What Novice Programmers Don’t Know

Gary Lewandowski
Alicia Gutschow
Robert McCartney
Kate Sanders
Dermot Shinnors-Kennedy

Which programming concepts have little or no meaning for students?

• How frequently do students indicate a concept has less meaning?
• Does “don’t know” differ from “not applicable”?
• Does classification vary based upon course performance?
• Is it the concept or the vocabulary?

Outline

• Bootstrapping
• Card Sorts and Ragbags
• Methodology
• Results
• Conclusions

Bootstrapping

• NSF sponsored workshop: Sally, Marian and Josh
• Multi-national, multi-institutional study
• 22 researchers from four continents, six countries
• Data from 247 students
• How do first-competency students organize programming concepts?

Card Sorting

Elements in data flow

- Data itself
- Not applicable
- Don’t know
- Things contained in a program
  - Things that are
  - Things that are not

- The way things work in programs
  - Functions
  - Things about variables
  - Things you can do
  - On its own

Ragbags

• Don’t Know (DK): never knows it
• Not Sure (NS): doesn’t know in this sort
• Not Applicable (NA): not applicable in this sort
  – Boolean sorts may or may not have an NA
Methodology: Tagging the data

Agreed on 4280 of 4410 categories

Sometimes you need a lawyer to settle your disputes...

Methodology: Performance

Students rated in each course
1 = failing
28 students not rated
...
5 = Picasso

50 students
106 students
63 students
Average Performance Rating

Methodology: Abstract Concepts

11 Scholars categorized concepts as
Abstract or Concrete

7 of 11 described as abstract

Methodology: Meaningfulness

How meaningful is a concept to a student?

Levels of how specific each one is

If-then-else is always meaningful (A)

Boolean is usually meaningful (U)

Type is limited meaningful (L)

Scope is never meaningful (N)

Data: Frequency of Use

<table>
<thead>
<tr>
<th>#students</th>
<th>247 total</th>
<th>50 low</th>
<th>106 mid</th>
<th>63 high</th>
</tr>
</thead>
<tbody>
<tr>
<td>At least 1</td>
<td>96%</td>
<td>94%</td>
<td>92%</td>
<td>98%</td>
</tr>
<tr>
<td>NA</td>
<td>75%</td>
<td>68%</td>
<td>69%</td>
<td>90%</td>
</tr>
<tr>
<td>Bool/not</td>
<td>75%</td>
<td>68%</td>
<td>69%</td>
<td>90%</td>
</tr>
<tr>
<td>DK</td>
<td>32%</td>
<td>14%</td>
<td>37%</td>
<td>40%</td>
</tr>
<tr>
<td>NS</td>
<td>48%</td>
<td>64%</td>
<td>44%</td>
<td>33%</td>
</tr>
</tbody>
</table>

Data: Which Concepts?
Life, the universe, and ragbags

• Ragbags used frequently, dominated by abstract concepts.
• Ragbag categories are used for different purposes.
• Low performers less likely to place concepts into don’t know. (Murphy/Tenenberg, do computer science students know what they know?) How do we help them see their confusion?
• Dichotomous pairs: a problem and opportunity?
• Yet again, programming language and paradigm are not the issue.