of the discipline.
3. **Creativity.** Be able to independently learn new ideas and techniques and to formulate and solve a novel problem in computer science.
4. **Connections.** Be able to incorporate computer science knowledge and skill into a wider interdisciplinary framework and especially into a personal faith and its accompanying worldview.

These four learning outcomes are much the same as the four outcomes established for the joint department of mathematics and computer science. [See the 2007 annual update (external link).] As a result of separating the assessment and program review schedules for mathematics and computer science, the faculty in each program are now free to modify the learning outcomes to fit their disciplines. The changes made this year to the computer science outcomes are mostly minor adjustments to the language to conform to issues specific to computer science. The most significant change is the addition of the word "interdisciplinary" to the fourth standard, and the corresponding change in the name of the standard from "Christian Connection" to "Connections."

II. Data and Interpretation

II.1. Assessment of Learning Outcomes

The Learning Outcome Matrix in Table 1 summarizes the current assessment program in computer science. Many of the designations in this matrix are preliminary and will be reviewed and revised as necessary during the Spring of 2009.

The Computer Science Program Roadmap in Figure 1 expands on the information in the matrix, including extracurricular activities and other possible sources of assessment data.

The four subsections (II.1.i–iv) that follow these two graphics detail what has been done over the last academic year for each learning standard.

Learning Outcome Matrix: Major in Computer Science				
Student Learning Outcomes	Core Knowledge	Communication	Creativity	Connections
Meaning	Know the core ideas and methods in the field of computer science.	Be able to communicate ideas in writing or orally, following standard conventions of the discipline.	Be able to independently learn new ideas and techniques and to formulate and solve a novel problem in computer science.	Be able to incorporate computer science knowledge and skill into a wider interdisciplinary framework and especially into a personal faith and its accompanying worldview.
Introduced	CS 5, 10*, 15*	CS 10*, 15*	CS 10*, 15*	CS 15*
Developed	CS 30*, 45	CS 30*	CS 30*	CS 50, PHI 104, 113
Mastered	CS 105, 116, 120*, 125, 130*, 135, 140, 145	CS 120*, 130*, 192*, 195*, 198*	CS 192*, 195*, 198*, MA 180	CS 190SS, CS 195*, Emphasis courses from other depts.
Assessment strategy	Major Field Test in Computer Science	Writing samples scored with rubric	Externally reviewed presentations and papers	Reflective paper scored with rubric
Tentative benchmark	50% above 75th percentile	75% show improvement during term	50% produce some externally reviewed work	TBD
Links to learning standards	Critical-Interdisciplinary Thinking, Active Societal and Intellectual Engagement (Effective Participants)	Written and Oral Communication, Research and Technology	Critical-Interdisciplinary Thinking, Active Societal and Intellectual Engagement (Lifelong Learning, Responsibility)	Christian Orientation, Diversity, Active Societal and Intellectual Engagement (Christian Vocation)

Notes: $*$ = required for major. One of CS 50, PHI 104, PHI 113 is required for the major.

Table 1: Tentative learning outcome matrix.

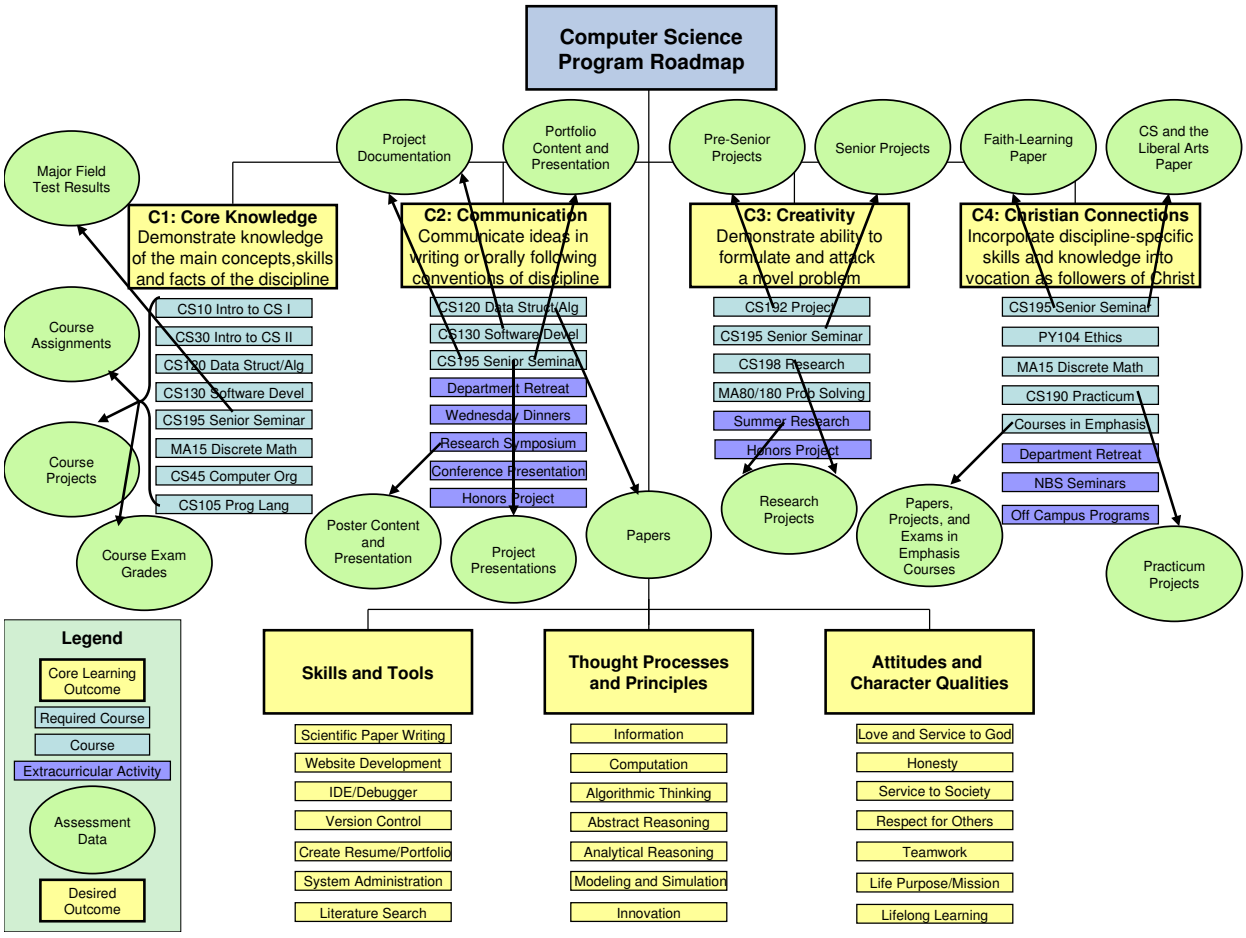

Figure 1: Computer Science Program Curriculum Roadmap.

II.1.i Core Knowledge

Starting with the class of 2008, all graduating seniors in computer science are required to take the ETS's Major Field Test in Computer Science (external link). The results for 2008 are as follows:

Student ID	Date Taken	Score	%ile
X	4/2/2008	165	≥ 80
Y	4/23/2008	170	≥ 85

Table 2: ETS Major Field Test in Computer Science results, 2008.

These results meet our tentative benchmark: 50% of students scored above the 75th percentile of students taking the test nationally.

The Educational Testing Service will not release subscore data until at least five people have taken the test, so we will have to wait until at least 2009 to get further information on subscores. Subscore data will tell us more than raw score data, because it will indicate areas in which students are doing well and areas which need improvement.

II.1.ii Communication

In August 2008, the computer science faculty met and formulated a rubric for evaluating student writing in CS courses. Data was collected from CS 130 (Software Development), a total of six papers. Both computer science faculty members read each paper, and evaluated them with respect to the rubric. This triggered a reorganization of the rubric and a lesson learned regarding the formulation of writing assignments and the clarity needed in the original prompt.

II.1.iii Creativity

The following students were involved in research during the past year:

Fall 2008: Ben Fisk (Senior)
Summer 2008: Kelsey Marshman (Fresh), Robin Elliot (Soph), Aaryn Smith (Soph), Ben Fisk (Junior)
Spring 2008: Toby Lounsbury (Soph)
Fall 2007: Adrian Rogers (Junior), Joel Stewart (Junior)
Summer 2007: Daniel Rufener (Soph), Michael Magnason (Grad), Adrian Rogers (Soph), Joel Stewart (Soph)

The following papers were accepted for publication in peer-reviewed journals (* indicates student author).

W. Iba, K. Marshman*, and B. Fisk*, Evaluating a Parallel Evolutionary Algorithm on the Chess Endgame Problem, in *Proceedings of the 2008 International Conference on Genetic and Evolutionary Methods*, Las Vegas, Nevada, July 2008, to appear.

K. P. Kihlstrom, R. S. Elliott*, K. A. Marshman*, and A. C. Smith*, Intrusion-Tolerant Dissemination in Large-Scale Systems, in *Proceedings of the 2008 International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, Nevada, July 2008, to appear.

K. P. Kihlstrom, J. L. Stewart*, N. T. Lounsbury*, A. J. Rogers*, and M. C. Magnuson*, Implementation and Performance Testing of a Gossip-Based Communication System, in *Proceedings of the 19th IASTED International Conference on Parallel and Distributed Computing and Systems*, Cambridge, MA, November 2007, pp. 194-199.

The following posters and presentations were made.

Joel Stewart*, Daniel Rufener*, Kelsey Marshman*, Ben Fisk*, and Wayne Iba , *Evaluating a Parallel Evolutionary Algorithm on the Chess Endgame Problem*, poster at Westmont Student Research Symposium, April 2008, and at Southwest Regional Conference of the Consortium for Computing Sciences in Colleges, April 2008 (received Best Student Poster Award).

K. P. Kihlstrom, J. Stewart*, N. T. Lounsbury*, M. C. Magnuson*, and A. Rogers*, *Intrusion-Tolerant Dissemination in Large-Scale Systems*, presentation at the Forty-first Annual Hawaii International Conference on System Sciences, Waikoloa, HI, January 2008.

J. Stewart*, A. Rogers*, N. T. Lounsbury*, M. C. Magnuson*, and K. P. Kihlstrom, *Implementation and Performance Testing of a Gossip-Based Communication System*, poster at Westmont Summer Research Celebration, September 2007, and at Westmont Student Research Symposium, April 2008.

N. T. Lounsbury*, A. Rogers*, J. Stewart*, M. C. Magnuson*, and K. P. Kihlstrom, *StarblabFS: Replicated userspace file system*, poster at Westmont Summer Research Celebration, September 2007, and at Westmont Student Research Symposium, April 2008.

Undergraduate research clearly remains an area of strength for the department. These peer-reviewed results represent the work of a range of students of varying abilities, from our outstanding graduate of 2007 to a student who failed introductory classes. These results easily exceed our tentative benchmark that 50% of graduates produce some peer-reviewed work.

II.1.iv Connections

Assessment work on the fourth learning standard will be the focus of assessment discussions in 2008-2009.

II.2. Interpretation of Results

The preliminary indications of our assessment work so far suggest that the program is effectively promoting the outcomes of Core Knowledge and Creativity. The other two outcomes, Communication and Connections, are going to be harder to assess, and work on these outcomes will be a priority for the next few years.

III. Using the Results and Next Steps

The computer science curriculum has undergone continual revision since its reinception in 1999. This revision has been necessitated by the nature of the discipline: on the national level, technology, techniques, and pedagogy have undergone many changes in past years.

While we do not yet have enough results of assessed data to justify changes in the computer science curriculum, the process of crafting learning outcomes and assessment strategies has helped focus the design of our curriculum.

The current 2008–2009 catalog reflects the most recent changes to the computer science major. These changes streamline the core of the major while allowing flexibility for pursing various emphases, within and beyond the discipline. The curriculum roadmap illustrates how our assessment strategy is related to the current redesigned major. To ensure inputs and provide assessment data for the Communication, Creativity, and Connections outcomes, all students are required to take a senior seminar, participate in research, produce significant written work in Software Development, and study ethics, preferably from a disciplinary perspective in CS 50.

The focus for the next academic year will be the fourth learning outcome: Connec-

tions. In particular, the computer science faculty plan to do the following:

- Decide on a mechanism and prompt for a reflective paper.
- Design a rubric for scoring the reflective paper.
- Collect writing samples.
- Apply the rubric and revise as necessary.
- Decide on appropriate benchmarks.

In addition, the program plans to continue the assessment activities of the other learning outcomes:

- Administer the MFT, discuss the results, and revisit benchmarks.
- Collect and analyze more writing data and revisit benchmarks.
- Continue to involve students in peer-reviewed research.

Throughout this process, the computer science faculty will consider revisions to several components of this report: the learning outcome matrix, the curriculum roadmap, and the six-year plan.

IV. Data for Program Review

The next several pages contain data that will be helpful when the six-year report is compiled in 2014.

Faculty Information for Wayne Iba, 2007–2008

Date of Hire: 2003

Sex and Ethnicity: Male Caucasian

Rank: Associate Professor

Tenure Status: Untenured

Teaching Load:

	Classes	# students	# advisees	Other departmental responsibilities	New Preps
Fall 2007			20		1
	CS 10	18			√
	CS 105	3			
Spring 2008			12		1
	CS 130	4			
	CS 150	4			√
Average load	2	7.25	16		1
Mayterm					
Ind. Studies					
Internships					

Table 3: Teaching load for Wayne Iba, 2007–2008.

Note: Computer Science faculty currently teach 5 courses per year. Wayne Iba received a course release for chairing the Academic Resources Committee.

Faculty Information for Kim Kihlstrom, 2007–2008

Date of Hire: 1999

Sex and Ethnicity: Female Caucasian

Rank: Associate Professor

Tenure Status: Tenured

Teaching Load:

	Classes	# students	# advisees	Other departmental responsibilities	New Preps
Fall 2007	CS 5 CS 30	10 5	23		0
Spring 2008	CS 30 CS 45	15 2	19		0
Average load	2	8	21		0
Mayterm					
Ind. Studies					
Internships					

Table 4: Teaching load for Kim Kihlstrom, 2007–2008.

Note: Computer Science faculty currently teach 5 courses per year. Kim Kihlstrom had a grant pay for a course release this year.

Computer Science Alumni Survey

LAST	FIRST	MIDDLE	Deg.	YEAR	Current Ocupation (title, company, etc.)
Eastvold	Benjamin	Seth	BS	2001	
Chambers	Luke	Wilson	BA	2002	
Te Velde	Michael	Henry	BA	2002	
Conant	Tyler	John	BS	2002	
Andrews	Christopher	Richard	BA	2003	
Hanson	James	Daniel	BS	2003	
Lassonde	Nicholas	Adam	BS	2003	Startup company (Vesora?)
Gibbs	Anna	Katheryn	BS	2004	
Labarbera	Benjamin	Carl	BS	2004	
Carroll	Dustin		BS	2004	Grad student finishing Masters at MBARI
Hanson	James	Rodney	BS	2004	
Thurman	Jesse	Lee	BS	2004	
Sullivan	Jonathan	Robert	BS	2004	System admin consultancy
Cox	Joseph	Walter	BS	2004	Cytrix
Kishiyama	Kevin	Takeo	BS	2004	Software developer
Cook	Patti	Lee	BS	2004	
Broadus	Robert	Lee	BS	2004	IT
Roby	Seth	A.	BS	2004	
Jensen	Brian	Alan	BS	2005	Opera singer
Ritchey	Christopher	Charles	BS	2005	
Phillips	Christopher	Dale	BS	2005	Grad student finishing Masters at UCSB and developer at Callwave
Johnson	Kerby	Obadiah	BS	2005	Finished Masters at UCSB and working Genentec
Yankoski	Michael	George	BS	2005	
McElfresh	Rebecca	Joy	BS	2005	
Newell	Ryan	M	BS	2005	
Cantrell	Thomas	Bruce	BS	2005	Software developer at Green Hills Software (SB)
Jensen	Bradley	Alan	BS	2006	
Shank	Daniel	Philip	BS	2006	Software developer/tester at Appfolio (SB)
Hilp	John	Tilford	BS	2006	
Kaddatz	Matthew	Martin	BS	2006	
Burwell	Nicholas	James	BS	2006	Software developer at Callwave
Watters	Kyle	Padia	BS	2006	PhD grad student in Physics at Stanford
Evans	Anne	Iten	BS	2007	High School computer science teacher, CO
Cannon	Cuyler	Elisha	BS	2007	
Holm	Joshua	Thor	BS	2007	Software developer at Callwave
Magnuson	Michael	Carl	BS	2007	
Raub	Stephen	Benjamin	BS	2007	Software developer at Green Hills Software (SB)
Willson	Julia	Kathleen	BS	2007	
Barquer	Kristin	Noel	BS	2007	Grad student in Kinesiology (Univ Iowa?)

Work with other departments and offices:

In September 2006, Dana Alexander and Celia Howen from the Office of Life Planning spoke at departmental retreat for mathematics and computer science.

On 9/7/2007, Kim Kihlstrom and Dave Hunter met with Jennifer Taylor of the Internship Office. Jennifer helped us implement a new course, CS190/191SS, Computer Science Seminar for Service Learning Internship, which satisfies the Serving Society component of General Education. This office and also OLP contribute to our fourth learning standard (Christian Connection) by helping students understand how to use computer science to serve others.

The Mathematics and Computer Science Department met with our library liaison, Claudia Scott, on 10/16/2007. The consensus at that meeting was that there are no major shortcomings in our collection that can be reasonably addressed at a small college.

V. Time-line for Completion of the Six-year Report

After discussions with Assessment Coordinator Marianne Robins, the department agreed to separate the program review schedules for mathematics and computer science. The Mathematics Program is submitting a Six-Year report this year (2008), while the Computer Science program will not submit a Six-Year report until 2014. Table 5 gives the current six-year plan for assessment and program review in computer science. This plan will be reviewed and revised in the 2008-2009 academic year.

Year	Program review overall	Details for assessment work
2008–09	Submit annual progress report, 9/15/2008. Justify need for a third department member using program review data. Discuss admissions practices and recruitment of students.	Ongoing annual tasks. Review Connections learning standard (#4).
2009–10	Actively pursue the creation of a third faculty line in computer science, using program review data.	Ongoing annual tasks. Review Core Knowledge learning standard (#1).
2010–11	Discuss the implications of a third faculty line for curriculum and program review. Discuss the transition to a separate computer science department, once a third faculty member is hired.	Ongoing annual tasks. Review Creativity learning standard (#3).
2011–12	Review contributions to GE.	Ongoing annual tasks. Review Communication learning standard (#2).
2012–13	Discuss quality of preparation for graduate school.	Ongoing annual tasks. Discuss/Revise learning standards.
2013–14	Prepare for six-year program review report, due 9/15/2014.	Ongoing annual tasks. Summarize assessment work from past six years.

Table 5: Tentative six-year plan for assessment and program review in computer science. This plan will be revised during the 2008–2009 academic year.