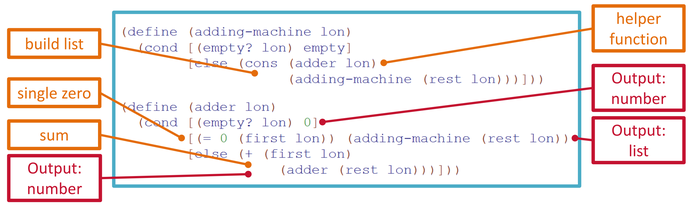
Developing Narratives on Students' Program Planning

Francisco Castro, fgcastro@cs.wpi.edu
WPI Computer Science

Advisor: Kathi Fisler, kfisler@cs.wpi.edu

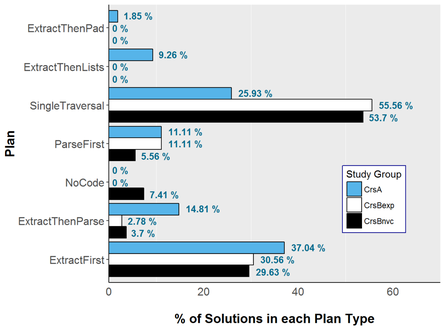
Observations from prior studies

Adding Machine – Input: [1, 2, 0, 7, 0, 5, 4, 1, 0, 0, 6] **Output:** [3, 7, 10]



Students struggled to develop working solutions, decomposing the problem **on-the-fly** rather than planning ahead; they were unable to figure out how to adapt a familiar process to a more complex problem [1]

Data Smoothing – Input: [95, 102, 98, 88, 105] **Output:** [95, 98.33, 96, 97, 105]



- Students were able to produce solutions for single-task problems after CS1
- Many students struggled to write multiple-task programs, even after initial lectures on such problems

What distinguishes those who can write multiple-task programs from those who can't?

Key observation: Not all learners are able to make use of program design principles effectively, even after interventions directed at addressing how to adapt techniques to more complex contexts

Study Concept & Research Questions

Goals: Explore how students plan programs and the factors that influence their decisions; develop **qualitative narratives** about how students progress in programming and planning

Method: Conduct a series of **interviews** and **think-aloud sessions** with students through their first two CS courses and follow their progress throughout these courses

Research questions:

- **Development of knowledge:** What factors and design principles do students draw on and use in their narratives of their programming processes?
- **Development of process:** In what ways do students' design choices evolve as they progress through their courses?
- **Management of learning:** What metacognitive processes or strategies do students articulate in their narratives and in what ways do these interact with their programming processes?

User Study

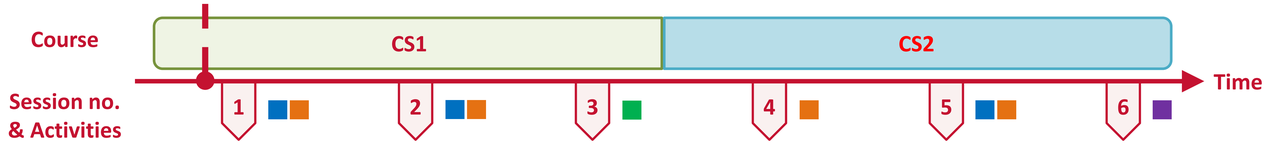
Participants (N=13)

Exam 1 Grade	A (90-100)	B (80-89)	C (65-79)	Mean (grade)
Male	3	2	1	87.67
Female	3	1	3	82
# Students	**6**	**3**	**4**	**84.62**
Mean (grade)	93.67	85.67	70.25	

Reported prior experience

- High school class (7)
- Online course (3)
- Self-study (4)
- Boot camps/workshops (1)
- Programming clubs (1)
- AP class (3)
- None (2)

- Java (5)
- Python (4)
- C++ (3)
- JavaScript (3)
- PHP (2)
- Racket (1)
- Ruby (1)



Session activities

- Review homework solutions
- Compare/rank code solutions
- Write a code solution
- Sketch a solution

Qualitative Analysis

Verbal protocols (+ user artifacts)

Directed interviews and/or **think-aloud protocols** [3] are conducted in each session to elicit student responses, reflections, and insight into cognitive processes during task performance.

Verbal transcripts

Verbal protocols collected from each session are **transcribed** for analysis.

Coded transcripts

Grounded theory methods [2] are adapted for coding verbal transcripts.

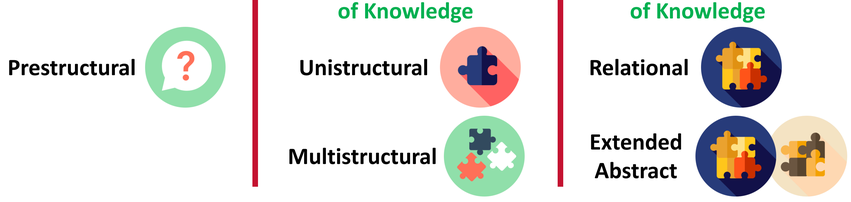
Coding involves attaching qualitative labels (codes) to data excerpts to identify *themes*, *categories*, and *summaries* for developing an emergent theory about the data.

Taxonomic mapping

A **semi-closed card sort** is adapted to classify the codes using the **SOLO taxonomy** [4] and the emergent themes from the qualitative coding process.

SOLO: Structure of Observed Learning Outcomes

Describes five (5) levels of increasing complexity in understanding of a discipline or subject area; increasing complexity of connections



Taxonomic Mapping

SOLO Category	Theme: Functions & Use of expressions [Exemplars]
Prestructural *Lack of knowledge*	*So "or" isn't always the first thing I go to, just because I'm not too familiar with it.*
Unistructural *About a single expression*	*if the list is empty, then it should return zero, because the output should be a number*
Multistructural *About a series of expressions*	*And so if that's false, it goes onto the next one where it asks if it's cons, and if it's true, it goes on to see if the name that was given in the function matches the first list of strings – so the first ad, the name in there, if that's true, then it adds the air cost*
Relational *Interactions among multiple expressions*	*So, that list of ads is then acted on by this function, total-cost-for-aloa, which takes a list of ad. It […] produces a number, so this list that I've just created is now acted on by a function that takes a list and creates a number*

References and Acknowledgement

[1] F. Castro and K. Fisler. 2016. On the Interplay Between Bottom-Up and Datatype-Driven Program Design. SIGCSE '16. ACM, 205–210.
[2] P. Kinnunen and B. Simon. 2012. Phenomenography and grounded theory as research methods in computing education research field. Computer Science Education 22, 2, 199–218.
[3] J. Nielsen, T. Clemmensen, and C. Yssing. 2002. Getting Access to What Goes on in People's Heads?: Reflections on the Think-aloud Technique. NordiCHI '02. ACM, 101–110.
[4] J. Whalley et al. 2006. An Australasian Study of Reading and Comprehension Skills in Novice Programmers, Using the Bloom and SOLO Taxonomies. ACE '06.

We thank the students who participated in the study. This research is partially funded by the US NSF.
Icons were made by Freepik, Vectors Market from www.flaticon.com.