Comparing Approaches for Learning Abstraction and Automation by Object Orientation
ISSEP 2019 Larnaca

Arno Pasternak, Johannes Fischer
TU Dortmund, Fakultät für Informatik

November, 18, 2019

Research Approach

Teaching experiment

Introduction to **Object-Oriented** Modeling and Programming

- **First:**
  Theoretical thoughts about **Modularization** and its implementation in the classroom.
- **Second:**
  The **implementation** in two ways:
  - **large** projects
  - **small** projects

Computational Thinking

Jeannette Wing 2006

Computational thinking

is a fundamental skill for everyone, not just for computer scientists.

C.T.: Abstraction and Automation [Wing, 2008]

Two A's to C.T. Combined

- Computing is the automation of our abstractions
  - They give us the audacity and ability to scale.
- Computational thinking
  - choosing the right abstractions, etc.
  - choosing the right "computer" for the task

CT&TC 9 Jeannette M. Wing

Abstraction

Abstraction of Data

A **module** describes data from reality,
we are talking about a **abstract data structure**
or a **abstract data type**

[Pomberger, 1984, p.85ff,p.154ff]

Abstraction of Operations

A systematic abstraction of the processes
leads to **structured programming**.

Abstraction

Abstraction in the classroom

Both forms of abstraction (data and operations) must be taught
in computer science classes
equally.

Pure Process Abstraction:

Exclusive abstraction of operations:
De facto variant of **Mathematics.**

Pure Data Abstraction:

Exclusive abstraction of data:
Description of a **static world**:, **Arts**.

Abstraction

Niklaus Wirth

„One more word about object orientation: [. . .]
You learn **new programming techniques** – but on the ground of what you've brought **with you**."

[Wirth, 1991, p.60,61]

Abstraction

Object Oriented Technology in Education

- Today:
  Mainly object-oriented languages in class.
- Consequence:
  Learners do not realize that **modeling** and programming do **not require object orientation.**

Automation

Automation Means:

- Starting position is the **abstract description**
- A **program** must be created in a **concrete programming language**.
- **Executable** on a **machine**.

Programming Means:

- Use a**programming language** containing certain **data types and flow structures**.
- **Creation** of a program is **by no way trivial**.
- **Programming** and **coding is not a low level task.**

This results in a Modelling Gap

- Between the **abstract model** and **concrete program** exists a **gap** .

Modeling Gap

'The students
• identify objects during the analysis of simple problems, their properties, their operations and their relationships,

Modeling Gap

'The students

- identify objects during the analysis of simple problems, their properties, their operations and their relationships,
- model classes with their attributes, their methods and their association relations,

Modeling Gap

'The students

- identify objects during the analysis of simple problems, their properties, their operations and their relationships,
- model classes with their attributes, their methods and their association relations,
- arrange attributes, parameters, and returns of methods simple data types, object types, or linear data collections for'

Modeling Gap

'The students
• identify objects during the analysis of simple problems, their properties, their operations and their relationships, • model classes with their attributes, their methods and their association relations,
• arrange attributes, parameters, and returns of methods simple data types, object types, or linear data collections for'

Modeling Gap

'The students
• identify objects during the analysis of simple problems, their properties, their operations and their relationships, • model classes with their attributes, their methods and their association relations,
GAP
• arrange attributes, parameters, and returns of methods simple data types, object types, or linear data collections for'

(after: Core curriculum for the higher secondary school in NRW(Germany) [MSW NRW, 2013, p.21])

The Modeling Gap

We know:

Implementation is very difficult!

Jump over the Gap

We have two choices:

- **From Modeling to Programming**
- **From Programming to Modeling**

School Practice

If it should be **object orientation, how** do you start?

Teaching Approach: OO-Guidelines (Kölling/Rosenberg)

- **Object orientation** (in Java) is for **large program systems** very well suitable.
- Start **in school also with these ideas**
- The **overhead** of OO is smaller the bigger the system is.
- Start with a **big** system.

Teaching Approach: PRIMM (Sentance/White)

Similar to OO-Guidelines:

- **Predict:** summarise the code
- **Run:** execute code
- **Investigate:** explain, trace, annotate, debug
- **Modify:** edit program
- **Make:** design a new program
- The **lesson should start with a small system**

Sounds really good, we will do so — both!

School Practice

Courses

- **BIG:** Basis of a **larger project** according to the ideas of **Kölling and Rosenberg**.
- **SMALL:** Basis of a **small project** according the ideas of **PRIMM**.

The **intended learning outcomes** of the two groups were **the same**.

School Practice

The Experiment

Simulation of a card game in different versions:

- **SMALL: 65 LOC, 1 class** (cardstack)
- **BIG: 190 LOC, 3 classes** (card,cardstack,player)

Student Activities:

- These program texts have been **read**, ...
- **tested**, ...
- slightly **changed** ...
- **new instances** have been created, ...
- single **operations** specifically modified

Composition of the Groups

- **Course BIG** had **17 students**.
- **Course SMALL** had **27 students**.
- Important: **Each student** could work on their **own computer** in a computerlab with 30 computers.

- Big **challenge**: Meeting of demands of all students.
- This was for the **smaller group (BIG) an advantage**.

Learning Effects?

But what did the students

in fact learn?

We tested the students.

Empirical Results

Questions

- Describe what is meant by the technical term 'array'
- Explain the technical term **'class'**
- Explain the technical term **'attribute'**
- Explain the technical term **'operation'** or **'method'**
- Explain the term **'object'**
- Explain the term **'abstract class'**
- Explain the term **'abstract datatype'**
- Give an example of how a **car class** could be **modeled** in a programme for a car sale
- Describes what is meant by **object-oriented programming**

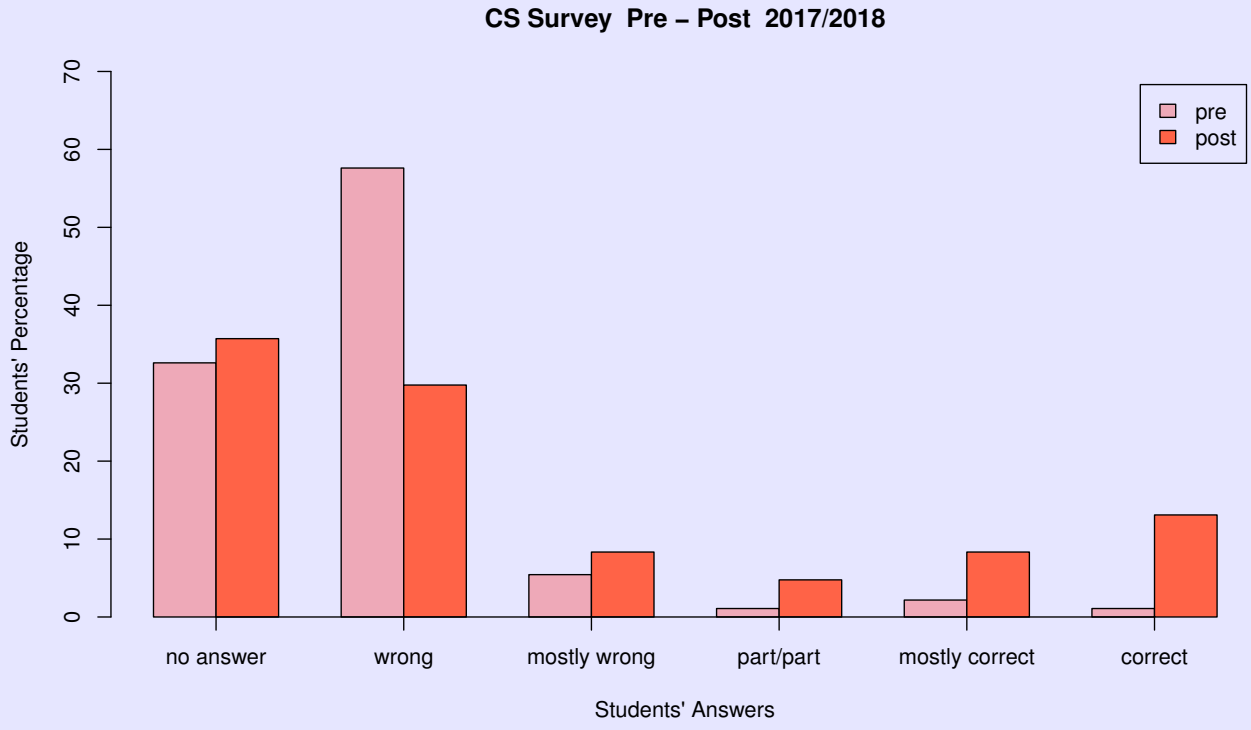
Empirical results

Example Comments by Students

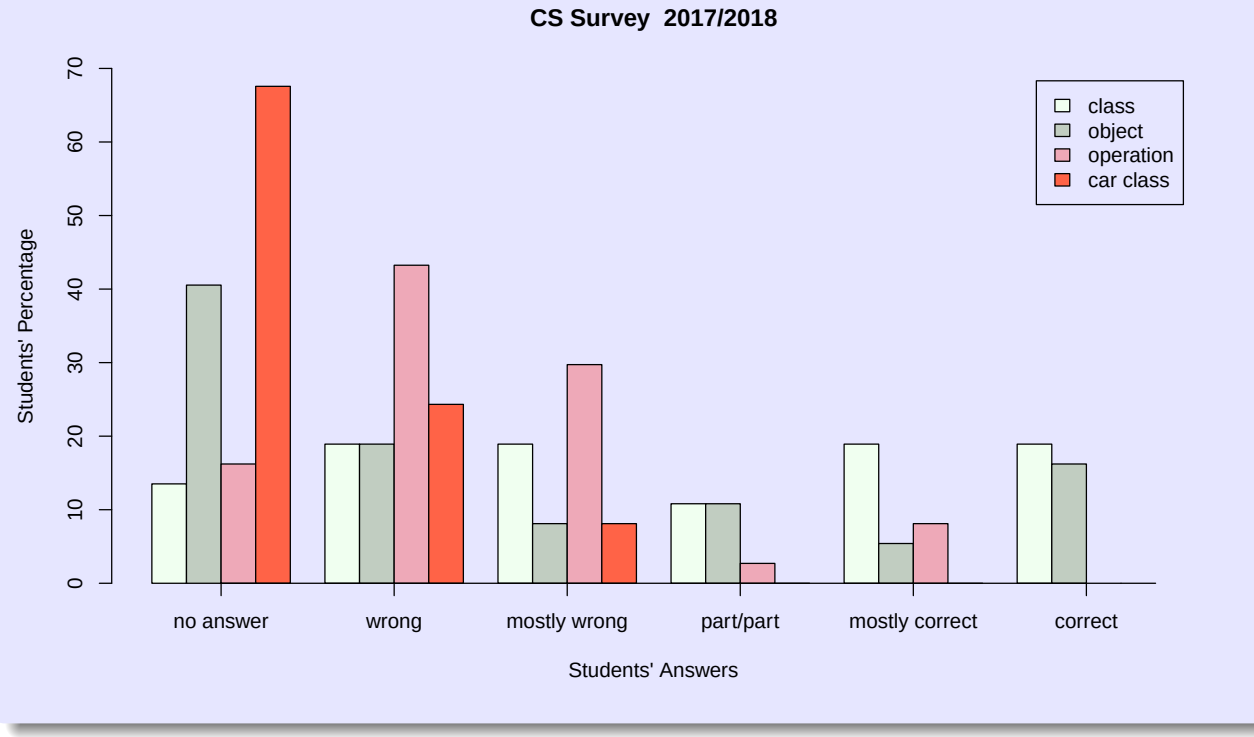
OPERATION/METHOD

- Instruction or procedure
- A calculation.
- The operation is also called an instance.
  And is simply something like `int`.
- If you've broken a leg, you need an operation.
- Operations are given in the form of procedures and functions.

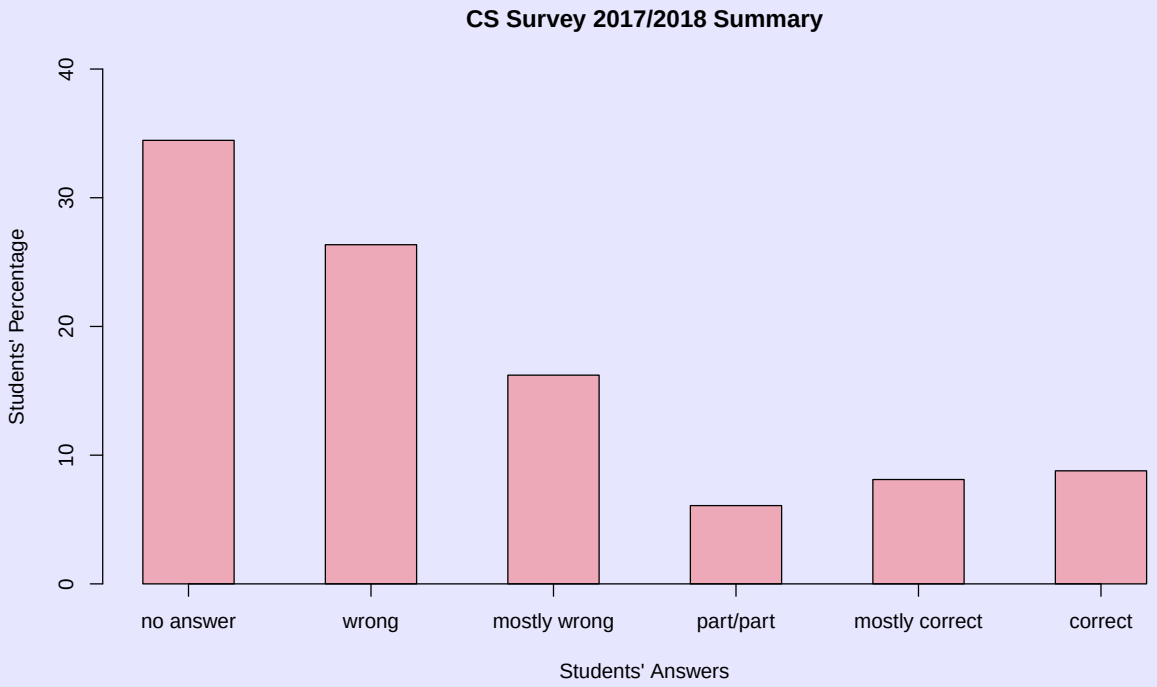
Summary of Student Answers PRE-POST



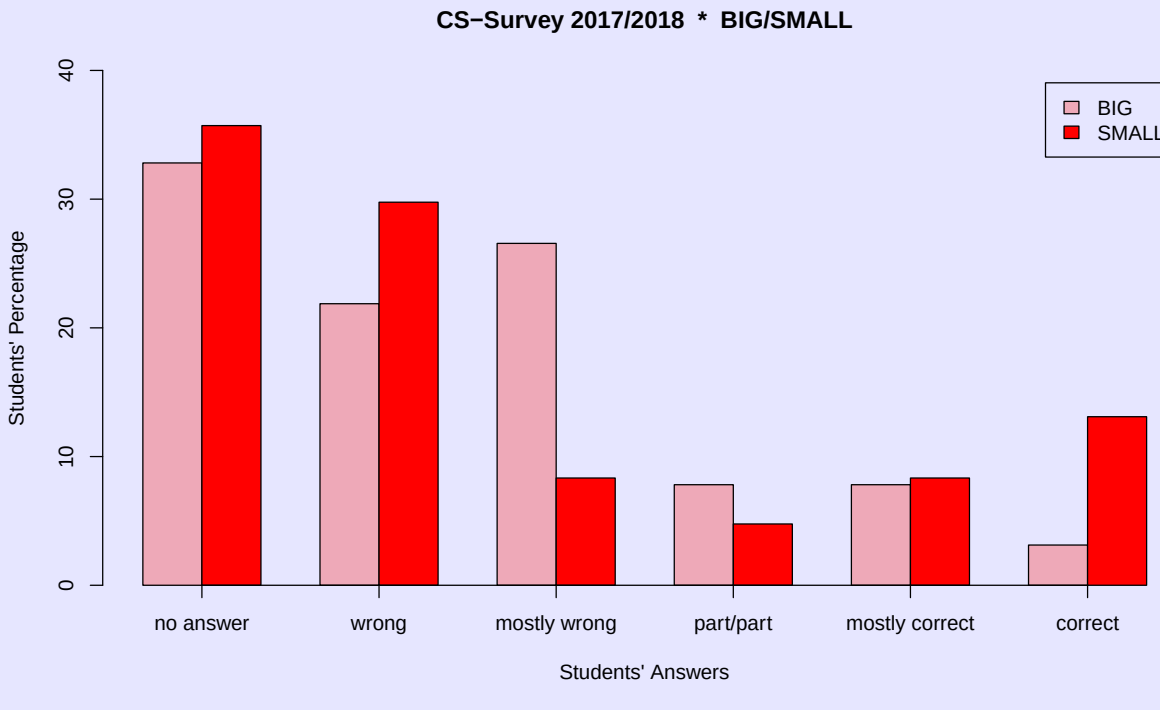
Student Answers 4 Questions



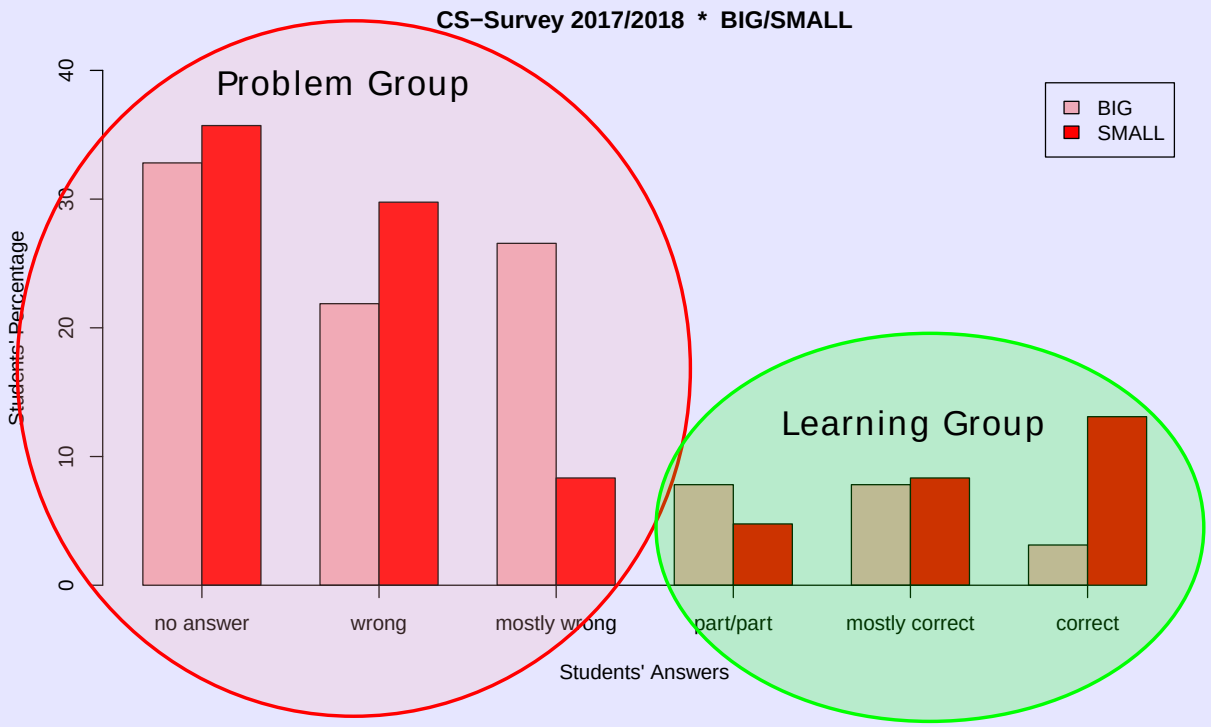
Student Answers Questions Summary



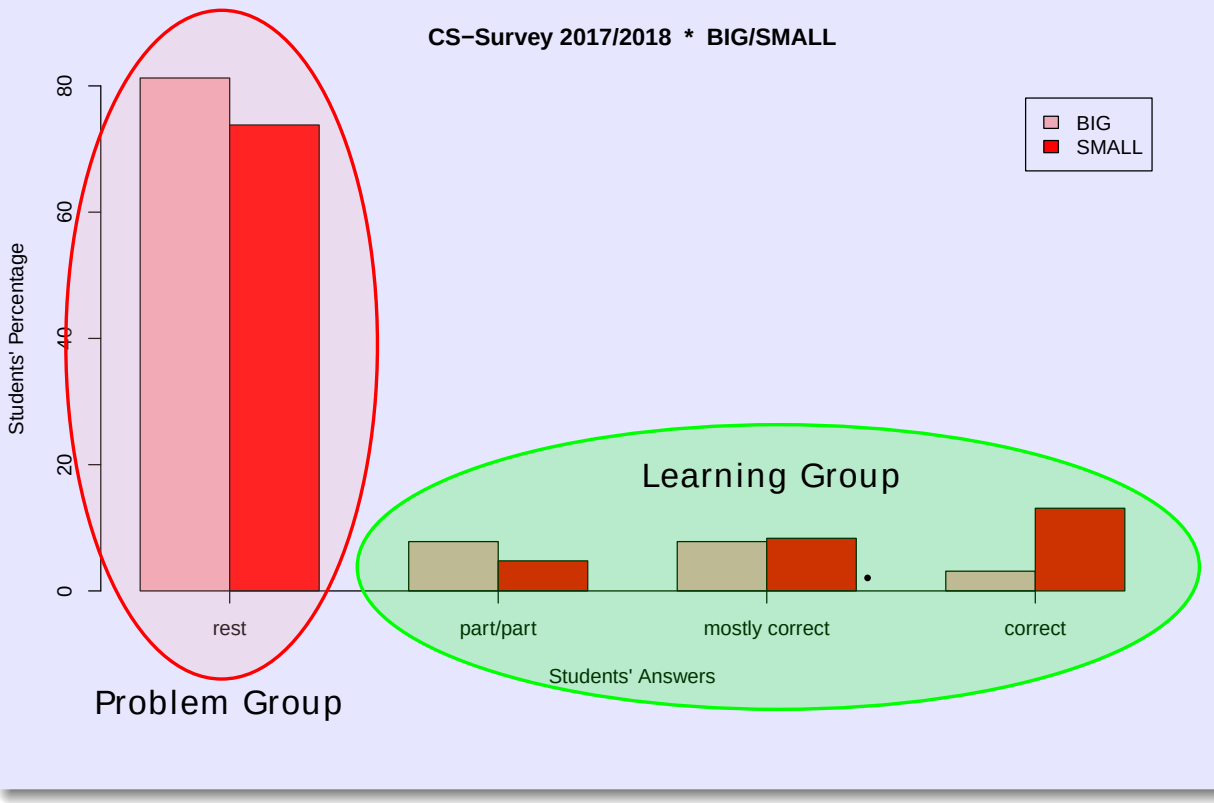
Student Answers SMALL versus BIG



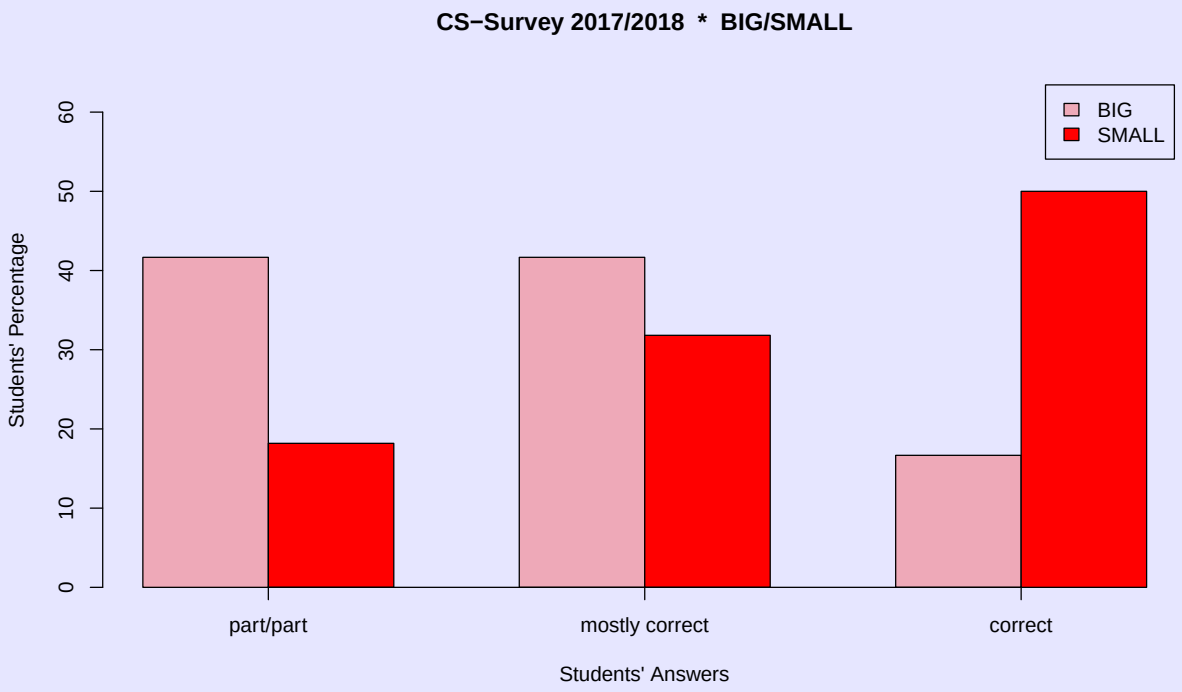
Student Answers SMALL versus BIG



Student Answers SMALL versus BIG



Student Answers SMALL versus BIG



Effectsize *d* BIG – SMALL

Effect size	all students
post	0.09

Effect size *d* SMALL - BIG ** *Summary OO - Items*

Effectsize *d* SMALL – BIG

Effect size	learning group
post	0.70

Effect size *d* SMALL - BIG ** *Summary OO - Items*

Learning Effects?

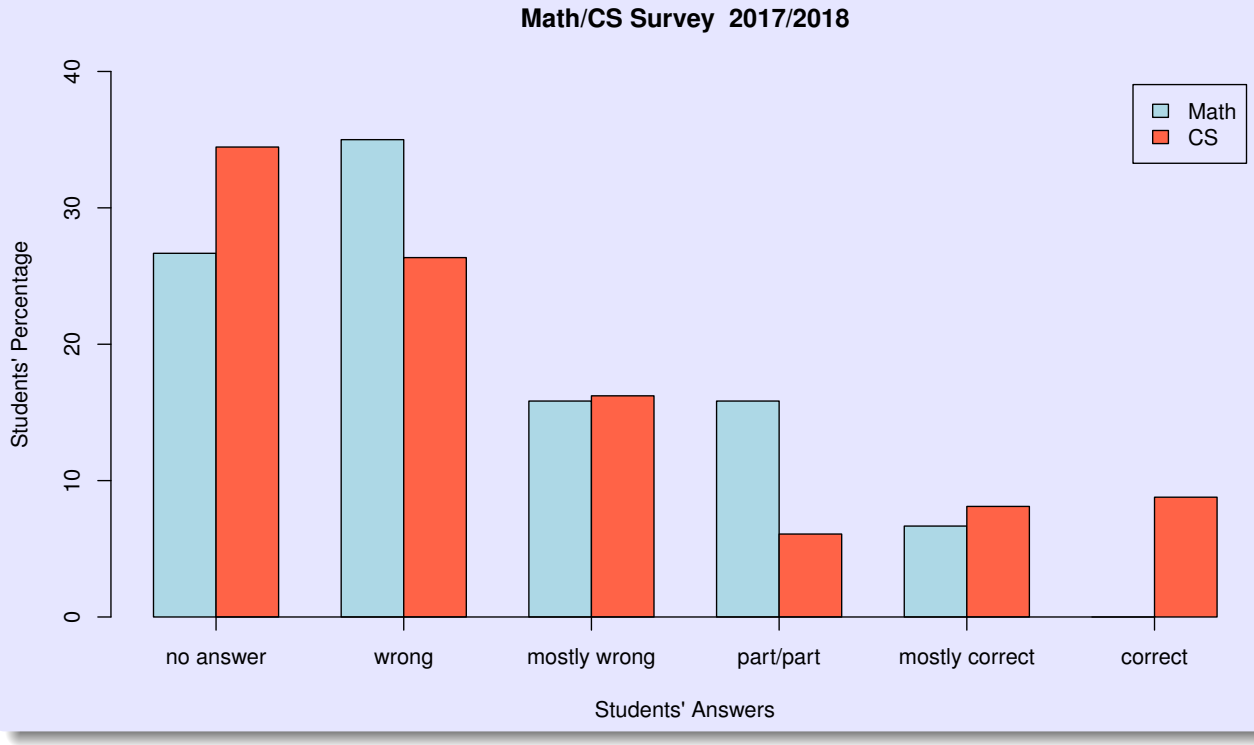
Did the students
Learned anything?

Yes, and really quantifiable!
Unfortunately, many of them have learned very little!

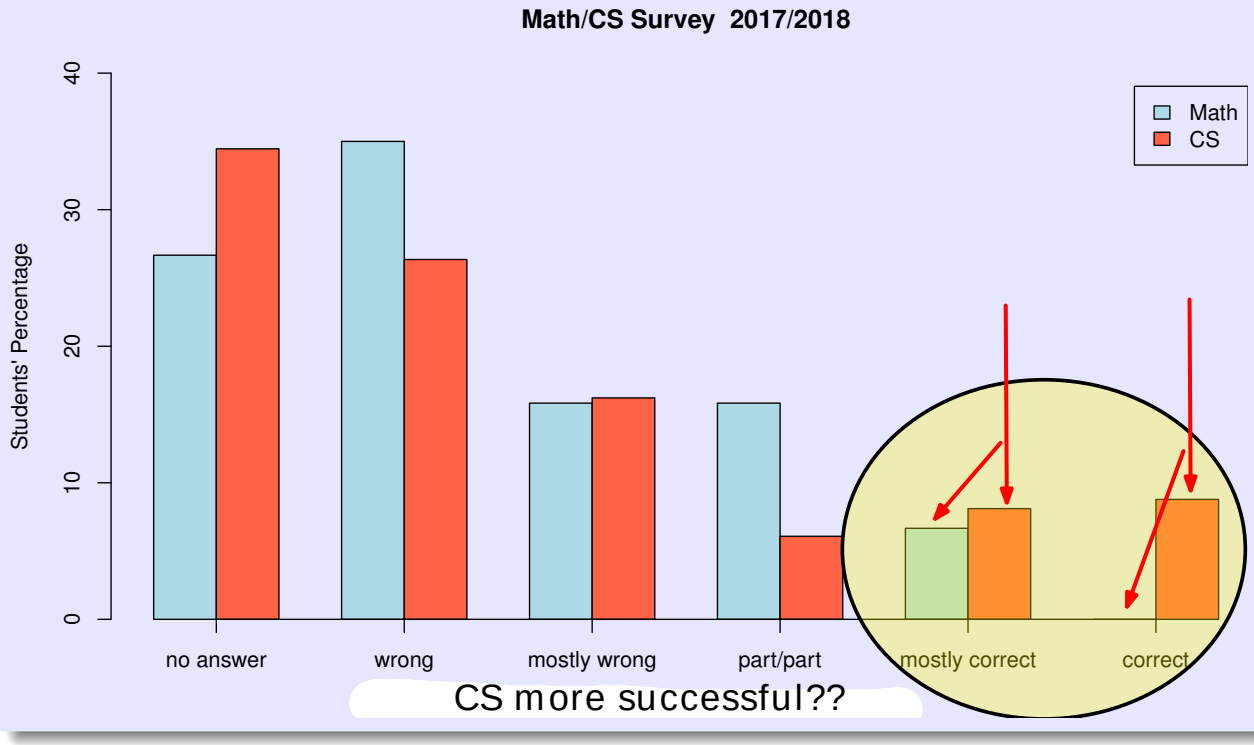
We wanted to know exactly!!

We asked students
in Math
and they tends to be similarly bad!

CS and Math Survey in Comparison



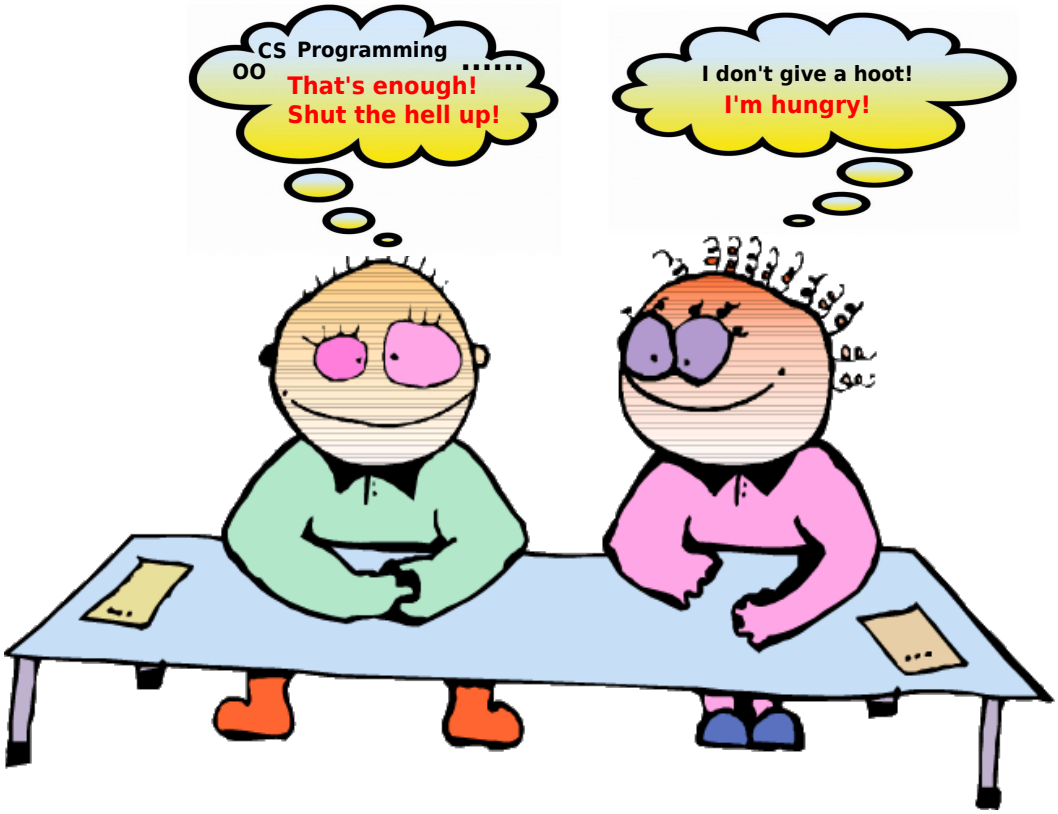
CS and Math Survey in Comparison



There must be a reason!

(Outlook) Questions

- Do we expect **too much** at the same time?
- Items positively **tested in exams:**
  Quickly forgotten?
- Special in CS:
  Big or small **projects**:
  Both are already **too much**?
- Following:
  Are **small steps** the **better approach**?



Bildquellen:
http://www.sks-tierpark.bildung-lsa.de/1_unsere_schule/schueler/cliparts/zwei_tisch.gif, 23.3.2017
http://www.publicdomainpictures.net/pictures/70000/nahled/thought-cloud.jpg, 23.3.2017

MSW NRW (2013).
Kernlehrplan für die Sekundarstufe II, Gymnasium/Gesamtschule in Nordrhein-Westfalen, Informatik.
Düsseldorf.

Pomberger, G. (1984).
Softwaretechnik und Modula 2.
Hanser Fachbuchverlag, München.

Wing, J. M. (2008).
Computational thinking and thinking about computing.
Philosophical Transactions of The Royal Society, (366):3717–3725.

Wirth, N. (1991).
Die dinge beim namen nennen.
In *Erstes European Software Festival*, pages 60–64, Würzburg. Chip – Vogel-Verlag.