Computer Science Program 2010 Annual Assessment Update

I. Mission Statement and Student Learning Outcomes

A. Mission Statement

Our mission statement is unchanged from our previous report:

> We desire to transform our program into one that efficiently and sustainably pumps students up to their full potential in computer science.

We have not changed our detailed vision statement for the program. This statement is included in the appendix and also available on our department's assessment archive at: \\myfiles\program_review\Computer_Science\Guiding Documents\2010-Mission Statement and Learning Outcomes.pdf

B. Student Learning Outcomes

Our four student learning outcomes remain unchanged from our previous report and are as follows:

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.

C. Alignment

We have prepared an alignment document giving the alignment of our courses, departmental outcomes, and institutional outcomes. This information is included in the appendix and also available on our department's assessment archive at: \\myfiles\program_review\Computer_Science\Guiding Documents\2010-Department Goals Alignment.pdf

II. Follow up on Action Items identified in Previous Reports

We will list each of the five action items identified on page 11 of our 2009 annual report, and then give an update on each item. (Because one of our two CS faculty members, Wayne Iba, has been on a year-long sabbatical, the responsibility for each item has been given to Kim Kihlstrom, the other CS faculty member.)

Action Item #1:

- Decide on a mechanism and prompt for reflective papers in first-year and senior seminars. (Responsibility: Kim Kihlstrom) (Note: this item pertains to our "Connections" outcome)

Update:
We developed the following prompt for our Fall 2009 first-year seminar course (CS 50):

> You are to write a term paper of 2000 words that provides a well-reasoned reply to the question, "What is the role of philosophy in a Christian liberal arts education with respect to understanding the nature of information and computation, how they interact with the human race, and how we

make value judgments regarding them?" You must include a list of works cited that includes at least 2-3 outside scholarly sources in addition to the course readings.

We have not yet developed a prompt or a rubric for our senior seminar (CS 195), as we have not yet offered the course. We will do so in Spring 2011.

Action Item #2:

- Apply the rubric to student papers written during Fall 2009 and revise as necessary. (Responsibility: Kim Kihlstrom)

Update:
Our rubric is given in the appendix. This rubric was supplied along with the assignment prompt to all 23 students prior to the writing of their papers.

We found that the rubric was too vague and had too many gradations that were difficult to distinguish. As a result, we revised our rubric as shown in the appendix. We will use this revised rubric to evaluate student papers in the Fall 2010 CS 50 course.

Action Item #3:

- Decide on appropriate benchmarks. (Responsibility: Kim Kihlstrom)

Update:
Our tentative benchmark is this: in each outcome category listed in the Connections rubric, 70% of students receive a score of acceptable or exemplary.

Action Item #4:

- Review the use of the MFT to assess students' acquisition of core knowledge. (Responsibility: Kim Kihlstrom)

Update:
We have now collected results on the MFT for three years, 2008-2010. These results will be summarized and discussed in Section III below.

Action Item #5:

- Review tentative benchmark for student performance on the MFT. (Responsibility: Kim Kihlstrom)

Update:
As discussed below, we are revising our benchmark as follows: 50% of students score at or above the 70th percentile on the MFT.

III. 2010 Focus

A. Connections

We plan to assess the Connections outcome in two courses: CS 50, a first semester course, and CS 195, a senior seminar. As indicated above, we will offer CS 195 for the first time in Spring 2011, and we will assess the outcome then.

We applied our rubric to all 23 student papers written in our first year seminar (CS 50) during Fall 2009. The results are as shown in the appendix, and can be summarized as follows:

Outcome	% of students scoring acceptable or above
Articulates and thinks critically about the foundational questions of philosophy regarding the nature of information and computation	91
Articulates and thinks critically about the foundational questions of philosophy regarding how information and computation interact with the human race	91
Articulates and thinks critically about the foundational questions of philosophy regarding how we make value judgments regarding information and computation	87
Articulates the relationship between philosophy and the Christian liberal arts	91
List of works cited includes at least 2-3 outside scholarly sources in addition to the course readings	87
Mechanics (grammar, spelling, format)	100
Length (2000 words nominal)	96

The data indicates that students are successfully able to make connections between faith, philosophy, and aspects of computer science including information and computation. The actual student work is stored on our department's assessment archive at: \\myfiles\program_review\Computer_Science\Assessment Data\C4 Connections\2009-CS 050 1-Student Work.pdf

As discussed in Section II above, the rubric was vague and it was difficult to distinguish between the gradations, particularly between "Excellent" and "Good." In the Fall 2010, we will use our modified rubric (given in the appendix). We were not able to perform collective evaluation on this data, as one of the two CS faculty members, Wayne Iba, went away on a year-long sabbatical immediately after this data was collected.

B. Core Knowledge

1. MFT

We assess our Core Knowledge outcome through administration of the Major Field Test (MFT) in Computer Science. We have now collected results on the MFT for from nine students over three years, 2008-2010. These results are summarized as follows:

Data Set	Year	Score	Percentile
Student 1	2010	132	10
Student 2	2010	159	70
Student 3	2010	164	75
Student 4	2010	156	60
Student 5	2010	143	35
Student 6	2008	165	80
Student 7	2008	170	85
Student 8	2009	160	70
Student 9	2009	154	60
Institutional Mean	2008-2010	156	70

We have mixed feelings about the results of the exam. Student 1, who scored the lowest of any of our students on the MFT, was one of our most outstanding students, and our first student to undertake and complete (with distinction) a major honors project. Thus, we are not convinced that the exam is an accurate reflection of students' acquisition of core knowledge. As suggested by the program review committee's response to our annual report from last year, we need to determine whether all of the subsections of the MFT test are in alignment with our program. We have not yet requested subsection results, but we plan to do so next year. The data is stored on our department's assessment archive at: \\myfiles\program_review\Computer_Science\Assessment Data\C1 Core Knowledge\2010-Major Field Test Results.pdf

Our benchmark was that 50% of students would score at or above the 75th percentile. Three students of the nine who took the exam, or 33.3%, scored at or above the 75th percentile, so we did not achieve the benchmark. On the other hand, five of the nine, or 55.5%, scored at or above the 70% percentile. If the benchmark was revised to 50% of students scoring at or above the 70th percentile, we would meet the benchmark. We believe this is a reasonable benchmark, and so will use the updated benchmark.

The mean score for all students taking the exam at our institution was 156. This puts us at the 70th percentile among all institutions administering the exam. In a liberal arts program such as ours, we offer a comparatively small curriculum and require a relatively small number of units for the major, in comparison with most computer science programs around the world. Thus, we feel that the results of the MFT are quite positive and validate that our program, while limited, is accomplishing our goal of giving students the core knowledge required in the field of computer science.

2. First Year Sequence (CS 10 and CS 30)

On July 7, 2009, Wayne Iba and Kim Kihlstrom met to collectively evaluate our first year sequence, CS 10 and CS 30. We have observed that large numbers of students drop out during the first year. This is not good for our program for several reasons.

1. We can't achieve our mission to "pump students up to their full potential in computer science" if the students don't remain in the program even through the first year
2. We can't hold effective class sessions without a critical mass of students enrolled
3. We can't offer the courses needed without a critical mass of students to take them
4. We can't utilize students as help session leaders, graders and TAs if there are no students to perform these functions
5. The students will not develop group problem solving skills if there are no students with whom to collaborate
6. The students will not develop group communication skills if there are no students with whom to communicate

In Spring 2009, an adjunct faculty member taught CS 30, and we experienced a particularly high dropout rate for that course. We obtained data from the Registrar's Office and found that, of thirteen students registered for CS 30 in Spring 2009, eight students (61.5%) dropped. In contrast, in Fall 2006, none of the seven students (0%) dropped, in Fall 2007, one student out of six (16.7%) dropped, and in Spring 2008, one student out of fifteen (6.6%) dropped. This data is available on our department's assessment archive at: \\myfiles\program_review\Computer_Science\Assessment Data\C1 Core Knowledge\2005-2010 CS-030 Registration.pdf We conducted interviews with students who dropped the course in Spring 2009 and have determined that the high dropout rate was directly to the use of an adjunct to teach this course. We will discuss this further below, under staffing issues.

In an effort to more fully understand and bridge the transition between CS10 (taught in Scheme) and CS30 (taught in C++), Kim Kihlstrom attended a TeachScheme/ReachJava workshop during summer 2009. We discussed this during our July 7, 2009 meeting, and began to talk about the specific language that might be taught in CS 30 in the future. We identified three qualities that would be desirable in a language for CS 30:

- Minimal syntax
- Facilities for testing (test-driven development)
- Graphics package

As a result, we restructured the CS 30 course in Spring 2010, including the use of a new programming language for that course, Ruby. The language was found to satisfy all of the desired qualities, and was well received by the students. In Spring 2010, two students out of eleven (18%) dropped; one of these students left Westmont to pursue a business venture, and the other was a staff member attempting to take the course for enrichment in addition to holding a full time job.

In the past year, we have become increasingly concerned about the number of students who drop out of our first semester course, CS 10. We obtained data from the Registrar's Office regarding students who have registered for CS 10 in the past five years. This data is available on our department's assessment archive at: \\myfiles\program_review\Computer_Science\Assessment Data\C1 Core Knowledge\2005-2009 CS-010 Registration.pdf

Out of 173 students registering for CS 10 in the years 2005-2009, 71 students (41%) dropped the course. This percentage is unacceptably high.

We then undertook a survey of the 173 students who registered for CS 10 in the years 2005-2009. The survey instrument is available on our department's assessment archive at: \\myfiles\program_review\Computer_Science\Guiding Documents\2005-2009-CS-010 Survey.pdf. We received complete survey responses from 44 students and partial responses from 6 additional students. As one partial response contained no usable information, we will consider the total number of responses to be 49. The responses are available on our department's assessment archive at: \\myfiles\program_review\Computer_Science\Assessment Data\C1 Core Knowledge\2005-2009-CS 10 Survey Responses.pdf

Here are some key findings from the survey:

- 16 students (32.6%) commented negatively on the use of Scheme as the programming language. These comments came in the form of two basic response statements:
  1. The course could be improved by changing to a different language
  2. The hardest aspect of the course was the syntax of the language
- 21 students (53%) reported that the hardest aspect of the course was the amount of time required to complete the homework. Additional comments included these statements:
  - "I worked harder on that class than my others combined and still had NO idea how to do the midterm project."
  - "I was completely lost all the time, even when I asked for help.... It would have taken way too much time away from my other homework"

In Fall 2010, we are teaching CS 10 in Ruby rather than Scheme, which we hope will address some of these concerns. We are also being conscious of the homework assignments chosen, attempting to keep within normal expectations for an introductory course (two hours of work outside class for every one hour inside class). We are designing in-class laboratory assignments that aim to prepare students for the homework assignments, and we are holding weekly help sessions.

We will administer the same survey to the students that registered for CS 10 in Fall 2010, and compare with the results obtained previously. We will also continue to analyze the results of the survey and use that information to look for ways in which we can improve our first year sequence.

C. Staffing Issues

In our multi-year plan for program review submitted in our 2009 annual assessment update, we identified the following overall **program review goal:**

- Actively pursue the creation of a third faculty line in computer science, using program review data.

We discuss will this issue here.

Computer science as a discipline has been changing extremely rapidly, requiring constant modifications to the courses and curriculum, and continuous learning of new techniques, languages, and paradigms. In the short time we have been offering the program, we have made numerous changes to the curriculum, and taught the introductory sequence in a total of four different languages (C++, Java, Scheme, and Ruby.) The upper division courses, which are primarily offered every other year, have in general needed to be completely restructured each time they are offered.

It is very difficult to offer a complete computer science major with only two CS faculty members. **No other science program in the college attempts to offer a major with only two faculty members in the discipline.** In order to offer a computer science major with only two CS faculty members, we have needed to make extensive use of adjuncts, as shown in the table below.

Academic Year	Number of CS courses taught by adjuncts	Number of CS courses taught by math faculty*	Number of CS courses taught by CS faculty
2005-2006	3	0	7
2006-2007	1	0	8
2007-2008	0	1	8
2008-2009	2	0	6
2009-2010	1	0	8
2010-2011	2	0	8
Total	9	1	45

*Excludes MA/CS 15 and MA/CS 135, which were taught seven times by math faculty

In the years 2005-2011, 45 CS courses were taught by CS faculty, nine CS courses were taught by adjuncts, and one CS course was taught by a math faculty. These numbers exclude two cross-listed courses, MA/CS 15 and MA/CS 135, which were taught seven times by math faculty. Thus, 10 out of 55 CS courses (18%) were taught by non-CS faculty. None of the adjuncts had a Ph.D. We believe this is an unacceptably high number of CS courses taught by faculty without a Ph.D. The large dropout rate (61.5%) in CS 30 during Spring 2009 was directly attributable to the use of an adjunct.

We have now offered the CS major for eleven years; four of those years with only one CS faulty member, Kim Kihlstrom. Since the hiring of Wayne Iba in 2003, Kim Kihlstrom and Wayne Iba have each gone on sabbatical for one semester and two semesters, respectively, and Kim Kihlstrom has/will also serve the college on the Europe Semester program in Fall 2008 and Fall 2011. Thus, in the years 2004-2011 there have been/will be an additional five semesters of offering a CS major with just a single CS faculty member.

The burdens of offering a major with only one or two faculty members in the discipline extends beyond the regular course teaching load to include implications for advising, research involving students, supervising internships, supervising junior and senior projects, supervising serving society internships, supervising major honors projects, and administrative assignments such as program review, collective assessment, and writing annual assessment reports. We formally requested a third position in CS in January 2008; this document is available on our department's assessment archive at: \\myfiles\program_review\Computer_Science\Records\2008-Request for Faculty Position.pdf

Not only have we been offering a science major with only one to two faculty members, we have also been offering the major without the services of a lab coordinator, a position that most of the other sciences (physics, biology, and chemistry) have enjoyed for a number of years. We formally requested a lab coordinator in August 2008; this document is available on our department's assessment archive at: \\myfiles\program_review\Computer_Science\Records\2008-Request for Lab Coordinator.pdf

The lack of a lab coordinator for computer science is creating significant challenges. Most recently, the Kim Kihlstrom's research program involving students has come to a complete halt due to a lack of staffing to maintain the servers and workstations on which this work is conducted. Problems with these machines have been increasing over time, coming to a head in Spring 2010, and now further exacerbated by the move to Winter Hall. Efforts to utilize student help and IT staff to correct these issues have been insufficient.

Research with students is the primary factor in achieving and assessing our Creativity learning outcome. In Spring 2010, Kim Kihlstrom was engaged in a significant research project involving her CS 150 Distributed Systems course. The goal of the course was to perform final testing and debugging of an intrusion tolerant distributed system that has been developed at Westmont and supported by NSF, and then to produce a journal paper describing the results. The class was making significant progress in this effort until the lab machines experienced major problems. These problems we not resolved despite massive efforts by the students, and consulting by IT staff, and are still not resolved. **Hiring of a lab coordinator** would be the most obvious solution to problems such as these, and **would allow us to achieve and assess our Creativity learning outcome.**

IV. Next Steps

A. Action Items

We have identified the following action items for the next year:

- Review Creativity learning standard (Wayne Iba, Spring 2011)
- Use modified rubric for assessing Connections in CS 50 Fall 2010 and perform collective evaluation (January 2011, Wayne Iba and Kim Kihlstrom)
- Developed a prompt and a rubric for assessing Connection in our senior seminar (Spring 2011,Wayne Iba and Kim Kihlstrom)
- Administer survey to students who registered for CS 10 in Fall 2010 (December 2010, Kim Kihlstrom)
- Obtain subsection results for MFT and analyze (June 2011, Wayne Iba)
- Record feedback from outside consultant and steps taken as a result (June 2011, Kim Kihlstrom and Wayne Iba)

B. Multi-Year Assessment Plan

Our multi-year assessment plan is given in the appendix and available on our department's assessment archive at: \\myfiles\program_review\Computer_Science\Guiding Documents\2010-Multi-Year Assessment Plan.pdf

V. Appendices

A. Detailed Vision 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 in 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.

B. Rubric used in assessing "Connections" outcome in CS 50, Fall 2009

	0 Unacceptable	1 Poor	2 Acceptable	3 Good	4 Excellent
Articulates and thinks critically about the foundational questions of philosophy regarding the nature of information and computation					
Articulates and thinks critically about the foundational questions of philosophy regarding how information and computation interact with the human race					
Articulates and thinks critically about the foundational questions of philosophy regarding how we make value judgments regarding information and computation					
Articulates the relationship between philosophy and the Christian liberal arts					
List of works cited includes at least 2-3 outside scholarly sources in addition to the course readings					
Mechanics (grammar, spelling, format)					
Length (2000 words nominal)					

C. Revised rubric that will be used to assess "Connections" outcome in CS 50, Fall 2010

Outcome	Underdeveloped	Acceptable	Exemplary
Foundational questions of philosophy regarding information and computation	The presentation or identification of philosophical ideas is missing or confused.	The presentation or identification of philosophical ideas is correct but minimally developed.	The presentation or identification of philosophical ideas is correct and well developed with coherent and convincing supporting arguments.
Components of a Christian liberal arts education and the interrelation of philosophy	No components are named or what is named is confused or irrelevant.	Some components of a Liberal Arts Education are identified and at least minimally related to philosophy or other areas of study.	Key components of a Liberal Arts Education are artfully connected to both philosophy and other areas of study in a multifaceted way that includes both content and skills.
Relationship between philosophical commitments and an integrated life	No discernable relationship between philosophical commitments/academic life and an integrated life is articulated or the articulated relationship is trivial.	A meaningful (but perhaps narrow) relationship between philosophical commitments/academic life and an integrated life is articulated with modest development.	A rich and meaningful relationship between philosophical commitments/academic life and an integrated life is articulated and developed with meaningful examples.
List of works cited	no works cited, or works cited are not scholarly works	Only 1 scholarly work cited in addition to the course readings	2-3 outside scholarly sources in addition to the course readings
Mechanics (grammar, spelling, format)	numerous grammatical, spelling, or formatting errors	a few grammatical, spelling, or formatting errors	no significant grammatical, spelling, or formatting errors
Length (2000 words nominal)	more than 20% outside nominal length	10% -20% outside nominal length	within 10% of nominal length

D. Data from assessment of Connections outcome, in CS 50, Fall 2009.

The number of students receiving each score is indicated in the body of the chart

	0 Unacceptable	1 Poor	2 Acceptable	3 Good	4 Excellent
Articulates and thinks critically about the foundational questions of philosophy regarding the nature of information and computation	0	2	3	4	14
Articulates and thinks critically about the foundational questions of philosophy regarding how information and computation interact with the human race	0	2	3	4	14

Articulates and thinks critically about the foundational questions of philosophy regarding how we make value judgments regarding information and computation	0	3	2	1	17
Articulates the relationship between philosophy and the Christian liberal arts	0	2	0	1	20
List of works cited includes at least 2-3 outside scholarly sources in addition to the course readings	2	1	1	0	19
Mechanics (grammar, spelling, format)	0	0	0	6	17
Length (2000 words nominal)	0	1	4	4	14

E. Multiyear Assessment Plan

Year	Program review overall	Details for assessment work
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.