UH Hilo Assessment Committee

2007-2008

Deliverable #1: Matrix of Program Outcomes and Courses

The workshop by Dr. Mary Allen on Feb 4-5, 2008, will address the development of program outcomes.

Degree or Program Name: Computer Science (BS)
Program / Department Chair: Dr. Judith Gersting gersting@hawaii.edu
Revision Date: February 19, 2008

Outcome 1: Understand classical algorithmic processes and data structures and be able to perform simple analysis of algorithms

Outcome 2: Be proficient in one high-level programming language and have basic skill levels in a variety of programming languages

Outcome 3: Understand the basics of logic design and computer organization and be aware of multiple architecture approaches and numerical limitations

Outcome 4: Be competent in techniques of discrete mathematics and understand the theoretical foundations of computing

Outcome 5: Understand the steps of the software development process and the activities/products appropriate to each

Outcome 6: Know the major issues in the design and implementation of major computing artifacts such as operating systems, programming languages, graphical user interfaces or systems programming, and databases, networks, or compilers

Outcome 7: Be able to adapt to changing development platforms and design/implementation tools

Outcome 8: Communicate effectively on technical matters in both oral and written forms, work well within a team, and understand the social/ethical issues of computing

Courses for Majors	Require Elective	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6	Outcome 7	Outcome 8
COM 251	**R**								**D**
ENG 209	**One R**								**D**
ENG 225									**D**
MATH 205	**R**	**P**			**P**				
MATH 206	**R**				**P**				
MATH 311	**R**			**P**					
PHYS 170 - 170L	**R**			**P**					
PHYS 171 - 171L	**R**			**P**					
CS 141 – Discrete Mathematics for Computer Science 1	**R**	**I**		**I**	**I**				**I**
CS 150 - Introduction to Computer Science	**R**	**I**	**I**			**I**		**I**	
CS 151 - Introduction to Software	**R**	**D**	**D**			**D**			

Development									
CS 241 – Discrete mathematics for Computer Science II	**R**				**D**				
Courses for Majors	Require Elective	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6	Outcome 7	Outcome 8
CS 266 - Computer Organization and Assembly Language	**R**			**D**				**I**	
CS 321 - Data Structures	**R**	**M**	**M**						**D**
CS 340 – GUI	**One R**						**D**	**D**	
CS 350 – Systems Programming			**D**				**D**	**D**	
CS 407 – Introduction to Numerical Anaylsis I	**R**		**D**	**M**				**D**	
CS 410 - Elements of Computer Architecture	**R**			**M**				**D**	**D**
CS 420 - File Management	**R**						**M**	**D**	
CS 430 - Operating Systems	**R**		**D**				**M**	**M**	
CS 450 - Organization of Programming Languages	**R**		**M**				**M**	**M**	
CS 460 - Software Engineering I	**R**		**M**			**M**		**M**	**M**
CS 461 - Software Engineering II	**R**		**M**			**M**		**M**	**M**
CS 470 - Theory of Computing	**R**				**M**				
CS 495 - CS Professional Seminar	**R**								**M**
CS 421 – Database Management System Design	**Two R**						**M**	**M**	
CS 431 – Computer Networks and Data Communication							**M**	**M**	
CS 451 – Compiler Theory			**D**				**M**	**M**	

I = Introduced, D = Developed & Practiced with Feedback, M = Demonstrated at the Mastery Level appropriate for graduation in this program, P = Prerequisite to Success in Later Courses

Deliverable #2: Assessment Plan for Program Outcomes

The workshop by Dr. Mary Allen on Feb 4-5, 2008, will address the development of an assessment plan.

Degree or Program Name: Computer Science
Program / Department Chair: Dr. Judith Gersting gersting@hawaii.edu
Revision Date: February 22, 2008

Program Outcome (same outcomes as Deliverable #1)	Assessment Plan	Status / Progress / Results
		All these assessment mechanisms are in place. The MFAT has been given for 20 years.
1. Understand classical algorithmic processes and data structures and be able to perform simple analysis of algorithms	Successful completion of projects and exams in CS 321. Above national average in MFAT sub-area **Discrete Structures and Algorithms**	
2. Be proficient in one high-level programming language and have basic skill levels in a variety of programming languages	Successful completion of projects and exams in CS 321 and CS upper division courses (except CS 470 and 495). Above national average in MFAT sub-area **Programming Fundamentals**	
3. Understand the basics of logic design and computer organization and be aware of multiple architecture approaches and numerical limitations	Successful completion of projects and exams in CS 266, CS 410, and CS 407. Above national average in MFAT sub-area **Systems (Architecture, Operating Systems, Networking, Database)**	
4. Be competent in techniques of discrete mathematics and understand the theoretical foundations of computing	Successful completion of assignments and exams in CS 141, 142, and 470. Above national average in MFAT sub-area **Discrete Structures and Algorithms**	
5. Understand the steps of the software development process and the activities/products appropriate to each	Contribute to production of a successful software engineering project in CS 460-461	
6. Know the major issues in the design and implementation of major computing artifacts such as operating systems, programming languages, graphical user interfaces or systems programming, and databases, networks, or compilers	Successful completion of projects and exams in CS 430, 450,340 or 350, and 421, 431, or 451. Above national average in MFAT sub-area **Systems (Architecture, Operating Systems, Networking, Database)**	
7. Be able to adapt to changing development platforms and design/implementation tools	Successful completion of CS upper division courses (except CS 470 and 495).	
8. Communicate effectively on technical matters in both oral and written forms, work well within a team, and understand the social/ethical issues of computing	Write and present meaningful reports in CS 141, 321, 450 and 460-461. Make meaningful contributions to group work in CS 141, CS 321, and CS 460-461. Successful completion of assignments in CS 495	

Deliverable #3: WASC Rubrics (Self-Assessment Checklist)

Degree or Program Name: Computer Science (BS)
Program / Department Chair: Dr. Judith Gersting gersting@hawaii.edu
Revision Date: February 22, 2008

Check the status of your program's learning outcomes for each WASC criteria below.

Comprehensive List:	___ Initial	___ Emerging	_X_ Developed	___ Highly Developed
Assessable Outcomes:	___ Initial	___ Emerging	_X_ Developed	___ Highly Developed
Alignment:	___ Initial	___ Emerging	___ Developed	_X_ Highly Developed
Assessment Planning:	___ Initial	___ Emerging	_X_ Developed	___ Highly Developed
The Student Experience:	___ Initial	_X_ Emerging	___ Developed	___ Highly Developed

Rubric for Assessing the Quality of Academic Program Learning Outcomes—August 10, 2007 Draft

Criterion	Initial	Emerging	Developed	Highly Developed
Comprehensive List	The list of outcomes is problematic: e.g., very incomplete, overly detailed, inappropriate, disorganized. It may include only discipline-specific learning, ignoring relevant institution-wide learning. The list may confuse learning processes (e.g., doing an internship) with learning outcomes (e.g., application of theory to real-world problems).	The list includes reasonable outcomes but does not specify expectations for the program as a whole. Relevant institution-wide learning outcomes and/or national disciplinary standards may be ignored. Distinctions between expectations for undergraduate and graduate programs may be unclear.	The list is a well-organized set of reasonable outcomes that focus on the key knowledge, skills, and values students learn in the program. It includes relevant institution-wide outcomes (e.g., communication or critical thinking skills). Outcomes are appropriate for the level (undergraduate vs. graduate); national disciplinary standards have been considered.	The list is reasonable, appropriate, and comprehensive, with clear distinctions between undergraduate and graduate expectations, if applicable. National disciplinary standards have been considered. Faculty have agreed on explicit criteria for assessing students' level of mastery of each outcome.
Assessable Outcomes	Outcome statements do not identify what students can do to demonstrate learning. Statements such as "Students understand scientific method" do not specify how understanding can be demonstrated and assessed.	Most of the outcomes indicate how students can demonstrate their learning.	Each outcome describes how students can demonstrate learning, e.g., "Graduates can write reports in APA style" or "Graduates can make original contributions to biological knowledge."	Outcomes describe how students can demonstrate their learning. Faculty have agreed on explicit criteria statements, such as rubrics, and have identified examples of student performance at varying levels for each outcome.
Alignment	There is no clear relationship between the outcomes and the curriculum that students experience.	Students appear to be given reasonable opportunities to develop the outcomes in the required curriculum.	The curriculum is designed to provide opportunities for students to learn and to develop increasing sophistication with respect to each outcome. This design may be summarized in a curriculum map.	Pedagogy, grading, the curriculum, relevant student support services, and co-curriculum are explicitly and intentionally aligned with each outcome. Curriculum map indicates increasing levels of proficiency.
Assessment Planning	There is no formal plan for assessing each outcome.	The program relies on short-term planning, such as selecting which outcome(s) to assess in the current year.	The program has a reasonable, multi-year assessment plan that identifies when each outcome will be assessed. The plan may explicitly include analysis and implementation of improvements.	The program has a fully-articulated, sustainable, multi-year assessment plan that describes when and how each outcome will be assessed and how improvements based on findings will be implemented. The plan is routinely examined and revised, as needed.
The Student Experience	Students know little or nothing about the overall outcomes of the program. Communication of outcomes to students, e.g. in syllabi or catalog, is spotty or nonexistent.	Students have some knowledge of program outcomes. Communication is occasional and informal, left to individual faculty or advisors.	Students have a good grasp of program outcomes. They may use them to guide their own learning. Outcomes are included in most syllabi and are readily available in the catalog, on the web page, and elsewhere.	Students are well-acquainted with program outcomes and may participate in creation and use of rubrics. They are skilled at self-assessing in relation to the outcomes and levels of performance. Program policy calls for inclusion of outcomes in all course syllabi, and they are readily available in other program documents.