Program Outcomes and Assessment

Degree Program: Bachelor of Computer Science (BS-CSCI)

Contact Person: Byron Jeff

Program Mission Statement: The IT Program (BIT, BSIT, CSCI) is committed to positively impact society (locally, nationally, and internationally) through academic, applied, and educational research to build, expand, disseminate and teach the information technology body of knowledge.

CSCI Learning Outcomes

Outcome 1: Be able to solve complex and significant problems with professional skill, by formulating efficient and effective algorithmic solutions to a wide variety of sophisticated problems normally encountered in real-world computing and in academe.

Outcome 2: Be able to express algorithms clearly and correctly in a variety of programming languages.

Outcome 3: Be able to demonstrate sufficient foundational knowledge of the following main subject areas of computer science: operating systems, databases, networks, graphics, software engineering, game design and implementation, and web technologies.

Outcome 4: Have an understanding of ethical responsibilities related to the areas of computing and the workplace.

Outcome 5: Have mastery of the theoretical underpinnings of Computer Science.

Outcome 6: Apply breadth of computer science knowledge to either post-baccalaureate study or a career field.

Assessment Methods/Type of Evidence

The IT Department at Clayton State used Bloom's taxonomy of the cognitive domain (Bloom 1956) to develop a scheme for mapping program outcomes to courses. Three levels of mastery have been defined based on Bloom: developing level of mastery, mature level of mastery, and proficient level of mastery. Correspondingly, assessment rubrics for each assignment element rely on these three levels of mastery (Booth 2006). See Table 1.

Table 1. Levels of Mastery

Level	Definition
Developing	Demonstrates an emerging level of knowledge and skills; can perform beginning skills and shows potential to perform independently.
Mature	Demonstrates a refined level of comprehension; is able to apply appropriate skills and perform both independently and as a team member
Proficient	Demonstrates a superior level of knowledge and understanding; integrates and applies skills across multiple areas both independently and as a team member.

One or more of the program outcomes is mapped to each course. The mapping indicates whether the outcome will be assessed at the developing level, mature level, or proficient level (see CSCI Matrix). One or more individual course objectives are mapped to each program outcome. The mapping of course objectives to program outcomes is recorded in public syllabi for each course.

The new Computer Science degree was implemented in the fall semester 2009. Assessment data will be collected for all Computer Science Courses.

Instructors develop and administer assessments, collecting data for each course they teach. A given assessment may address many objectives and an objective may be addressed by many assessments. To effectively map this many-to-many relationship, assessments may be viewed as having one or more elements. Each element is designed to measure one or more course

objectives. A given course objective might be assessed by more than one element. Course objectives support specific program outcomes and should be measured against a specific mastery level (Booth, et al. 2009).

Assessment data is collected for each program outcome in each course where that outcome is addressed (see CSCI Matrix). A variety of assessment tools are used: Quizzes, tests, assignments, presentations, projects, portfolios, and internships. Data is summarized for each course. Because each program outcome is addressed by many courses, data is summarized longitudinally for each program outcome across all relevant courses.

References

Bloom, B. S. (1956). *Taxonomy of Educational Objectives Handbook 1: Cognitive domain*. New York: Longman, Green & Company.

Booth, L. (2006). A database to promote continuous program improvement. *Proceedings of the 7th Conference on Information Technology Education SIGITE '06*. 83-88.

Booth, L., Booth, V., Hartfield, F. (June, 2009). Continuous course improvement, enhancements and modifications: Control and tracking. *Online Journal Distance Learning Administration*.

Course Mapping Matrix

B.S. Computer Science - Program Outcomes Inventory						
Instructions: Each course should address one or more of the program outcomes. Indicate by placing an x in the appropriate column.	1) Solve complex and significant problems with professional skill by formulating efficient and effective algorithmic solutions .	2) Express algorithms clearly and correctly in a variety of programming languages.	3) Demonstrate sufficient foundational knowledge of computer science: OS, DB, Netks, Graphics, SE, Gaming, Web.	4) Understand ethical responsibilities related to the areas of computing and the workplace	5) Demonstrate mastery of the theoretical underpinnings of computer science.	6) Apply breadth of computer science knowledge to either post-bac study or a career field.
CSCI 1100 -- Applied Computing	D	D	D			D
CSCI 1301 - Computer Science I	D	D	D		D	
CSCI 1371 - Computer Science for Engineers	D	D	D		D	
CSCI 1302 - Computer Science II	M	M	D		D	
CSCI 2302 - Data Structures and Algorithms	M	M			M	
CSCI 2305- Computer Organization and Architecture	D		M		M	
CSCI 3300 - Computer Ethics				M		
CSCI 3301 - Game Design & Programming I	M	M	M			
CSCI 3305 - Operating Systems	M	M	P		P	
CSCI 3306 – Computer Networks and Security I	M		P	M	P	
CSCI 3310 – Databases Design & Implementation	M		P	M	P	
CSCI 3320 - Software Engineering Design	P	M	P	M	P	M
CSCI 3333 - Programming Languages	M	M	M		M	
CSCI 4301 - Game Design & Programming II	P	P	P			P
CSCI 4304 – Computer Graphics	P	P	P		M	
CSCI 4305 – Unix (Linux) Systems Programming & Administration	P	M	M			
CSCI 4306 – Computer Networks and Security II			P			
CSCI 4307 - Artificial Intelligence	P	P	M		P	
CSCI 4310 – Advanced Issues in Databases	P		P			
CSCI 4315 - Human Computer Interface	P	M	M	M		
CSCI 4316 – Cluster and Grid Computing	P	P	P			
CSCI 4320 - Software Engineering Practicum	P	P	P	P	P	P
CSCI 4333 - Theory of Computation	P		M		P	
CSCI 4334 - Algorithm Design and Analysis	P		M		P	
CSCI 4314 Multimedia Production and Development			M			
CSCI 4360 – Computer Science Research	P		P		M	P
CSCI 4370 - Internship in Computer Science						P
CSCI 4800 - Special Topics in Computer Science	P					

Discussion of Results and Changes - CSCI

The Computer Science degree was implemented in the fall of 2009. At the time of this report (Fall 2009), data has not yet been collected. No changes have been made.

Assessment Plan

One or more of the program outcomes is mapped to each course. The mapping indicates whether the outcome will be assessed at a developing level, mature level, or proficient level (see CSCI Matrix). One or more individual course objectives are mapped to each program outcome. The mapping of course objectives to program outcomes is recorded in a public syllabus for each course. Instructors use the public syllabus to construct a section syllabus for each course they teach.

Every time a course is taught, data will be collected . Assessment data is collected for each program outcome in each course where that outcome is addressed. A variety of assessment tools are used: Quizzes, tests, assignments, presentations, projects, portfolios, and internships. Data is summarized for each course. Because each program outcome is addressed by many courses, data is summarized longitudinally for each program outcome across all relevant courses.

Responsibility

To implement continuous program improvement, analysis of data must lead to planned curriculum revision. CIMS has developed clearly defined levels of responsiblilty for change management.

Work at the College Level:

Develop college outcomes in line with university outcomes.

Establish goals, objectives and guidelines – an overall plan for achieving college outcomes.

Develop a feedback loop that takes into account departmental performance, faculty feedback, and departmental evaluation of program effectiveness. Revise college outcomes, goals, objectives, and guidelines as necessary. Document changes so that the college can verify continuous improvement.

Work at the Program Level:

Develop program outcomes in line with college outcomes.

Map program outcomes to courses. Establish acceptable performance criteria.

For each course, develop core course objectives that support each program outcome mapped to the course. This is a departmental level, top down design, exercise because courses and their prerequisites flow together to create the curriculum as a whole. Courses do not exist in isolation.

Develop a feedback loop that takes into account course performance data, student feedback, and faculty evaluations of courses. Revise program outcomes, course outcomes, and acceptable performance criteria as necessary. Document changes so that the program can verify continuous improvement.

Work at the Faculty Level:

Develop additional course objectives. This secondary set of course objectives encourages bottom-up evolution of the curriculum. For both core course objectives and secondary course objectives, develop instructional components designed to teach course objectives.

Develop assessments and corresponding rubrics for each course objective.

Create a spreadsheet for recording student scores based on assessments and rubrics. Note: be as discrete as possible. For example, if a test covers two or more objectives, the spreadsheet elements for recording the test should have a column for each objective.

A separate page of the grading spreadsheet should contain ‘roll-up’ formulas that summarize overall student performance that can be compared to established performance criteria. While each faculty member may have individual and creative instructional components, assessments, and rubrics, the ‘roll-up’ should be standardized so that program level summaries of course objectives and program outcomes are easy to achieve.

Develop a feedback loop that takes into account student performance, student feedback, and peer evaluations of teaching effectiveness. Revise instructional components, assessments, and rubrics as necessary. Changes should be documented so that faculty can verify continuous course improvement. Proposed changes to program outcomes and/or core course objectives should be submitted to the program curriculum committee for consideration by the faculty as a whole.