Impact of Alternative Introductory Courses on Programming Concept Understanding

Allison Elliott Tew
W. Michael McCracken
Mark Guzdial
College of Computing
Georgia Institute of Technology

Overview

- Research Question
  Do different approaches in introductory CS really matter?
- Methodology
  Two alternative intro courses
  Pre- and post-test in common second course
- Results
  Significant differences on pre-test
  No significant differences on post-test
- Discussion
  Rethinking the CS 1 debate

Research Questions

- What is the impact of alternative pedagogical approaches to introductory computing?
- What computing concepts do students understand when beginning their second course?
- How does the introductory approach influence the learning outcomes of the second course?

Methodology

- Identified common topics
  Introductory
  - Conditionals, arrays, binary search trees, searching, sorting, recursion
  Advanced
  - OO Class definitions, polymorphism, dynamic binding, linked lists, hash tables, GUIs
- Quasi-experimental study design
  Pre-Test/Post-Test in 2nd course
  - Introductory and advanced topics on both tests
  Multiple choice question format (Lister et al 2004)
- Tracing questions
- Code completion

Sample Tracing Question

```java
int[] array1 = { 4, 5, 3, 6, 2, 7, 1 };
int[] array2 = { 7, 4, 2, 1 };
array1[3] = array1[5];
array1[2] = ...
```

<table>
<thead>
<tr>
<th>Course</th>
<th>Content</th>
<th>Audience</th>
<th>Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Computing'</td>
<td>Objects Early Simple data structures &amp; algorithms</td>
<td>CS Majors</td>
<td>Python</td>
</tr>
<tr>
<td>'Engineering'</td>
<td>Objects Early Traditional CS content in engineering context</td>
<td>Engineering Majors</td>
<td>MATLAB &amp; Java</td>
</tr>
<tr>
<td>2nd Course</td>
<td>Object-oriented programming Advanced data structures</td>
<td>CS &amp; Engineering Majors</td>
<td>Java</td>
</tr>
</tbody>
</table>

Impact of Alternative Introductory Courses on Programming Concept Understanding

Allison Elliott Tew
W. Michael McCracken
Mark Guzdial
College of Computing
Georgia Institute of Technology

Overview

- Research Question
  Do different approaches in introductory CS really matter?
- Methodology
  Two alternative intro courses
  Pre- and post-test in common second course
- Results
  Significant differences on pre-test
  No significant differences on post-test
- Discussion
  Rethinking the CS 1 debate

Research Questions

- What is the impact of alternative pedagogical approaches to introductory computing?
- What computing concepts do students understand when beginning their second course?
- How does the introductory approach influence the learning outcomes of the second course?

Methodology

- Identified common topics
  Introductory
  - Conditionals, arrays, binary search trees, searching, sorting, recursion
  Advanced
  - OO Class definitions, polymorphism, dynamic binding, linked lists, hash tables, GUIs
- Quasi-experimental study design
  Pre-Test/Post-Test in 2nd course
  - Introductory and advanced topics on both tests
  Multiple choice question format (Lister et al 2004)
- Tracing questions
- Code completion

Sample Tracing Question

```java
int[] array1 = { 4, 5, 3, 6, 2, 7, 1 };
int[] array2 = { 7, 4, 2, 1 };
array1[3] = array1[5];
array1[2] = ...
```
The following method "isSorted" should return true if the array "x" is sorted in ascending order. Otherwise, the method should return false:

```java
public static boolean isSorted(ArrayList<Integer> x)
{
    if (x.size() < 2)
        return true;
    if (x.get(0) > x.get(1))
        return false;
    return isSorted(x.subList(1, x.size()));
}
```

Connections to other coursework/concepts in engineering majors

- Engineering students who had completed the traditional intro CS course
- Electrical & Computer Engineering
- Binary Search Tree
- Students who had completed the engineering context intro CS course
- Electrical & Computer Engineer
- Logic in digital circuit design
- Flow-chart notation
- Scoping syntax more explicit in Java than in Python
- Listed nodes read logically from left to right, top to bottom
- Engineering Group understood this concept better than the Computing Group
- 45.87% of Engineering students answered correctly
- 18.18% of Computing students answered correctly
- Conditional (continued)
- Engineering and Computing both chose the same distractor most frequently
- Misunderstood scope of nested conditional statements
- Computing also often chose an additional distractor
- Mis evaluates first Boolean logic statement
- Interpretation based on Course Analysis
- Scoping syntax more explicit in Java than in Python
- Boolean logic covered more in Engineering lecture
- Connections to other coursework/concepts in engineering majors
- Flow-chart notation
- Logic in digital circuit design

Pre-Test Results
Results (continued)

- Results (continued)
  - Computing and Engineering both chose the same distractor most frequently
  - Correctly evaluated element comparison, but failed to aggregate evaluations
  - Engineering also often chose an additional distractor
  - Sorted in descending order

- Interpretation based on Course Analysis
  - Findings are consistent with Lister et al
  - Sorting not connected to engineering problem solving context
  - Issue of motivation

Pre-Test Results

- Pre-Test Results

Post-Test Results

- Post-Test Results

Comparison

- Pre-Test/Post-Test Comparison
Discussion

§ After the introductory course, students have different levels of understanding of introductory concepts
§ Yet, these differences tend to disappear after a second common course
§ Conclusions
Students are continuing to learn introductory concepts in their second course
Other conceptual/procedural differences may appear later in the curriculum
Reconsider the debate about “what” to teach in CS1

Future Work

§ Repeating pre/post-test study Fall Semester 2005
Course | Content | Audience | Languages
--- | --- | --- | ---
Computing | Objects Early Simple data structures & algorithms | CS Majors | Python
Engineering | Objects Early Traditional CS content in engineering context | Engineering Majors | MATLAB & Java
Media Computation | Objects Early Most traditional CS topics in media context | Liberal Arts Majors | Python

Future Work (continued)

§ Repeating pre/post-test study Fall Semester 2005 (continued)
Gathering data on reliability
§ Additional qualitative component
Observing students taking the pre-test
Using think aloud protocol
§ Studying the 2nd course

Acknowledgements

§ Undergraduate Research Student
Boris Goykhman
§ Faculty
Yannis Smaragdakis
John Stasko
Ashwin Ram
§ National Science Foundation
CCLI-ASA Program
CISE EI Program
Questions?