A Placement Test for Computer Science: Design, Implementation, and Analysis

Gwen Nugent[1], Leen-Kiat Soh[2], Ashok Samal[2], and Jeff Lang[1]

[1]National Center of Information Technology in Education
University of Nebraska
269 Mabel Lee Hall, Lincoln, NE 68588-0230 USA
[2]Department of Computer Science and Engineering
University of Nebraska
256 Avery Hall, Lincoln, NE 68588-0115 USA

Correspondence: Gwen C. Nugent, 216 Mabel Lee Hall, University of Nebraska-Lincoln, Lincoln, Nebraska 68588-0235, gnugent@unl.edu

Abstract

An introductory CS1 course presents problems for educators and students due to students' diverse background in programming knowledge and exposure. Students who enroll in CS1 also have different expectations and motivations. Prompted by the curricular guidelines for undergraduate programs in computer science released in 2001 by the ACM/IEEE, and driven by a departmental project to reinvent the undergraduate computer science and computer engineering curricula at the University of Nebraska-Lincoln, we are currently implementing a series of changes which will improve our introductory courses. One key component of our project is an online placement examination tied to the cognitive domain that assesses student knowledge and intellectual skills. Our placement test is also integrated into a comprehensive educational research design containing a pre- and post-test framework for assessing student learning and providing valuable feedback for needed instructional revisions. In this paper, we focus on the design and implementation of our placement exam and present an analysis of the data collected to date.

A Placement Test for Computer Science
Design, Implementation, and Analysis

Introduction

Student preparation for the undergraduate study of computer science has been and will likely continue to be extremely varied. Not only is the level of preparation diverse, but also the form and content are inconsistent. High school computer science is generally taught with a mathematics, science, or business emphasis and may or may not be associated with an advanced placement (AP) program. It may cover programming, networking, or web page development, and may utilize functional, object-oriented, or imperative programming paradigms supported by an assortment of programming languages. This observation applies not only to the rural Great Plains states, but also to most other universities that teach computer science. A placement examination that correctly and reliably assigns students with diverse backgrounds to the most appropriate first course in computer science would have widespread applicability.

Our research group, a collaboration between faculty from the Department of Computer Science and Engineering (CSE) and the College of Education and Human Sciences (CEHS) at the University of Nebraska-Lincoln (UNL), has taken on the challenge of creating a placement test as one component of a broader "reinvention" of the introductory computing curriculum in CSE. In addition to assigning students to their first course in computer science, the test is designed to serve as a dependent variable in a pre-post research design, validate the instructional impact of CS1, and serve as a knowledge measure in other research designs to determine ways to maximize student success and retention. This paper documents the processes our research group has followed in planning, creating, implementing, and analyzing a placement test, and describes the group's anticipated future work. We detail the design and implementation of the test itself, including initial work, building the question pool, categorization of questions by topic and level of knowledge, integration with a delivery system, and testing of the entire package. We discuss the results and analyses based on student performance, and we conclude with our plans for future work. While the focus on the paper is the design, analysis and implementation of a placement test for a CS1 course, the techniques described here can be broadly applicable to testing and evaluation in classes in general.

Background

The primary purpose of the placement exam is to appropriately place students into one of two introductory computer science courses offered by our CSE department. The first option, CSE105 (pre-CS1), is an introductory course intended for students who lack prior exposure to logic constructs and fundamental computer science terminology, and students who are not CSE majors but who want to gain a basic understanding of the field of computer science including programming. Credit earned in CSE105 does not count towards a computer science or computer engineering degree. The second option, CSE155 (CS1), is intended for students who have a basic understanding of computing concepts including basic control-flow structures and Boolean logic and who have programming experience in a high-level programming language. These students typically plan to major in an area related to computer science and engineering. The needs of each group of students are different; therefore our placement test is designed to distinguish between the groups and place students into the appropriate course. In the past, students entered the same introductory course regardless of their major field of study or prior computing experience. This left some students feeling overwhelmed and others complaining of boredom. Having two separate introductory CS courses allows our CSE Department to serve students who have diverse backgrounds and different goals.

The placement test is based on the recommendations of the ACM/IEEE Computing Curricula 2001 (ACM/IEEE-CS, 2001) document. The document details the topics leaders in the field of computer science deem to be most critical. The CSE Department at the University of Nebraska-Lincoln has used the ACM2001 guidelines to design their introductory CSE courses and to determine the topics to be covered on the placement exam. Each topic included in the placement exam is also addressed in both introductory courses (pre-CS1 and CS1). Table 1 lists topics and subtopics we are using. The first five groups of topics are used as the placement criteria, and the second five groups are used in the pre- and post-test analysis for CSE155 (CS1).

Insert Table 1 about here

The exam covers each of the major topics recommended by the Computing Curricula 2001 guidelines and tests the students' knowledge of each topic at multiple levels. Testing at multiple levels is accomplished by developing our question pool based on the cognitive model of Bloom's Taxonomy (Bloom, l956). The cognitive domain includes the recall or recognition of specific facts, procedural patterns, and concepts that serve in the development of intellectual abilities and skills. There are six major categories in this model, as shown in Table 2, starting from the simplest behavior, knowledge, to the most complex, evaluation. These categories can also be viewed as a way to measure learning from the most basic to the most complex, requiring the mastery of one level before the next level of learning can occur. Table 2 also provides examples of questions at each level.

Insert Table 2 about here

Since the placement exam covers topics discussed in CSE155, the exam is also used as a pretest-posttest measure. At the end of the semester students retake the exam, and we are able to measure gains at both the individual student level and at the topic level for the class as a whole. This gives us some insight into areas where students are performing poorly and where instructional interventions or revisions are needed.

The placement exam is more than a random sample of CS-related questions. Through careful analysis we have developed a tool which provides an objective measure to reliably place students into the appropriate introductory course, measure student's computer science background knowledge, understand the depth of knowledge students possess, evaluate both students and instruction, and measure outcome learning.

Related Work

Using a placement exam to assign students to a CSE course is not a novel idea. At the Department of Computer Science at Northwestern University, the CSE placement exam is designed to evaluate students' programming skills (http://www.cs.northwestern.edu/academics/undergrad/placement/). There are three possible outcomes of this exam: 1) students are allowed to enter any course that requires pre-CS1 or C/C++/Java programming experience, 2) students are allowed to enter any course that requires "basic programming experience" but not courses requiring pre-CS1 or C/C++/Java experience, or 3) students must take pre-CS1 or other introductory CSE courses that do not require programming experience. The pre-CS1 course does not count towards the CSE credit hours. The exam includes open-ended questions and programming questions.

At California Institute of Technology the CSE Department has a placement exam for incoming freshmen students who want to skip the CS1 course (http://www.cs.caltech.edu/placement). This placement exam also serves as the basis for case-by-case consideration placement into more advanced CSE classes. In the CS1 exam, the students are required to describe the representation (data structures) for a singly linked list of integer-string pairs, write a routine (or routines) to manipulate the linked lists, describe how his or her "sort" procedure is invoked, and perform a complexity analysis on their algorithm. If a student wants to test out of CS2 (skipping both CS1 and CS2), an additional exercise requires the student to write a specification in clear English for a class to represent a graph, provide two different implementations meeting the specification/interface, and write a procedure to find the length of the shortest path between two graph nodes on top of the interface specified.

At the Department of Computer Science and Engineering at the Washington University at Saint Louis, students with significant computer science background and experience in object-oriented programming are invited to take the CSE placement exam (http://www.cse.seas.wustl.edu/EntryCourses/PlacementExam.asp) during the orientation week for possible placement in a course beyond CSE101G (CS1). The placement exam is designed assuming that students are comfortable with the following concepts: recursion, iteration (loops), constructors, methods with parameters and return values, classes and objects, references (pointers), simple data structures (arrays, linked lists, binary search trees, maps, etc.), class hierarchies, inheritance, method overriding, and polymorphism. The exam assumes a working knowledge of Java that is sufficient to read, modify, and write Java source code. Familiarity with specialized Java packages is not assumed.

The Computer Science and Engineering Department of Arizona State University has an exam to place students into CSE110 (pre-CS1) or CSE200 (CS1) (http://cse.asu.edu/AcademicPrograms/Undergraduate). The placement exam has 40 multiple-choice questions on the syntax of Java and Java-based programs. Most of the questions are not specifically tied to problem solving in general computer science, but are about program syntax.

The Advanced Placement (AP) exams (http://www.collegeboard.com/ap) by the College Board are administered to high school students to test their ability to perform at a college level in 19 subjects, including Computer Science. The exams consist of both free-response and multiple-choice questions, with the former graded by college professors and high school teachers, while the latter are automatically scored by computer. The computer science test is designed to test students' ability to design, write, analyze, and document programs and subprograms. These AP tests have been extensively researched, with numerous reliability and validity studies conducted to ensure appropriate psychometric properties. For example, reliability statistics for the 2003 computer science advanced placement exam range from .91 to .95. Validity studies compare how students with a given AP grade perform in an advanced college course. If students consistently succeed in that course, those AP grade levels provide an indication that they have the prerequisite skills. To check the validity of such decisions, studies are conducted in which the AP Exam is administered to students taking the college courses in question. The college students' grades on the AP Exam are then compared with their grades in the college course. The College Board also conducts numerous item statistics, and may remove items that do not meet certain criteria, such as a low or negative correlation with performance on the entire exam. These statistics and procedures are similar to ones we used and are described in a later section.

There are other CSE placement exams (e.g., http://www.fullerton.edu/admissions/ProspectiveStudent/PlacementExams.asp and http://www.bama.ua.edu/~tserve/placement2.html) that focus on placement in CS1. In some CSE departments these exams are mandatory while in others, they are optional or only required under certain circumstances. Most of the placement exams we researched serve one purpose—to place incoming, first-semester CSE students into the appropriate introductory CSE course (generally pre-CS1 or CS1). Further, most of the placement exams do not explicitly consider pedagogical contexts such as Bloom's taxonomy nor was there any indication the results were used to improve course instruction. In addition, we are not able to find any references or publications on the quality of these examinations—there are no formative or summative analyses available. Our placement exam design is also based on problem solving using computers rather than on programming. Our objective is to place students who demonstrate comprehension and application capabilities into CS1 and to advise the other students to take pre-CS1. Overall, in our project, we aim to emphasize pedagogical contexts as well as validation of our work-in-progress. In the next section, we discuss the design details of our placement exam.

Design of a Placement Test

Designing and validating a good placement test is challenging. We relied heavily on classical test theory (Thorndike, 2004) in the design and validation of our placement test. However, the use of our test for both placement decisions, as well as providing feedback for needed instructional revisions, reflects new conceptual frameworks of assessment which consider curriculum development, psychological theories, and measurement theory (Mislevy, 2003; Shepard, 2000). Multiple dimensions must be considered in order to develop an exam that correctly and reliably assigns students to the proper first course in computer science. First, the areas of student knowledge and skills to be assessed must be carefully selected. Secondly, the exam questions must be designed to match the level of understanding expected of the students. Further, the questions must have an appropriate level of difficulty. If the questions are too difficult or too easy, the exam will not effectively distinguish between the knowledge and skill levels of the students. In addition, the test must be reliable and valid. And finally, placement criteria that reliably assign students to the proper course must be established. All these issues (excluding the placement criteria) are relevant to the design of a test for any course. Each of these aspects is described below.

Knowledge and Skills

As described earlier, we have designed our placement test to serve two purposes: (a) to determine the readiness of a student for our CS1 course, and (b) to validate student learning in our CS1 course. In order to accomplish these objectives it was important to accurately identify which topics should be tested as prerequisite skills for the course and topics to be tested as post-test measures. We used the ACM/IEEE Computing Curricula 2001 guidelines and our CS1 course descriptions to determine five content areas to be tested for prerequisite skills and five content areas to assess student learning in our CS1 course. For each content area, CSE instructors and researchers teaching and developing the curriculum for the lower level CSE courses developed a detailed list of topics and associated levels of competency expected of students in each course. Table 1 lists the ten content areas covered on our placement test. The first five content areas (1-5) address prerequisite skills and second five content areas (6-10) represent the topics students are expected to know after completion of CS1.

Design of Individual Questions.

Our work on design of the placement exam began in Spring 2003 with development of individual questions. Questions were drawn from multiple sources, including textbooks, previous exams and classroom exercises. It was also necessary to generate new questions to ensure that all content areas were adequately covered. Both CSE and education personnel carefully reviewed all questions to verify their appropriateness. We decided to use only questions at four Bloom levels: knowledge, comprehension, application and analysis. The motivation is that questions in evaluation and synthesis would require longer student time per question, would make the test longer, and are more difficult to frame within a multiple choice format.

The quality of a question can be measured in different ways. Item difficulty is determined by the percentage of test takers who answer the question correctly. If a question is too easy (very high mean) or too difficult (very low mean), it does not provide a meaningful discriminator and should be carefully considered for revision or exclusion. For our test, the target mean for each question was between 0.4 and 0.85.

The item-total correlation for a question shows the strength of the relationship between the students' response to a question and their total score. A good question should have a strong positive correlation between the two. Low values for this measure—i.e., a student who scores high overall yet fails to answer the question correctly, and vice versa—indicate that the question may be confusing, ambiguous or even wrong. While a value of 0.3 is generally regarded as a good target, we chose a value of 0.2 as acceptable.

For multiple-choice questions, the frequency of response for the choices also may be used to measure the overall quality of a question. If students choose only two of the five possible choices, then the three unpicked choices are not providing any discrimination and should either be modified or dropped.

At the end of each testing cycle, exam questions in the pool are evaluated for their effectiveness and discrimination power. Questions that are either too easy or too difficult are eliminated or modified and returned to the question pool for validation before being used in the next placement test. Similarly, questions in the pool must be periodically replaced to keep the questions from being repeated too many times.

Other ways to compute these measures are reported in literature. See Haladyna (2004) for a broad overview of literature in designing and validating multiple choice questions. For example, item discrimination attempts to capture the belief that students with high scores would choose the right answer and vice versa. It describes the ability of the question to measure individual differences sensitively among the test takers. Many different approaches have been used to measure item discrimination including item-total correlation (which we used) and trace line analysis (Haladyna 2004).

Reliability and Validity.

In addition to careful delineation of the skills and knowledge to be measured, the test must exhibit reliability and validity. Reliability is the degree of consistency, precision, and repeatability. Scores on reliable measures are not greatly influenced by random error. Thus reliability is the extent to which a test is repeatable and produces consistent scores. Several types of reliability are described in literature:

1. Inter-Rater or Inter-Observer Reliability: This is the degree to which different test takers answer the questions consistently.
2. Test-Retest Reliability: This is used to assess the consistency of the test from one administration to another.
3. Parallel-Forms Reliability: This measures the consistency of the results of two tests constructed in the same way from the same content domain.

4. Internal Consistency Reliability: This is used to assess the consistency of results across items within a test.

For our research, we focused on internal consistency reliability, which is a measure of the item-to-item consistency of a subject's responses within a single test. This type of reliability is measured by a reliability coefficient, which ranges from 0 (no reliability) to 1 (perfect reliability). In assessing the reliability of our test, we used Cronbach's Alpha statistic. Reliability measures of our test have ranged from .70 to .74, which is acceptable for research purposes. We do not, however, believe that our reliability index is high enough, given that we are making critical, individual decisions about student placement. Our goal is to continually refine the test, with the goal of reaching a reliability coefficient of .80 or higher.

Cronbach's Alpha can be defined as a function of the number of questions and the average inter-correlation among the questions. The standard formula for the standardized Cronbach's Alpha is given by:

$$\alpha = \frac{N \cdot \bar{r}}{1 + (N - 1)\bar{r}} ,$$

where N is equal to the number of questions and $\bar{r}$ is the average inter-item correlation.

While reliability reflects the consistency of test scores, validity refers to how well the test measures what it purports to measure. Our main concern is with content validity and predictive validity. Content validity reflects how adequately the test samples and represents the content domain. It is not measured statistically, but instead is judged by expert opinion on the representativeness of the items. The panel of CSE and education personnel working on this project carefully reviewed the items to determine content validity.

For any screening test to be useful, it must have predictive validity. For example, the GRE scores are used to predict success in graduate school; SAT is used to do predict performance in undergraduate program. In our case, we want to use the result of this test to measure the likelihood of a student to succeed in CS1 course. Predictive validity is computed by correlating the score in the placement test and some outcome measure (say course grade). We computed two predictive validity measures for our test. First, the overall predictive validity for our test was determined by correlating a student's total score on the placement test (all 50 questions) with his/her exam scores in the course. Second, the placement predictive validity is computed by correlating students' scores on the first twenty-five questions with their total points in the course.

Placement Criteria.

Determining the cutoff score for placement is a critical decision and must be arrived at carefully. Ideally, students who are placed into a CS1 course should succeed in the course. A cutoff score that is too low will result in a higher percentage of students having difficulty in the course. In our placement test, we give one point for each correct answer irrespective of the difficulty of the question. There is no penalty for a wrong answer. During the development of the test, we initially chose 50% to be the cutoff score. After initial complaints from students regarding the rigidity of the cutoff and some students' ability to work hard to perform well in the course, we have since adopted two cutoff points: 40% and 48%. That is, students who scored less than 40% on the first questions were advised to take pre-CS1. Students who scored above 48% were advised to stay in CS1. Students who scored between the two cutoff points were advised that they would need to put in extra effort if they chose to stay in CS1. As the test is updated and revised to improve measurement statistics, however, we expect to have 50% as the final cutoff score.

Implementation of the Placement Examination

Software developed by the National Center for Information Technology in Education (NCITE), within the College of Education and Human Sciences at UNL, was used for administration of the placement exam (Zygielbaum & Feese, 2005). The system allows students to go back and change answers until a final submission of the entire test. It also allows instructors to set up tests using a specially developed teacher interface and to view the results on-line. The individual student information captured is student ID, student response to each question, correct and incorrect statistics for each test item, total correct for first 25 questions, total correct for second 25 questions and a total score. Special controls are included to allow the instructor to restrict the number of times students can take the test and to allow a proctor login for supervised testing. Although we administered the test in a supervised environment, the ability exists for students to take the exam using a web browser on any computer within the university network. In Fall 2004, we started administering the test in our general-purpose computer lab at the CSE Department. Each student was required to provide an ID and our lab monitor approved that, logged into the test (by providing the proctor login password), and allowed the student to take the exam.

The duration for the placement exam is one hour, with a total of 50 questions and 10 content categories. There are five questions in each category, with one each from knowledge, comprehension, application, analysis, and synthesis competence levels (using Bloom's taxonomy). The time allocated for each group of questions is roughly six minutes, with a breakdown as follows: (1) knowledge: 10 seconds, (2) comprehension: 30 seconds, (3) application: 1 minute, (4) analysis: 2 minutes, and (5) synthesis: 3 minutes. Students are not informed about the competence level of each question.

The presentation order of the questions is by the competence level within each content area. All the "knowledge" questions are presented first, followed by the "comprehension" questions, and so on. This procedure allows higher level questions to build upon the lower level questions, e.g. to answer a comprehension question, one should understand the requisite knowledge. The test is administered online at the Testing Center on the UNL campus. Students are allowed to take the exam only once, and are given a week to take the exam.

Analysis of the Placement Examination

After extensive work in question development during Spring 2003, the exam was first administered as a post-test at the end of the Summer 2003 session. We asked our CSE155 (CS1) students to take the placement exam to fine-tune the logistics of test administration and the questions themselves. Subsequently, the placement exam was refined prior to administration to students enrolling in the regular sessions of CSE155 in Fall 2003. Tables 3 and 4 present data for Fall 2003 through Fall 2004, showing placement test decisions and pre- and post-test scores by content area and level of Bloom's Taxonomy.

Insert Tables 3 and 4 about here

In our reporting below, we divide our discussions into two major sections. First, we report on our comprehensive validation results, based on the placement exam scores collected from four different courses CSE105 (CS0), CSE155 (CS1), CSE155H (CS1 honors), and CSE156 (CS2). Second, we report on the trends and patterns, based on the data collected for CS1 (both CSCE155 and CSCE155H) students in Fall 2003, Spring 2004, and Fall 2004.

Comprehensive Validation Results (based on Fall 2003)

At the beginning of the Fall 2003 semester, students scheduled for CSE105 (pre-CS1), CSE155 (CS1), and CSE156 (CS2) took the revised version of the placement exam. Since this was the first administration of the exam as a placement test, it was administered across a variety of student populations, including the honors class and students from CS2. This procedure allowed detailed analyses between various population groups. Fifty-five CSE105 students, 78 CSE155 students, 13 CSE155H (honors) students, and 23 CSE156 students took the test. The reliability of our instrument was approximately the same as the test given in Summer 2003 (coefficient alpha = .74); however, the larger number of students who participated during Fall 2003 (n=223) resulted in better stability. In general, CSE156 students performed the best on both sets of questions (61.6% and 63.2%, respectively). Honor students in CSE155H (61.5% and 58.6%, respectively) also did better than other CSE155 students (53.8% and 52.8%, respectively). Students in all groups except CSE155 (Honors) performed better on the first set of questions than on the second set. CSE105 students performed most poorly overall (43.1% and 42.4%, respectively).

After examining the placement test scores, we advised the CSE105, CSE155 and CSE155H students as follows (see Table 3).

We observe that, in general, students who performed well, did so on both the first set of questions and the second set of questions. CSE155H students scored well on the first 25 questions (61.5%), much better than their counterparts in CSE155 (53.8%). The majority of CSE105 students scored below 40% on the first 25 questions (38.2%) or were in the borderline group (34.5%). Most of the CSE155 and CSE155H students scored 48% or better on the first 25 questions. Based on these statistics, we think the cutoff points for the exam were appropriate.

Pre-post comparisons from Fall 2003 semester showed an increase from pre- to post-testing; the overall mean for the pre-test was 27.59 (55%), which improved to 34.63 (69%) for the post-test. A t-test for the significance of difference between these two mean values was highly significant ($t(63) = 11.036$, $p < .001$), providing validation of the instructional effectiveness of the CS1 course. Placement test score also served as a significant predictor of total test scores in CS1, an indication of the test's predictive validity.

Another analysis involved looking at differences in total points scored in the CSE155 course between students scoring less than 48% (the cut percentage for CSE155) and those scoring above 48% on the first twenty-five questions on the placement test. A one-way ANOVA found a significant difference between these two groups on total course points ($F(1, 64) = 4.76$, $p < .05$), indicating that students who scored higher on the placement test accumulated more course points and consequently received a higher grade.

For more detailed analysis such as efficiency and time-on-task factors, please refer to Soh et al. 2005.

Trends and Patterns (based on Fall 2003, Spring 2004, and Fall 2004)

The placement exam again underwent revisions before Spring 2004 and Fall 2004. The statistics on the placement exam based are shown in Tables 3 and 4. Of the borderline students, most elected to take CSE155 (all in Spring 2004 and 19 out of 23 in Fall 2004).

In general, the use of the placement test across three semesters, and its ongoing revision, has resulted in fewer students being identified as borderline and more students identified as ready for CS1. We are also seeing gradual increase in placement test scores, which could be attributable to an improved test or greater knowledge and skills of students coming from high school

In terms of the overall predictive validity, the placement test score (all 50 questions) in all three semesters was a significant predictor of his/her exam scores in the course (Fall 2003: $r = .48$, $p < .001$, $n = 64$; Spring 2004, $r = .58$, $p < .001$, $n = 66$; Fall 2004: $r = .77$, $p < .001$, $n = 58$) This shows that the full test, even though the students are not expected to know the contents in the second part, is a good predictor for success in the course. The increasing trend in correlation also indicates that our placement exam has improved after each revision in its predictive validity.

In terms of the placement validity, the correlation of .29 was found at the .01 level in Spring 2004, with the regression analysis showing that student's score on the first half of the placement exam was a significant predictor of their total points in the course. In Fall 2004, the placement predictive validity was even higher, with a correlation of .63 that was significant at the .001 level. This shows the placement section of the test is also a good indicator for success in the CS1 course, and has also improved in its validity in the semesters in 2004.

In addition to the test's use for placement decision, it is also used as a means to assess learning in CS1. Results from pre-post testing for the CS1 course across three semesters showed a significant increase in achievement. (See Table 4). Subscale scores, reported by content area and level of Bloom's taxonomy also showed increases from pretest to posttest. Greatest increases were in content areas emphasized in the CS1 course. In general, we believe the scores in Table 4 provide evidence of measurement validity because they show consistent patterns reflecting what we know about the content knowledge of students entering the program and what was emphasized (or not emphasized) in our CS1 courses.

Students scored highest on the comprehension questions and lowest on the higher-level analysis questions. This result is to be expected, since comprehension knowledge is a precursor to higher-level analysis skills. It would be anticipated, however, that the highest scores would be on the knowledge questions. Our hypothesis is that students are coming from high schools without the necessary computer science factual knowledge, including terminologies and definitions, but are able to use general mathematical and reasoning aptitudes to perform well at the comprehension level. In other words, students can logically come up with a solution for the comprehension questions more easily than they can generate a definition for a concept they have not previously encountered. The difference in scores on the comprehension and application pretest problems across the reported semesters are likely due to revisions of the exam. For example, in Fall 2004 application questions involving pseudo code were replaced with problems using actual Java code. In addition, some of the easier comprehension questions were made more difficult in order to improve discrimination.

In examining content area scores, results show that students tend to score high on object-oriented programming and low on algorithms and problem solving. The low scores on algorithms and problem solving are likely due to the higher- order thinking and analysis required for these difficult topics. Pretest scores on object-oriented programming show an upward trend, suggesting greater coverage of the topic in high school.

Individual Question Analysis

Our procedure for the placement exam is to continually validate and revise the questions. Through continual testing and revision we are gaining knowledge about how to design exam questions including how to word the questions, how to set up the options in a multiple choice format, and how to assess the difficulty level of each question. In order to support this work we obtain for each question the scoring mean, the corrected item-total correlation value, the list of choice percentages of all the options, the scoring standard deviation, and the Cronbach's alpha if the question were deleted. These values and their indications have been discussed in the previous sections. Here, we provide several individual examples on how we use these values (Table 5) to revise our questions.

———————————————————
Insert Table 5 about here
———————————————————

<u>Question 3 – A confusing question</u>

3. There are 25 students, 12 of them are taking an English class, 11 of them are taking a history class. 4 students are not taking either English or a history class. How many students are taking both an English and a history class?

a. 1
b. 6
c. 2
d. None of the above

Though this question's level of difficulty is acceptable (mean = 0.600), its corrected item-total correlation value is not sufficient (.129). This means students who scored high on the overall exam performed poorly on this question, and vice versa. After looking at the distribution of choices, we see that choices (a) and (b) are not often selected as the correct answer and therefore are not performing their role of distracters. Thus, we changed option (a) to 21, and option (b) to 11. The rationale behind selection of these two new options is to see if students will solve the problem by performing a simple subtraction (25 – 4 = 21) or a simple minimum function (min (12, 11) = 11).

<u>Question 5 – A difficult question</u>

5. Which of the following statements is saying the same thing as this quote?
"Only if Jim is at least 13 years old and Sue is not dating anyone else, then Jim and Sue may go out together."

a. If either Jim is younger than 13 or Sue is dating someone else, then Jim and Sue may not go out together.
b. If it is not the case that "Jim is younger than 13 and Sue is dating someone else", then Jim and Sue may go out together.
c. If it is not the case that "Either Jim is younger than 13 and/or Sue is dating someone else", then Jim and Sue may not go out together.
d. All of the above are the same.

Though this is a difficult question (mean = 0.388), its corrected item-total correlation is good (.26), and we do not want to significantly change this question. To reduce the difficulty slightly (to reach mean = 0.400), we can make choice (b) less "attractive" to students. As can be seen from Table 5, both choices (a) and (b) dominate the student's response. One possible change is:

b. If Jim is at least 13 years old, then Jim and Sue may go out together.

<u>Question 13 – An easy question</u>

13. Which if-else statement correctly expresses the following two if-then statements?

If (a is less than b) then x = y
If (a is greater than or equal to b) then x = z

a. If (a is less than or equal to b) then x = y
Else x = z
b. If (a is less than b) then x = z
Else x = y
c. If (a is greater than b) then x = z

Else x = y

d. *If (a is greater than or equal to b) then x = z*
   Else x = y

This question is too easy (mean = .157) and is therefore a poor discriminator of student knowledge and should be rewritten. In analyzing the test results we noted that choices (a)-(c) were rarely selected as the correct answer. In order to improve this question we can replace the question with a brand new question, make the two if-then statements in the question more complicated, or make the choices (a)-(c) more "attractive." For example, we could replace choice (a) with the following:

If (b is greater than or equal to a) then x = y
Else x = z

To make the question even more difficult, we could also rephrase the question by changing the question to "*Which if-else statement <u>in</u>correctly expresses the following two if-then statements?*" Further, if we use the options listed below, students would be forced to understand every option in order to answer the question.

(a) I only
(b) I & II
(c) II & III
(d) *I, II, and III*

<u>Question 15 – A good question with a non-factor choice</u>

15. Assume that sum (a, b) adds a and b while prod (a, b) multiplies a and b. What is prod(sum(a, c), sum(a, prod(c, d)))?

a. $a^2 + ac(1 + d) + c^2d$
b. *a + ca + cd*
c. *(a + c)a + cd*
d. None of the above

This question is adequate in terms of difficulty and is a satisfactory indicator of the overall exam score (correlation = .270). However, option (b) was almost never selected as the correct answer. Thus, we changed the option to the following.

(b) 2ac + ad

The rationale behind this choice is to determine if students will confuse "sum" with "product".

<u>Question 24 – An almost-perfect question</u>

24. In the computer world, how many numbers can be represented by eight bits?

a. 1
b. 4^2
c. 4^3
d. 4^4

This question exhibits good psychometric properties. The scoring mean .425 could be a little bit higher; however, we leave this question unchanged.

Note that although we follow the above as guidelines on how to revise the questions in our placement exam, we have to keep in mind that not all questions should be difficult, however the questions should be reliable in discriminating between those who should be placed into CS1 and those should not. Thus, it is critical for a question to have a good corrected item-total correlation.

Conclusions and Future Work

Although we have used the placement exam to assign students to pre-CS1 and CS1 for the past three semesters, we plan to make additional changes to the exam and conduct more comprehensive analyses and investigations. We are investigating changing the scoring policy to discourage guessing. We are exploring a procedure to give 0 points for a wrong answer, 1 point for an unanswered question, and 3 points for a correct answer. This change will necessitate modifying the cutoff points. We also plan to add more questions to the question pool and randomize selection of the questions. We will continue question-by-question, group-by-group, topic-by-topic, and Bloom's competency level-by-level analyses of all questions. These findings will form the basis for the evaluation of our CSE155 (CS1) course. Our long-term goal is to make the placement test adaptive, allowing the test to accurately measure the student's competence in the least amount of time.

Summary

We have described a placement exam designed within the overall context of a Reinventing CSE Curriculum Project. As such, we have utilized educational research components in the design of the exam, such as Bloom's taxonomy, and we have adhered to the ACM/IEEE Computing Curricula 2001 guidelines. We have presented the process we followed to develop the exam, discussed exam results, and highlighted interesting observations. Much of this research is broadly applicable to testing and evaluation for any course. We plan to carry out further investigations in an effort to continually refine the placement exam, provide additional insights to our preliminary findings, and improve our introductory CSE courses and our overall curriculum.

Acknowledgement

This work is supported in part with funding from National Center for Information Technology in Education (NCITE). We would like to thank Art Zygielbaum for his support and Suzette Person, Charles Riedesel, Chao Chen, Smitha Kasinadhuni, Joseph Bernadt, Andhy Kosenander and Traci Fink for their invaluable help in implementing this project.

References

ACM/IEEE-CS (2001). *Computing Curricula 2001*. Retrieved May 12, 2004 from http://www.sigcse.org/cc2001/.

Bloom, B. S. (l956). *Taxonomy of educational objectives: Book 1, Cognitive domain*. New York: Longman.

Haladyna, M. (2004). *Developing and validating multiple-choice test items*. Marwah, NJ. Lawrence Erlbaum Associates. Third Edition.

Mislevy, R. J., Steinberg, L. S., & Almond, R. G. (2003). *On the structure of educational assessments*. *CSE Technical Report 597*. Los Angeles, Ca: UCLA Center for the Study of Evaluation.

Shepard, L. (2000) The role of assessment in a learning culture. *Educational Researcher*, 29 (7) 4-14.

Soh, L.-K., A. Samal, S. Person, G. Nugent, J. Lang (2005). Designing, implementing, and analyzing a placement test for introductory CS courses, *Proceedings of the 36th SIGCSE*

Technical Symposium on Computer Science Education (*SIGCSE'2005*), St. Louis, MO, February 23-27, pp. 505-509.

Thorndike, R. M. (2004). *Measurement and evaluation in psychology and education*. New York: Prentice Hall.

Zygielbaum, A., & Feese, R. (2005). Affinity learning. In L. PytlikZillig, L, Bodvarsson, M., & Bruning, R. (Eds.), *Technology-based Education: Bringing Researchers and Practitioners Together*. Greenwich Connecticut: Information Age Publishing.

Table 1. Selected Curriculum 2001 Topics Covered in the Placement Exam

G	Topics to be covered	Expected Competence
1	Functions, sets, relations [DS1]	Application
	Basic logic [DS2]	Analysis
2	Fundamental data structures [PF3]	Synthesis
3	Fundamental programming constructs [PF1]	Synthesis
4	Algorithms and problem-solving [PF2]	Application
	Fundamental computing algorithms [AL3]	Comprehension
	Recursion [PF4]	Comprehension
5	Machine level representation of data [AR2]	Comprehension
	General knowledge	Comprehension
	* Computer Architecture [AR1]	Comprehension
	* Programming Languages [PL1]	Knowledge
	* Software design [ES1]	Knowledge
	* Software tools and environments [SE3]	Knowledge
6	Mastery of fundamental data structures [PF3]	Synthesis
7	Mastery of fundamental programming constructs [PF1]	Synthesis
8	Object-oriented programming [PL6]	Application
9	Fundamental computing algorithms [AL3]	Comprehension
	Recursion [PF4]	Analysis
10	Event-driven programming [PF5]	Synthesis
	Software engineering	Comprehension
	* Object-oriented design, reusable classes [SE1]	Comprehension
	* Testing fundamentals [SE6]	Comprehension

Table 2. Six Categories of the Cognitive Model in Bloom's Taxonomy and Sample Questions

1. Knowledge: recall of data

Example Question:

An algorithm is best characterized as

a. a precise rule (or set of rules) specifying how to solve a problem
b. a logical expression to solve a problem
c. a rule (or set of rules) used in writing a program
d. all of the above

2. Comprehension: Understand the meaning, translation, interpolation, and interpretation of instructions and problems. State a problem in one's own words.

Example Question:

Access time to the i^{th} element of an array is
a. logarithmic with respect to the number of elements in the array.
a. linear with respect to the number of elements in the array.
b. independent of the number of elements in the array.
c. a single machine instruction.

3. Application: Use a concept in a new situation or unprompted use of an abstraction. Apply what was learned in the classroom into novel situations in the workplace.

Example Question:

Let
$X_{10} = 0$
$X_9 = 1$ and
$X_{i-1} = 2 * X_{i+1} + X_i$ for $1 \leq i \leq 9$

What does X_5 equal?
a. 5
b. 11
c. 13
d. 7

4. Analysis: Separate material or concepts into component parts so that its organizational structure may be understood. Distinguish between facts and inferences.

Example Question:

Assume that sum(a, b) adds a and b while prod(a, b) multiplies a and b. What is

prod(sum(a, c), sum(a, prod(c, d)))? a. $a^2 + ac(1 + d) + c^2d$ b. a + ca + cd c. (a + c)a + cd d. none of the above
5. Synthesis: Build a structure or pattern from diverse elements. Put parts together to form a whole, with emphasis on creating a new meaning or structure.
6. Evaluation: Make judgments about the value of ideas or materials.

Table 3. Placement Test Recommendations: Percentage of Students

Semester	**Number of Students**	**CS 105 (Below 40%)**	**CS 155 (Above 48%)**	**Borderline (40 – 48%)**
Fall 2003	91	24%	45%	31%
Spring 2004	84	12%	76%	12%
Fall 2004	93	9%	63%	21%

Table 4. Student Mean Scores on Placement Exam

Measures	Fall 2003 pre	Fall 2003 post	Spring 2004 pre	Spring 2004 post	Fall 2004 pre	Fall 2004 post
Placement exam content area						
Functions, relations, sets, and basic logic	.67	.74*	.57	.71***	.76	.77
Fundamental data structures	.53	.77***	.50	.69***	.59	.73***
Fundamental programming constructs	.67	.77**	.66	.77***	.67	.76***
Algorithms and problem solving	.39	.54***	.42	.54***	.47	.70***
Machine level representation of data	.53	.61***	.56	.60	.51	.52
Fundamental data structures	.69	.71	.60	.71***	.64	.80***
Fundamental programming	.37	.50**	.36	.53***	.35	.52***
Object-oriented programming	.59	.89***	.67	.85***	.81	.92***
Fundamental computing algorithms	.46	.66***	.44	.57***	.49	.70***
Event-driven programming, software engineering	.54	.75***	.63	.77***	.54	.80***
Bloom's level						
Knowledge	.60	.73***	.55	.70***	.60	.72***
Comprehension	.60	.75***	.66	.77***	.56	.78***
Application	.56	.66***	.49	.66***	.62	.68**
Analysis	.35	.47***	.46	.56***	.54	.71***
Total exam (50 questions)	.55	.69***	.54	.76***	.58	.72***

*p < .05, ** p< .01, p < .001

Notes. Mean scores can range from 0 – 1. A mean score of 1.0 would indicate that all students responded correctly to the questions in this content area.

Table 5. Item Statistics of Five Example Questions

Question	**Mean**	**Corrected Item-Total Correlation**	**Choice (a)**	**Choice (b)**	**Choice (c)**	**Choice (d)**
3	0.600	0.129	4.7%	8.2%	60.7%*	26.2%

5	0.388	0.260	38.8%*	38.8%	5.9%	16.5%
13	0.913	0.157	3.5%	1.2%	3.5%	91.8%*
15	0.525	0.270	51.8%*	1.2%	24.7%	22.4%
24	0.425	0.381	18.8%	16.5%	22.4%	42.4%*

*An asterisk by the percentage value of a choice indicates that the choice is the correct answer for that question.