How Students Measure Up: An Assessment Instrument for Introductory Computer Science

Adrienne Decker

Dissertation Defense for the Degree of Doctor of Philosophy

May 1, 2007
How should we teach Introductory CS?

- Programming or no programming
- Graphical or text-based programs
- Individual or collaborative learning techniques
- Which paradigm
- Which language
Predictors Research

- We can find predictors, but what are we using as a measure of success? (Ventura, 2003)
- No good assessments available for task
  - GRE Subject-Test in Computer Science
  - ETS Major Field Test in Computer Science
  - AP Computer Science Exams (A & AB)
How can we measure success?

- Exam Grades
- Assignment Grades
- Overall Course Grades
- Lack reliability and validity
  - Reliability
    - “degree of consistency among test scores”\(^1\)
  - Validity
    - the ability of a test to “reliably measure what is relevant”\(^1\)

\(^1\) – Marshall and Hales (1972)
Proposed Solution

- Create an assessment for the introductory computer science courses.
  - Language-independent
  - Paradigm-independent
  - Programming-first approach (CC 2001)
  - Timed paper-and-pencil exam
  - Reliable
  - Valid
First Year Recommendations

- CC2001 gives six approaches to the introductory curriculum
  - Programming-first
    - Functional-first
    - Imperative-first
    - Objects-first
  - Non-programming-first
    - Algorithms-first
    - Breadth-first
    - Hardware-first
Introductory Approaches

- Goals for the first year across all six approaches including discrete math coverage
- Two-semester introductory sequence
Intersection of the Programming-first Approaches

- CS1 only: 60% of topics
- CS1+CS2: >80% of topics
Creation of Intersection

- Several inconsistencies uncovered
  - Indication by CC2001 that all topics from a knowledge unit are covered. The reading of the sample syllabi makes no mention of the topics from that knowledge unit.
Refining the Topic List

- 75 topics total
- Topics omitted from assessment due to:
  - Time constraints
    - Programming Process (design, debugging, testing)
    - Concepts Underlying Process (abstraction)
    - Exploration of Programming (tools, environments)
  - Coverage in other courses
  - Difficulty in determining coverage
Omitted Topics

- Belong in CS1-CS2 sequence
- Taught in classroom
- Must be assessed in other ways
  - Laboratory exercises
  - In-class assignments
  - Quizzes
Learning Objectives

- CC2001 gives learning objectives for knowledge units
- Matched topics to learning objectives
- Found that definitions of some terms were needed
  - Within learning objectives
  - Within topics themselves
- Some topics did not have corresponding learning objectives
Exam Creation & Critique

- Questions were based on topics not omitted and learning objectives
- Language was necessary: Java used
- Critique by course instructors who would give exam as well as independent reviewers
  - Two schools
  - Five total reviewers
Grading Rubric

- Multiple Choice
- Free Response
- Subjective Questions
  - Partial Credit
Exam Administration

- Closed book
- Closed notes
- Closed neighbor exam
- 3-hour time limit
- Exam answer booklets only identified by number, not name
Grading Exam – the Raters

- Subjective questions:
  - Scored by two raters
  - Rater scoring disagreements range:
    - 12-29% of exams
  - Raters met to resolve discrepancies
Grading Exam – the Raters

- Rater 1 correct 44.5% of the time
- Rater 2 correct 47% of the time
- For all subjective questions, there was at least one exam where both raters had given the incorrect grade the first time.
Grading Recommendations

- Multiple raters and ratings for all subjective questions
  - Resolution of discrepancies
- Grade simultaneously
- Grade anonymously
Study Design

- Students enrolled in CS2 (CSE 116) at UB
  - Fall 2005
  - Spring 2006
- IRB approval obtained
- Participants completed exam and filled out demographic questionnaire
- Data collected about participants’ grades in CS1 & CS2
CS2 Population Information

- 135 students enrolled
- 14 students resigned
- 121 students eligible for participation in study
  - 110 men, 11 women
  - 45 freshmen, 76 non-freshmen
- Major:
  - 37 computer engineering (CEN)
  - 46 computer science (CS)
  - 38 other majors
Study Participants

- 100 students chose to participate (83%)
  - 90 men, 10 women
  - 52 freshmen, 47 non-freshmen
  - Majors:
    - 26 CEN
    - 50 CS
    - 23 other majors

- Year in school and major suffer from self-report vs. university records mismatch
Non-participants

Number of Students Not Participating in Study

Letter Grade Received in CS2

- A: 1
- A-: 2
- B+: 3
- B: 3
- B-: 1
- C+: 1
- C: 1
- C-: 1
- D: 1
- F: 6

Number
Exam Statistics

- Minimum score: 138 (38.9%)
- Median score: 254 (71.7%)
- Maximum score: 334 (94.3%)
- Mean score: 243.13 (68.6%)
Time to Complete Exam

- Average time to complete: 2:31
- Time to complete vs. Exam Performance
  - No correlation
Time to Complete Exam

- Students who took full 3 hours vs. students who left early
  - Did worse on exam
  - Did equally well in CS1 & CS2
Reliability

- Cronbach’s Alpha for internal consistency reliability 0.94.
Demographics

- No difference in performance between:
  - Men and women
  - Typically aged CS1-CS2 students and older students
  - Freshmen and non-freshmen
  - CEN or CS majors and non-majors
  - CEN and CS majors
Demographics

- No difference in exam performance:
  - How students took course
    - Consecutive vs. non-consecutive semesters
    - CS1 in Fall and CS2 in Spring vs. others
    - CS1 in Spring and CS2 in Fall vs. others
  - Students who repeated CS1 or CS2 vs. those who did not
Demographics

- No difference in exam performance:
  - Previous Programming Experience
    - Programmed prior to CS1 vs. no programming
    - Programmed in Java prior to CS1 vs. not
  - First Language Learning
    - Java first vs. some other language first
Demographics

- Possible difference (borderline $p$-value)
  - Programmed in C-derived language (Java, C, C++, C#) before taking CS1 vs. not
Positive Correlations

- Exam score correlated positively with:
  - CS1 grade
  - CS2 grade (with final exam)
  - CS2 grade (without final exam)
  - CS1-CS2 average grade
Contributions

- Process for analyzing and using curriculum document for creation of exam.
- Uncovered inconsistencies with CC2001 within sample syllabi and learning objectives.
- Intersection of programming-first approaches identified.
Conclusions

- Reliability Acceptable
- Face Validity
- Content Validity
  - Outside critique and review by experts
- Criterion Validity
  - Positive correlation between exam score and CS1 and CS2 grades
Conclusions – Potential Bias

- No difference in performance based on:
  - Gender
  - Year in school
  - Age
  - Major
  - Prior programming experience

- Possible difference in performance if previously used C-derived languages.
Future Work

- More student data
- Continuation of predictors research
- Additional versions (languages)
- Multiple institutions
- Updates for future curricula