



UNIVERSITÀ DEGLI STUDI DI MILANO - BICOCCA
Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di Laurea Magistrale in Informatica



Università
della
Svizzera
italiana

Qualitative analysis of Mastery Checks in a programming course

Relatore USI: Prof. Matthias Hauswirth
Relatore UniMiB: Prof. Giovanni Denaro

Laureando:
Luca Chiodini

Soloway, 1986

Write a program that will read in integers and output their average.
Stop reading when the value 99999 is input.

Students' performance on Rainfall problem after CS1

Students' performance on Rainfall problem after CS1



Soloway et al. 1980s

17% correct

83% wrong

Students' performance on Rainfall problem after CS1



Soloway et al. 1980s

17% correct
83% wrong



McCracken et al. 2000s

21% correct
79% wrong

- Interdisciplinary research area
 - Pedagogy, learning and cognitive sciences
 - Core CS fields (e.g., programming languages)

- Interdisciplinary research area
 - Pedagogy, learning and cognitive sciences
 - Core CS fields (e.g., programming languages)
- Find better *means* and *strategies* to enable students to master CS topics

- Interdisciplinary research area
 - Pedagogy, learning and cognitive sciences
 - Core CS fields (e.g., programming languages)
- Find better *means* and *strategies* to enable students to master CS topics
- ACM SIGCSE conferences (ITiCSE, ICER)

We want to understand:

- which **misconceptions** students develop
- which **strategies** are used to tackle problems
- how **learning trajectories** evolve over time

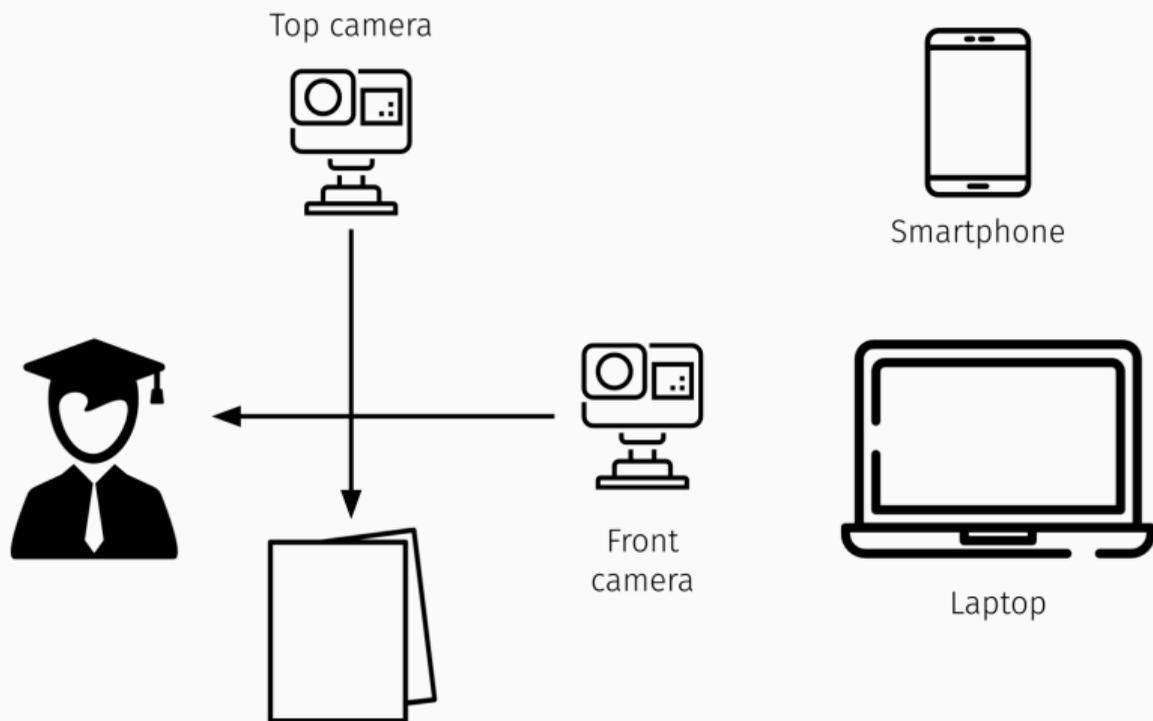
We want to understand:

- which **misconceptions** students develop
- which **strategies** are used to tackle problems
- how **learning trajectories** evolve over time

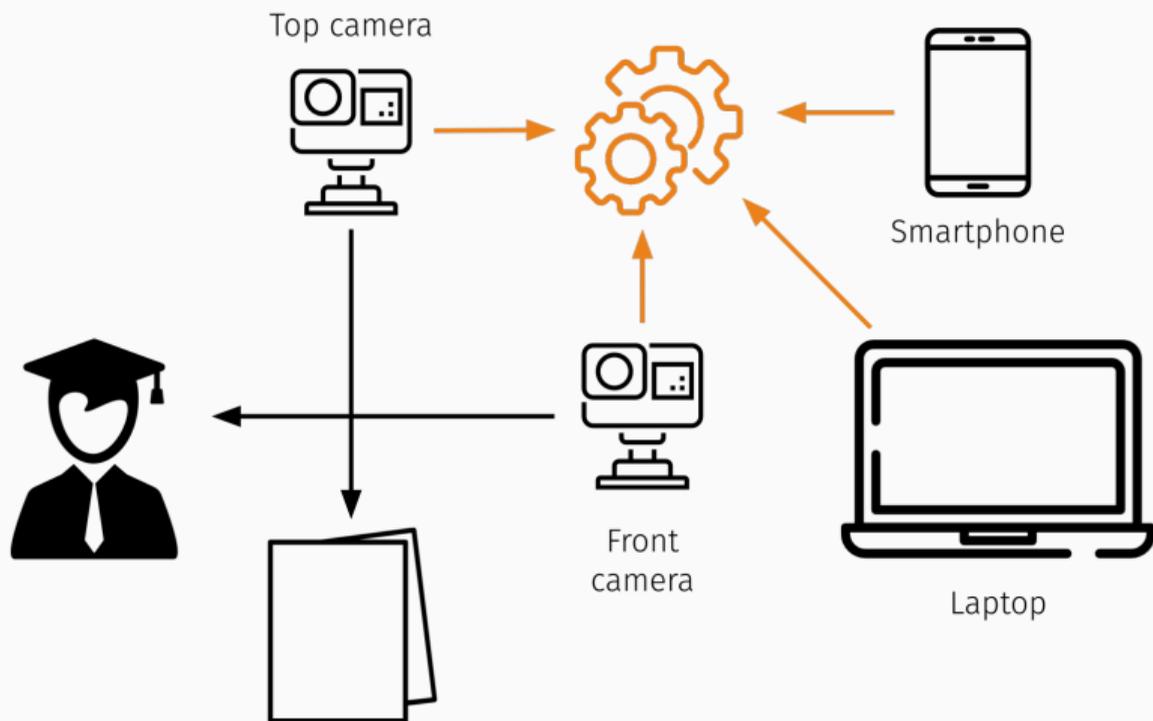
Study outline:

- recruited 6 first-year students attending Programming Fundamentals 2
- held and recorded 10 individual Mastery Check sessions (roughly 30')
- collected over 1600 minutes covering a wide range of Java topics

Tech setup

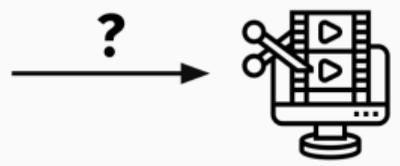


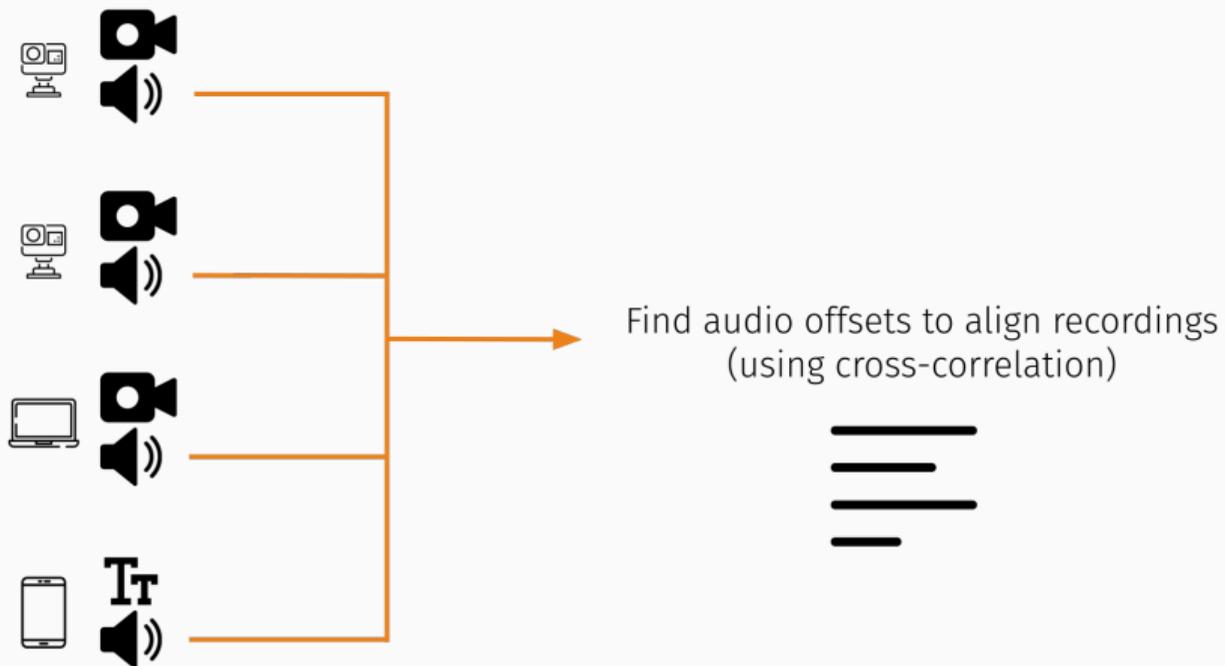
A new tool to sync everything automatically...



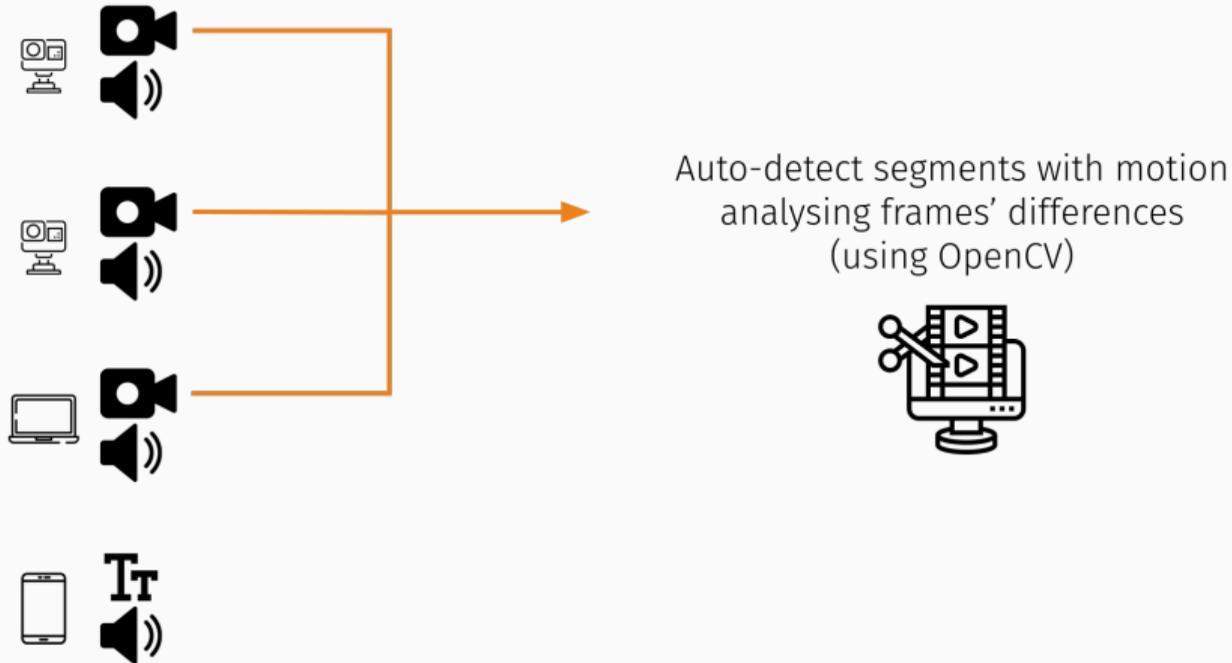


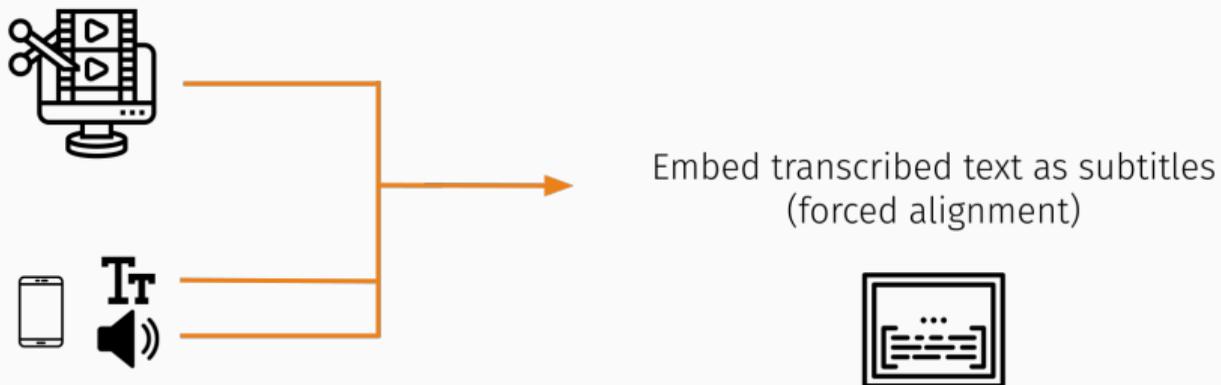
Huge amount of
manual tedious work





Pipeline





Final edited video

The screenshot shows the BlueJ IDE interface. The main window displays the source code for the `Calculator` class. The code is as follows:

```
1 public class Calculator {  
2  
3     private double value;  
4  
5     public Calculator(){  
6         this.value = 0;  
7     }  
8  
9     public void add(double x){  
10        this.value = x + value;  
11    }  
12  
13    public void clear(){  
14        this.value = 0;  
15    }  
16  
17    public double get(){  
18        return this.value;  
19    }  
20  
21    public void set(double y){
```

On the right side, the class diagram shows a `Node` class with a `Calculator` class as a field. Below the diagram, a snippet of code is visible:

```
second = new Node(4, null);  
first = new Node(1, second);  
countAcc(0)  
(int)
```

At the bottom of the image, there is a white text overlay with a black outline that reads: "Yeah, if I have value here, I could use I have to use this." The text is partially obscured by a blurred desktop background.

The screenshot displays the MAXQDA Analytics Pro 2020 interface. The top menu includes Home, Import, Codes, Memos, Variables, Analysis, Mixed Methods, and Visual. Below the menu is a toolbar with icons for New Project, Open Project, Document System, Code System, Document Browser, Retrieved Segments, Logbook, and Team.

The main workspace is divided into three panels:

- Document System:** A tree view showing a hierarchy of documents. Under 'Documents', there are three folders: 'Week1' (4'007), 'Week2' (443), and 'Week3' (592). Each folder contains multiple 'Session' documents. The 'Week1Sessio' folder is expanded, showing documents like 'Week1Sessio' (80), 'Week1Sessio' (82), 'Week1Sessio' (86), 'Week1Sessio' (97), and 'Week1Sessio' (92).
- Code System:** A tree view showing a hierarchy of codes. Under 'Code System', there is a folder 'MetaInfoAboutQuestionsAndAnswers' (4'007) containing codes like 'Question2ndTime' (6), 'SaysUncertaintyExplicitly' (71), and 'DoesntKnowConcept' (9).
- Video Player:** A video player showing a person's hands writing on a piece of paper. The video is paused at 0:20:22.8. Below the video is an audio waveform and a timeline with markers at 0:20:10, 0:20:12, 0:20:14, 0:20:16, 0:20:18, 0:20:20, 0:20:22, 0:20:24, and 0:20:26.

At the bottom of the video player, there are three horizontal bars representing different code segments: 'ThisObjectNotExistingInStackAndHeapDiagram' (orange), 'RecognisesAndFixesError' (blue), and 'Correct' (green).

200+ codes divided into 9 macro categories

200+ codes divided into 9 macro categories

- MISCONCEPTIONS
Specific wrong conceptions about syntax or semantics of Java.

200+ codes divided into 9 macro categories

- MISCONCEPTIONS
Specific wrong conceptions about syntax or semantics of Java.
- STRATEGYERRORS
Bad processes used by novices to solve tasks.

200+ codes divided into 9 macro categories

- MISCONCEPTIONS
Specific wrong conceptions about syntax or semantics of Java.
- STRATEGYERRORS
Bad processes used by novices to solve tasks.
- METAINFORMATIONABOUTERRORS
Additional information about errors
(inconsistencies, patterns, novelty, ...)

200+ codes divided into 9 macro categories

- MISCONCEPTIONS
Specific wrong conceptions about syntax or semantics of Java.
- STRATEGYERRORS
Bad processes used by novices to solve tasks.
- METAINFORMATIONABOUTERRORS
Additional information about errors (inconsistencies, patterns, novelty, ...)

4000+ video segments coded

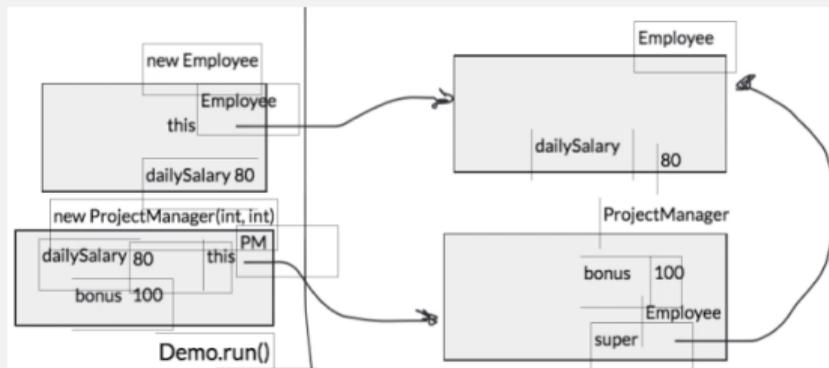
SUPERCLASSOBJECTISALLOCATED

Title	SuperclassObjectIsAllocated
Context	Class Child extends class Parent
Description	When new Child() is executed, two objects are created: a Child object with the fields that belong to the class Child and a Parent object with the fields that belong to the class Parent .
JLS	§8.2 Class Members
Observations	Sessions 8 and 10

Example of a misconception

SUPERCLASSOBJECTISALLOCATED

```
public class Empolyee {  
    private int dailySalary;  
    ...  
}  
public class ProjectManager extends Empolyee {  
    private int bonus;  
    ...  
}
```



- Some misconceptions are one the **dual** of another
 - Duality in the type system: `IMPLICITNARROWING` vs `NOIMPLICITWIDENING`
 - Duality in “collection” types: `ARRAYHASLENGTHMETHOD` vs `STRINGLENGTHFIELD`

- Some misconceptions are one the **dual** of another
 - Duality in the type system: `IMPLICITNARROWING` vs `NOIMPLICITWIDENING`
 - Duality in “collection” types: `ARRAYHASLENGTHMETHOD` vs `STRINGLENGTHFIELD`
- Some misconceptions are caused by **wrong analogies**
 - `ARRAYLISTELEMENTACCESSUSINGSQUAREBRACKETS`
 - `ARRAYSHAVECONSTRUCTOR`

- Some misconceptions are one the **dual** of another
 - Duality in the type system: `IMPLICITNARROWING` vs `NOIMPLICITWIDENING`
 - Duality in “collection” types: `ARRAYHASLENGTHMETHOD` vs `STRINGLENGTHFIELD`
- Some misconceptions are caused by **wrong analogies**
 - `ARRAYLISTELEMENTACCESSUSINGSQUAREBRACKETS`
 - `ARRAYSHAVECONSTRUCTOR`
- Tackling a problem the right way is hard
 - `THINKINGALGOCOMPLEXITYBEFORESIMPLEANDCORRECT`
 - `MISSINGBASECASEINRECURSION`
 - `MISSINGRETURNINRECURSION`
 - `NOTRELYINGONINDUCTIONINRECURSION`

Insights from coded segments on Notional Machines

Notional Machine (Fincher et. al., 2020)

A notional machine is a pedagogic device to assist the understanding of some aspect of programs or programming.

Insights from coded segments on Notional Machines

Notional Machine (Fincher et. al., 2020)

A notional machine is a pedagogic device to assist the understanding of some aspect of programs or programming.

- NOTIONALMACHINEHELPSRECOGNIZINGERROR
- NOTDOINGSTEPSINORDERINSTACKANDHEAPDIAGRAM

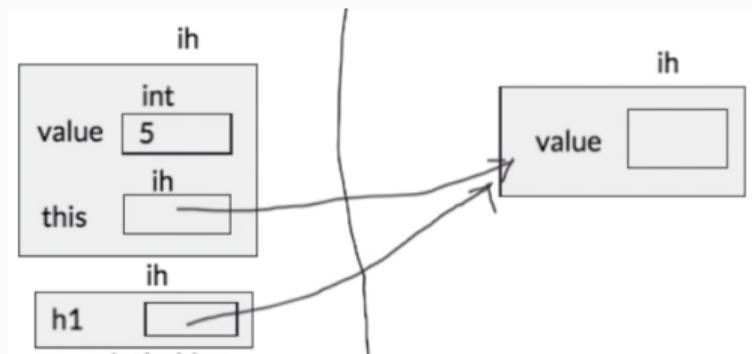
Insights from coded segments on Notional Machines

Notional Machine (Fincher et. al., 2020)

A notional machine is a pedagogic device to assist the understanding of some aspect of programs or programming.

- NOTIONALMACHINEHELPSRECOGNIZINGERROR
- NOTDOINGSTEPSINORDERINSTACKANDHEAPDIAGRAM

```
IntHolder h1 = new IntHolder(5);
```



	P1	P2	P3	P4	P5
Session 6	Correct	Wrong	N/A	Wrong	N/A
Session 8	Wrong	Wrong	Wrong	Wrong	Wrong
Session 9	Wrong	Wrong	Wrong	Correct	Wrong
Session 10	Wrong	Wrong	Wrong	Wrong	Wrong

Table 1: Correctness of THIS EXISTS IN STATIC METHOD across four sessions.

	P1	P2	P3	P4	P5
Session 6	Correct	Wrong	N/A	Wrong	N/A
Session 8	Wrong	Wrong	Wrong	Wrong	Wrong
Session 9	Wrong	Wrong	Wrong	Correct	Wrong
Session 10	Wrong	Wrong	Wrong	Wrong	Wrong

Table 1: Correctness of THIS EXISTS IN STATIC METHOD across four sessions.

	P1	P2	P3	P4	P5
Session 8	Correct	Correct	Wrong	Wrong	Wrong
Session 10	Correct	Correct	Correct	Correct	Correct

Table 2: Correctness of SUPERCLASS OBJECT IS ALLOCATED across two sessions.

Conclusions and follow-up studies

Exploratory phase:

- Developed a useful tool for all qualitative research studies
- Added 100+ newly uncovered Java misconceptions
- Initial attempt to understand learning trajectories
- Observed solving strategies looking at the way students produce artifacts

Conclusions and follow-up studies

Exploratory phase:

- Developed a useful tool for all qualitative research studies
- Added 100+ newly uncovered Java misconceptions
- Initial attempt to understand learning trajectories
- Observed solving strategies looking at the way students produce artifacts

Ideas for possible teaching improvements (subject of future targeted studies):

- Make teachers aware of common misconceptions
- Prepare assessments to detect misconceptions
- Classify misconceptions and strategies across different programming languages
- Know which interventions successfully solve an existing issue