



Computer Science | 2013-2014 Assessment Report

In the past Computer Science has used a variety of assessment methods including many indirect methods. This year the program decided to concentrate on direct methods by using embedded assessment measurable by grades on various projects, exam questions, homework assignments, and whole courses. The results generally show a satisfactory level of learning by students, but there are occasional problems with writing skills and there are several instances of bimodal grade distributions which are interpreted to mean that some students give up (high absence rates), do not expect the level of rigor required, or are not adequately prepared by their previous courses. The curriculum has been redesigned this year to correct some of these problems and to clarify expectations of individual courses.

Technique: embedded assessment by grades of projects, exam questions, and courses

Computer Science Learning Outcomes:

1. Students must be able to design the logic and information structures necessary to create software capable of solving problems subject to specified constraints.
2. Students must develop both written and verbal communications skills that support the design and documentation of software products and help utilities.
3. Be able to analyze software to determine correctness and, if incorrect, be able to determine the cause of errors and fix them.
4. Students must understand fundamental principles and theory of both computer hardware and software and the mathematical foundations on which Computer Science is built.

Fall 2013

Learning outcome to assess: #1 Students must be able to design the logic and information structures necessary to create software capable of solving problems subject to specified constraints.

CS 201: (course grades) introductory course

Data: A:15; AB:5; B:3; BC:2; C:1; CD:1;F:2

CS 256: (project grades) students were given a detailed specification and asked to design/implement their solution. In the case of CS256 these were individual assignments.

Data: grade distribution: A: 4; AB: 6; B: 8; BC: 2; C: 3; CD: 0; F: 2

Comments: This was a successful project for most the students. The two who had trouble were students that very rarely showed up for class. One of them is in the CS256 class again this semester and doing better.

CS 351: (course grades) theory course with heavy mathematics

Data: A:5; AB:3; B:2; BC:3; C:5; CD:2; F:1

Comments: reflects common bi-modal distribution

CS 371: (project grades) this consisted of a two person project creating a text based adventure game. Students were given detailed constraints in the form of a 10 page software specification, they then had to design and implement their solution.

Data: A: 8; AB: 4; B: 6; BC: 5; C: 4; CD: 0; D: 0; F: 0

Comments: Students generally did very well on the project. If we consider that a B grade is an above-average grade for a computer science course, then 67% of our students achieved this. Meanwhile the students who did not are still performing satisfactory in terms of being able to progress further in the CS curriculum (a C grade is needed for the next courses).

CS 450: (exam question grades) Three algorithm design questions on the final exam evaluated and summarized.

Data: About $\frac{3}{4}$ of the students were able to design algorithmic structures to solve various problems. The performance was NOT strong, except in the case of one or two students.

Comment: Suggestion to restructure the course to concentrate on algorithm design with less emphasis on efficiency analysis.

CS 460: (course grades) a work-intensive course not theoretically demanding

Data: A:12; AB:1; B:1; F:1

Learning outcome to assess: #2 Students must develop both written and verbal communications skills that support the design and documentation of software products and help utilities.

CS 450: (exam questions) Three algorithm design questions on the final exam evaluated and summarized.

Data: The writing on the three selected problems was poor. Students were largely unable to express with any precision their algorithmic approach to solving the specified problems. I found them very hard to read and follow. Again, there were notable exceptions.

Comment: Keeping in mind that the writing was for a final exam, I would hope that given adequate time for revision and review, these students could articulate their thoughts more clearly. Perhaps assignments, in various courses, intended to improve students' technical writing skills would be helpful. Learning outcome to assess: #3 Students must be able to analyze software to determine correctness and, if incorrect, be able to determine the cause of errors and fix them.

CS 201: (course grades) introductory course

Data: A:15; AB:5; B:3; BC:2; C:1; CD:1;F:2

Spring 2014

Learning outcome to assess: #1 Students must be able to design the logic and information structures necessary to create software capable of solving problems subject to specified constraints.

CS 201: (course grades) introductory course

Data: A:14; AB:5; B:3; BC:1; C:1; CD:1;F:0

CS 241: (exam question grades) assessment embedded in exam and homework questions on representing geographic data as an adjacency graph and representing digraphs as binary adjacency matrices.

Data on adjacency graph: 99% accurate on one problem; 78% accurate on another

Data on matrices: 27 of 28 students correct

Data on exponential manipulation: 54% mean accuracy on exam (bimodal distribution)

CS 242: (exam question grades) assessment embedded in exam and homework questions on representing geographic data as an adjacency graph and representing digraphs as binary adjacency matrices.

Data on adjacency graph: 87% accuracy

Data on matrices: 13 out of 15 students correct

CS 352: (course grades) theoretical course with heavy mathematics and programming

Data: A:9; AB:2; B:0; BC:2; C:0; CD:0;F:7

CS 372: (project grades) embedded assessment of a design project

Data: A:5; AB:5; B:10; BC:0; C:5; D:0

Learning outcome to assess: #2 Students must be able to analyze software to determine correctness and, if incorrect, be able to determine the cause of errors and fix them.

CS 201: (course grades) introductory course

Data: A:14; AB:5; B:3; BC:1; C:1; CD:1;F:0