How Students "Measure Up": Creation of an Assessment Instrument for Introductory Computer Science

Adrienne Decker

Dissertation Proposal Defense for
the Degree of Doctor of Philosophy

April 29, 2004

Curriculum Through the Ages

- Curriculum ’68, ’78, & ’91
- CC2001
  - Knowledge Areas carried over from Curriculum ’91
  - Distinction of Introductory/Intermediate/Advanced Courses
  - Introductory courses:
    - Programming-First
      - Functional-First, Imperative-First, Objects-First
    - Breadth-First
    - Algorithms-First
    - Hardware-First

Problems with Curricula

- CC2001 (and previous) gives topics for coverage, not full course models.
- Implementation in hands of instructor.
- Instructor creates new course materials in effort to satisfy curriculum suggestions
- How do we know if materials are effective?

The Usual Suspects

- Exam Grades, Assignment Grades, Overall Course Grades
  - Lack important characteristics of reliability and validity
    - Reliability - “degree of consistency among test scores” (Marshall and Hales 1972)
    - Validity - the ability of a test to “reliably measure what is relevant” (Marshall and Hales 1972)

Reliability

- Time-sampling Method
  - Test administered at two different times.
- Parallel-forms Method
  - Two tests are created that are designed to test same content.
- Internal-consistency Method
  - One test split in halves. Both halves are compared.

(Marshall and Hales 1972; Ravid 1994; Kaplan and Saccuzzo 2001)

Validity

- Face-validity
  - Does the test seem to ask the right questions?
- Content-validity
  - Expert opinion on the contents of the test.
- Criterion-validity
  - Test corresponds to a particular criterion.

(Marshall and Hales 1972; Ravid 1994; Kaplan and Saccuzzo 2001)

Motivation

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

AP Exam Analysis

- Compared AP Exam Grades and CSE 115 letter grades (2000 – 2002)
  - Computer Science A Exam
    - Correlation shown between AP CS A Exam and CSE 115 Letter grade $r_s(49) = .42$, $p < .01$
  - Computer Science AB Exam
    - No correlation between AP CS AB Exam and CSE 115 Letter grade $r_s = .21$, $n = 27$, $p > .05$

Proposed Solution

- Create an assessment for the introductory computer science courses.
  - Language-independent
  - Paradigm-independent
  - Programming-first approach (CC 2001)
  - Reliable
  - Valid

Prior Investigations

- Introductory Curriculum
  - Before CC2001 (and currently) – what is best way to teach introductory CS1?
    - Fincher (1999)
    - Evans (1996)
  - Graphical and/or Event Driven Approaches
    - Guzdial & Soloway (2002)
    - Alphonce & Ventura (2003)
    - Reges (2000)
  - Paired Programming (Nagappan, Williams et al. 2003)
  - When teaching programming, what paradigm? (Alphonce & Ventura 2003, Culwin 1999)

Prior Investigations

- Predictors Research
  - Pre-CC2001
    - Using mainly imperative programming.
    - No clear definition of curriculum
    - Not validated/reliable metrics
  - Ventura (2003): predictors for objects-first using CC2001
    - Missing good measure of success

Prior Investigations

- Non-majors course performance
  - Showed that students taught with Objects-Emphasized approach were better at Object-Oriented Programming in the long run.
  - Assessment through means of exam and resignation rates
  - No reliability or validity

Prior Investigations

- Assessment of Programming Skill for CS1
  - Conference on Innovation and Technology in Computer Science Education (ITiCSE) Working group (McCracken et al. 2001)
    - Created learning objectives for the introductory curriculum
    - Grading rubrics created
    - Math-intensive problems
    - Students lacking basic data structures background would be at a disadvantage.

Proposed Work

- Phase I - Topics, Questions & Grading for Programming-First Approaches
  - Is there an intersection of common topics for all three programming-first approaches?
    - What if there isn't one?
  - What types of questions should be asked?
  - How should the questions be graded?

Phase I Preliminary Results

- Identified that within programming-first approaches there are nine overlapping knowledge units.
- This set of topics accounts for ~50% of the topic coverage and ~50% of the course contact hours for each of the approaches.

Proposed Work

- Phase II - Survey of CS1 Educators
  - Ask for opinions about topical coverage and grading scheme in an effort to use data in future to establish content validity.

Proposed Work

- Phase III - Analysis of Survey
  - Does the test “measure up”?
  - If not, back to Phase I to reassess the assessment.

Proposed Work

- Phase IV – Field Testing
  - CSE 113/114/115/116 will take the test and their results will be graded using the grading rubric.

Proposed Work

- Phase V – Analysis of Grades, Reliability, and Validity
  - Test for inter-rater consistency
  - Internal consistency
  - Determination of Validity

Contributions & Significance of Proposed Work

- Intersection of Programming-First Introductory CS courses
- Assessment Instrument that can be used to test curricular innovations and/or as a measure of success.