College of Letters and Science
Computer Sciences Department
Assessment Report, 1997-98
May 1, 1998

Abstract

The Computer Sciences (CS) Department Curriculum Committee has conducted an extensive review of its introductory programming and data structures courses, involving instructors of those courses (lecturers, TA's, and a faculty associate) and faculty who are involved primarily in teaching upper-level CS courses. The review was based also on input from additional faculty in CS and Engineering, surveys of introductory CS courses nationally, surveys of employers of CS majors, and statistics on enrollment patterns in CS courses.

Based on the findings of this review, in the year 1998-99 we will (i) realign course materials in our introductory programming course to emphasize important course objectives, namely development of good programming and problem-solving skills, (ii) change the format of the course to meet pressing staffing constraints, and (iii) improve training and use of TA lab consultants and graders in order to provide better and more timely feedback to students on their work. We propose to assess the effectiveness of these changes in our introductory courses via student surveys and testing in the upcoming year.

At the graduate level, we have compiled information from exit interview surveys of CS graduate students since 1995. These interviews have provided us with useful information on the success rate of students entering our graduate program, time to completion of degree, and the success rate of fellowship students who enter the department.

1 Assessment Tools Used (or Planned)

1.1 Direct Indicators

	Undergraduate Major	Graduate Program
National Exams		
Local Exams		
Capstone Course(s)		
Embedded Testing		
Student Portfolios		
Review theses & dissertations		
Performance Evaluations		
Pre and Post Testing	Spring 1999	

1.2 Indirect Indicators

	Undergraduate Major	Graduate Program
Student Surveys	Fall 1998,Spring 1999	
Exit Interviews		Ongoing
Alumni Surveys		
Employer Surveys	Spring 1998	
External Reviews		

Other indirect indicators that we have used are (i) comparison of course content with that of other schools nationally, (ii) faculty and instructor surveys, and (iii) statistics on student enrollment patterns.

2 Narrative

2.1 Undergraduate-level Assessment Activities

Our assessment activities at the undergraduate level were done in the context of an in-depth review of our introductory programming (CS302) and data structures (CS367) sequence. This review was based on

- frequent discussions of the CS Curriculum Committee (including lecturers, faculty, and graduate student instructors),
- a survey of CS and Engineering faculty and CS instructors on appropriate programming language for our introductory courses,
- a survey of introductory CS courses in other schools nationally, covering content, class assignments, class format, staffing, use of major versus non-major tracks, class prerequisites, and instructors' views of the effectiveness of the course content and structure,
- a survey of employers' views on desired content of CS introductory programming courses, and
- statistics on student enrollment patterns in CS302.

Based on this review, we are developing a detailed document on CS302, covering course objectives, course content, and expected student achievement (this will be completed in mid-May). In addition, we have put in place for Fall 1998 several changes that are intended to improve the quality of instruction in CS302 and CS367:

- CS302 content will emphasize important course objectives, including programming and problem-solving skills. In January 1998, our department approved a switch to the Java programming language, starting in Fall 1998. Java is simpler than the currently used language (C++) and has features that simplify the task of finding errors in programs. We also plan to introduce students earlier in the course to object-oriented programming methodologies.
- Currently, CS302 graders and lab consultants receive little training and have limited contact with the instructors of the course. Lab consultants are often underutilized. We have plans for improved training of TAs and consultants in order to provide better and more timely feedback to students on their work. We will provide instruction in Java to all TAs for CS302 in Fall 1998.
- We have changed the course descriptions and prerequisites for both courses, in order to facilitate better integration of this two-course sequence.
- We have changed staffing of CS302 and class format in Fall 1998. The course is currently taught in small sections by graduate student TA's. However there is a shortage of qualified instructors from the TA pool and an extremely high turn-over of instructors. A faculty associate (Skrentny) will teach two large sections of the course in Fall 1998 (with 150 students per section; there will be two 75-minute large lectures and per week, with an additional small-group discussion/lab session for students once per week) and the remaining students (expected 300 or so) will be taught in the traditional small-section format (25 students per section). Skrentny will also work on development of lab sections and provide training for CS302 TAs.

- Prior to our review, there was no mechanism for dialogue between (non-tenure track) instructors who teach and largely determine the content and structure of CS 302 and some sections of CS367, and the tenure-track faculty who teach upper-level courses. We now understand the importance of this dialogue and plan to continue it (partly facilitated by our proposed assessment activities).

- We are developing a new 1-credit C++ course to prepare CS majors for upper-level courses that use this language and for jobs. This course, which is expected to start in Spring 1999, will be offered in both lecture-based and self-guided formats.

The next steps are to assess whether these changes will lead to desired student achievement and to determine how we can do even better. For the upcoming year, we are proposing assessment activities designed to gauge (i) student achievement in the introductory sequence, (ii) student preparedness (both upon entry to our introductory courses and also preparedness for upper-level CS courses upon completion of the introductory courses), and (iii) effectiveness of staff resource allocation. A detailed proposal and request for funding is attached.

2.2 Graduate-level Assessment Activities

We have compiled data from exit interview surveys of CS graduate students. Although these surveys have been completed by exiting graduate students for several years, this is the first time that we are summarizing the data from these interviews in a format accessible to all faculty.

Over the years 1995-1997, 124 CS graduate students completed the exit interview survey. 25 of these were PhD students, 95 were MS students and only 4 left without completing a degree. (Of these four, just one transferred to another university.)

Of the 25 PhD students who completed the survey, 7 completed their degree in 5 years, 9 in 6 years and 9 in seven or more years. We would like to reduce the mean time to completion of degree for PhD students.

Of the 95 MS students who completed the survey, most completed their degree within two years, with 13 taking three years, 4 taking four years and 3 taking five or more years. This indicates that MS students are for the most part completing their degree in a timely manner. 48 out of the 95 students entered with plans to obtain a PhD degree but switched to a MS degree after entering the program. 34 of the MS students were supported as an RA for some period during their graduate education.

14 of the students surveyed obtained prestigious university fellowships upon entering the graduate program. It is interesting to note that of these, only 7 obtained the PhD degree; 6 left the program with a MS degree with one leaving without a degree.

Some questions on the survey that were intended to get information about why students chose our program and what research areas students chose were poorly designed, and as a result it was hard to get useful information from these questions. In the summer of 1998 we will revise the exit interview survey and put it on-line, to facilitate compilation of data and to make it easier for students to complete the survey.

Proposal for Assessment, June 1998 - August 2000
Computer Sciences Department
May 1, 1998

Overview

We propose a two-year plan to assess the introductory Computer Science curriculum; in particular, the two-course sequence CS302 (Introduction to Computer Programming) and CS367 (Data Structures). We are requesting $15,000 from the Assessment Council to support our assessment activities in the first year (no funds would be needed for the second year). This proposal contains the following sections:

1. **Goals:** The high-level goals of the Computer Sciences Department that motivate this proposal.
2. **Proposed assessment activities:** What we need to do to achieve our goals.
3. **Measuring success:** How will we determine whether changes instituted in response to our assessment are effective?
4. **LEAD Center:** Why we need the help of the LEAD Center.
5. **Future Assessment:** How the materials that we develop during the next year will be used as part of an on-going assessment plan, and how they will be used to prepare a plan for the assessment of our higher-level courses.
6. **Time line:** Proposed activities from summer 98 through summer 2000.
7. **Benefits to other departments:** How other departments will benefit from our work.
8. **Budget**
9. **Assessment participants**

1 Goals

Our high-level goals are:

- To design introductory programming courses (variants of the current CS302) so that students with a wide range of abilities and previous experience will be able to learn introductory programming.
- To increase the number of the best students (especially women) who go on to take higher-level Computer Science courses after completing the CS302/CS367 sequence.
- To ensure that those students who do go on to higher-level Computer Science courses are adequately prepared for those courses.

2 Proposed Assessment Activities

In order to achieve our high-level goals, we need to determine:

- what factors predict success for students in CS302,
- how to restructure CS302 so that more students succeed,
- what factors influence the best students' (especially the women) decisions whether to continue on in Computer Science,
- how well CS302 and CS367 prepare students for higher-level Computer Science courses, and
- how to make the most efficient use of our limited resources in running our introductory courses (CS302 and CS367).

We plan to work with Diane Bowcock at the LEAD center to develop good assessment instruments to be used to answer these questions. Some preliminary ideas are presented below. Ongoing research at Carnegie Mellon University, conducted by Professor Allan Fisher (Associate Dean for Undergraduate Education) and Dr. Jane Margolis (social scientist), will also be useful to us as we develop our surveys: Fisher and Margolis are investigating the experiences of undergraduate Computer Science students at Carnegie Mellon, primarily women, with the goals of attracting and retaining women students in Computer Science.

An important aspect of our proposal is that we intend to make the majority of the materials that we develop web-based. This will facilitate the collection of data, and will also provide templates that can be reused both by our own department and by other departments in future assessment activities.

2.1 Predicting Success in CS302

We expect that several factors can be used to predict how likely a student is to succeed in CS302 as it is currently taught:

- amount and quality of previous programming experience (e.g., high-school programming courses, Advanced Placement Computer Science courses, using computers in other courses, working on a home computer);
- number of high-school math courses, and grades in those courses;
- scores on math exams (given to all incoming freshmen).

Instructors of the course have pointed out that other factors, such as students allocation of time for the course (it takes more time to do well in CS302 than in other 3-credit 330-level courses), also play an important factor in students' success.

We plan to gather data in these three areas for the students who register for CS302 in the fall of 1998, as well as data on their achievements in that class. We will then look for correlations between the three areas, and success in CS302.

CS302 will be taught using two different formats in the fall (about 300 students will be taught in two large lecture sections – taught by an instructor – with additional small TA-taught recitation sections, while another 300 students will be taught in small TA-taught lecture sections). This will give us a unique opportunity to take class format into account as a possible factor that predicts success or failure for certain groups of students.

2.2 Restructuring CS302

Based on the results of the "predicting success" study, we will consider how best to structure CS302 so as to maximize the number of students who successfully meet the desired learning goals for that course. In particular, we will consider:

- large versus small lectures,
- separate sections based on amount of previous programming experience, and
- separate sections for students planning to major in Computer Science and/or Engineering.

2.3 Why students continue in CS

We expect that students' experiences in CS302 and CS367 are important factors in their decisions as to whether to take more Computer Science courses. We will test this hypothesis by surveying students at the ends of these course to determine:

- whether they plan to continue in Computer Science,
- why they have decided to continue or not to continue, and
- what kind of experience they had in CS302 or CS367.

Using data from these surveys as well as their grades from the assignments in CS302 and CS367, we will construct profiles of students who do and do not tend to continue in Computer Science after taking those courses. We will use this information to modify CS302 and CS367 to encourage more top students to continue in Computer Science.

2.4 Preparedness of Students for Higher-Level CS Courses

To determine how well students are prepared for higher-level Computer Science courses after finishing the CS302/CS367 sequence, we will give students exams (at the end of CS367 and/or at the beginning of their first higher-level course) that cover the material that the instructors of the higher-level courses feel is vital to incoming students. We will also survey instructors of higher-level courses to determine what skills they think their incoming students lack. Finally, we will survey students when they exit their first higher-level course to determine whether they feel they were well prepared for that course, and if not, what skills were lacking.

We will use this information to decide whether revisions need to be made to the CS302 and CS367 curricula.

2.5 Efficient use of Resources

This activity involves determining how well human resources like undergraduate consultants, graduate consultants, graders, and lecturers are being used in teaching CS302 and CS367. Are consultants familiar enough with the material being covered in class and with the current homework assignments to provide effective help? Are assignments graded promptly enough and with enough feedback to help students learn from their mistakes? We will answer these questions by gathering numerical data for things like grading times, and by surveying students to learn what they think are the most and least effective aspects of the way human resources are currently used.

We will use the results to adjust our use of human resources appropriately.

3 Measuring Success

It is vital to be able to measure the efficacy of the actions we take in response to the results of our assessment activities. We will measure success in achieving our goals as follows:

Goal: Students with a wide range of abilities should be able to succeed in CS302.

Measurements of success:

- Significant number of students with profiles that predicted failure in CS302 as originally formatted succeeding under the new format.
- Lowered percentage of students who drop CS302.

Goal: Increase number of top students continuing in Computer Science.

Measurements of success:

- Higher fraction of students in the top 20% of their CS302 and CS367 classes who choose to take more CS courses.

Goal: Ensure that CS302 and CS367 prepare students for higher-level CS courses.

Measurement of success:

- Better performance on exams given at the end of CS367 and/or at the beginning of a student's first higher-level CS course to determine their mastery of vital material. Higher level of satisfaction expressed by instructors and students in higher-level courses (in the area of student preparedness).

4 Working with the LEAD Center

We believe that working with the LEAD Center is vital to the success of our proposed assessment. While we have some ideas (outlined above) for how to proceed, we are sure that the experts at the Center will be able to suggest other assessment activities that will better help us to determine what changes need to be made to achieve our goals, and to measure the success of those changes. We will also rely on their help in developing surveys initially, and on revising questions based on one round of surveying (see "time line" below).

Diane Bowcock at the LEAD center has already worked with Botany and nine departments in Human Ecology. She will be the primary person from LEAD who will work with us on the design of our proposed assessment activities. She has already lead retreats and workshops with faculty in Human Ecology on assessment and can help educate our faculty on good assessment practices.

Additionally, our department plans to participate in a proposed pilot project, lead by Bowcock, in which LEAD will train a pool of graduate students in assessment activities in the fall of 1998. As part of their training, those graduate students would interview students taking our introductory courses, and would provide qualitative data that will help us revise some of our survey materials. (It is not clear yet if this pilot project will be approved but if so, CS will participate.)

5 Future Assessment

Some of the materials created for this project will be useful in on-going assessments (possibly with minor updates over the years). For example:

- the surveys used to determine which students choose to continue in Computer Science after completing our introductory sequence;
- the exams that measure how well prepared students are for higher-level Computer Science courses after completing the introductory sequence;
- the surveys that measure instructor and student perceptions of students' preparedness for higher-level Computer Science courses after completing the introductory sequence;
- the surveys that determine the effectiveness of the human resources used in the introductory courses.

In addition, as mentioned above, many of these materials will be web-based; thus, they will provide templates that will greatly simplify the creation of similar, web-based materials for use both by the Computer Sciences Department as well as by other departments.

Finally, this will be our department's first major venture into assessment. We expect to learn a lot about the process and the benefits and limitations of such activity. We plan to use our newly acquired knowledge as the basis for an assessment plan for our upper-level undergraduate and graduate programs.

6 Time Line

Summer 98: Work with the LEAD Center to identify appropriate assessment mechanisms and to develop materials (e.g., surveys).

Fall 98: Gather data.

Spring 99: Evaluate data gathered to-date; revise assessment materials; gather more data.

Summer 99: Evaluate data; plan changes based on assessment results.

Fall 99 and Spring 2000: Gather data to be used to measure the success of the changes made to CS302 and CS367 in achieving our goals.

Summer 2000: Evaluate success; plan further changes as needed.

7 Benefits to Other Departments

As noted above, a major benefit will be the development of web-based materials that will be made available as templates for use by other departments. In addition, some of our assessment results should be of interest in other areas. For example, the trade-offs between large lecture courses and small TA-taught sections will be relevant to any department with large introductory courses.

8 Budget

We are requesting $15,000 from the Assessment Council to support the activities outlined in this proposal.

CS student hourly to prepare web-based surveys	$ 4,000
Technical assistance from LEAD staff in developing and revising assessment instruments	$ 11,000
Total:	$ 15,000

Supplies and computer expenses will be covered by the Computer Sciences Department. The CS Department also plans to commit $2,000 towards the proposed LEAD pilot graduate student training project described above in section 4.

9 Assessment Participants

A committee of four CS faculty (Anne Condon, Charles Dyer, Susan Horwitz, and Deborah Joseph) and one CS faculty associate (Jim Skrentny) will work on this assessment plan over the proposed two-year period.

Condon, Horwitz, and Skrentny have been members of the CS Course and Curriculum Development Committee during the past year (with Condon serving as committee chair). The major task of that committee has been to perform initial evaluation and revision of CS302 and CS367 (as described in the CS assessment report). Jim Skrentny has been responsible for overseeing CS302 in recent semesters and will teach two large-lecture sections in Fall 1998.

Diane Bowcock will be the LEAD person assigned to this project. LEAD's services will be needed during the 1998-99 academic year, including both summers. Once the survey materials have been designed and subject to one revision, CS will be in a position to use, maintain and revise them during the second year of the proposal and periodically in the future.

A CS student will be identified to work on web-based implementation of surveys. Once templates for the surveys and for presentation of data are developed, along with documentation, it should be possible to revise our surveys and reuse the templates for new surveys with little or no further technical help. For this reason, we expect that the web-based implementation work will be completed by the end of 1998.