Computer Science Teaching Knowledge: A Framework and Assessment

Aleata Hubbard, *WestEd*
Yvonne Kao, *WestDd*
Contact: ahubbar@wested.org

Educators, researchers, politicians, tech companies, and others continue to advocate for the importance of K-12 students learning computer science in our increasingly tech-driven society. One way school districts in the United States address this growing demand is by allowing teachers certified in other disciplines to lead computer science courses. Summer and weekend professional development opportunities support these educators in developing the expertise needed for effective computer science teaching, but a great portion of their learning to teach computer science will occur through on-the-job experiences. Our four-year NSF EHR grant explores how a job-embedded professional development program that pairs high school teachers with tech industry professionals supports educators in acquiring computer science teaching knowledge. The research presented in this poster focuses on the third year of the study and includes (a) a theoretical component focused on creating a framework to explain on-the-job computer science teaching knowledge development based on case studies with six teachers, and (b) an empirical component focused on the creation and administration of a computer science teaching knowledge assessment. By the time of the SIGCSE symposium, we expect to have pre-test results from the first administration of our teaching knowledge assessment, completed by both high school teachers and their collaborating tech industry professionals. This poster will present our theoretical framework, resultant teaching knowledge assessment with sample items, and analysis of participants' assessment responses and their relationship to specific teaching experiences.

Keywords: Pedagogical content knowledge**;** high school teachers**;** professional development; partnerships

DOI: http://dx.doi.org/10.1145/3017680.3022399

Computer Science Teaching Knowledge: A Framework and Assessment

Aleata Hubbard - ahubbar@wested.org; Yvonne Kao – ykao@wested.org

WestEd
WestEd.org

BACKGROUND

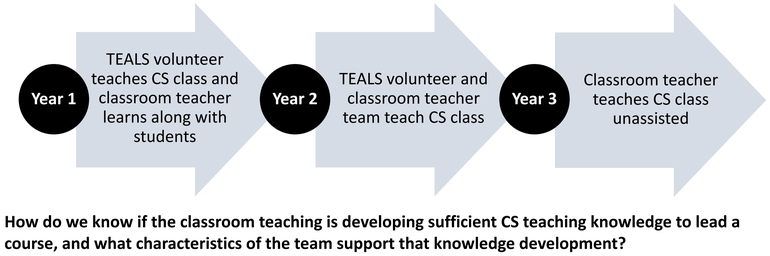
- The majority of K-12 computer science (CS) teachers lack either teaching experience or CS experience (Gal-Ezer & Stephenson, 2010).
- Neither teacher preparation programs nor state teacher certification agencies are equipped to help people become highly qualified CS teachers (Gal-Ezer & Stephenson, 2010).
- The field needs to create alternate ways of preparing CS teachers.
- **TEALS** is a non-profit organization that partners with US high schools to help them launch and grow a sustainable CS program.
- TEALS recruits and trains volunteer CS professionals and pairs them with high school teachers to team teach CS.

TEALS Classes

Introduction to Computer Science	AP Computer Science A
Pre-AP course that teaches students to program using Snap!	Introduces students to the fundamentals of object-oriented programming in Java

TEALS Co-Teaching Team

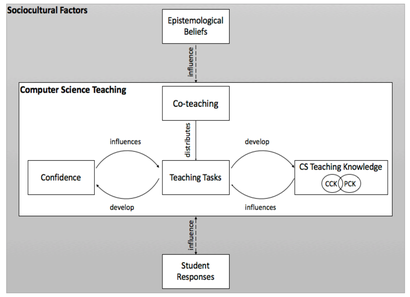
Volunteer	Classroom Teacher
Industry professional with expert knowledge of computer science	High school teacher who may or may not have computer science background



How do we know if the classroom teaching is developing sufficient CS teaching knowledge to lead a course, and what characteristics of the team support that knowledge development?

FRAMEWORK

Our framework, based on 2014-15 pilot data and CS education literature (Baxter, 1987; Bender et al., 2015; Lapidot, 2005), describes how in-service TEALS teachers develop PCK.



- ***Teaching tasks*:** practices related to planning, delivering, and reflecting on instruction
- ***CS teaching knowledge:*** knowledge bases drawn upon in instruction (see below)
- ***Co-teaching:*** the professional development model used in the TEALS program
- ***Confidence:*** belief in one's ability to accomplish teaching tasks
- ***Epistemological beliefs*:** ideas about the nature of teaching and learning
- ***Student responses:*** student reaction to lessons, performance on assessments
- ***Sociocultural factors:*** external factors influencing how teachers work/learn (e.g., district mandates)

CS Teaching Knowledge

There are many conceptualizations of teaching knowledge, but we focus on the following components:

Knowledge Area	Examples
Common Content Knowledge (CCK)	• *Variables create a space for the computer to store information.* • *This is a working definition, I would improve it by saying that the variable's information can be changed.*
Knowledge of Content & Teaching (KCT)	• *Variable is simply a holding place. I refer to it as a bucket and use visuals to demonstrate.* • *If you say "later," you need to get into a discussion of scope because it may not be around later.*
Knowledge of Content & Students (KCS)	• *Learning while being successful at putting together a few lines of code is important for building confidence, especially for students who have never coded before.*
Horizon Knowledge	• *Students understand the concept of variable through mathematics, but in CS they need to also understand how a memory location is created for the variable, what type it is and what its limitations are.*

ASSESSMENT

- Focused on 3 topics in CS that are common to the TEALS Introduction to CS and AP Computer Science A courses: **algorithms**, **variables and assignment**, and **control structures**
- Three questions per topic: **prior experience with teaching tasks**, **teaching content**, and **understanding student thinking**

Prior Experience with Teaching Tasks

Which of the following tasks have you done before today?

- Created original instructional materials about this topic
- Adapted instructional materials on this topic in order to make them more appropriate for a particular class
- Completed readings or other assignments on this topic (either to prepare for a lesson or as a student yourself)
- Delivered a lecture on this topic to a class
- Graded or otherwise evaluated student work on this topic
- Assisted students with learning this topic in one-on-one or small group settings (including tutoring)
- None of the above
- Other, please explain _____

Teaching Content

What is important about this topic? Create or evaluate an explanation of a given CS concept.

- What is most important for your students to learn about _____?
- Is this an accurate definition of ____? If not, what is misleading about it, and how would you improve it?
- What are the difficulties or limitations of using the given exercise to explain the concept?

Understanding Student Thinking

Predict and evaluate student work on this topic.

- What incorrect responses to this question do you expect to see from your students, and why do you think students will make those errors?
- Given a student's response to a code-writing prompt, what does this student understand well about the time? What does the student not understand well? What kind of feedback would you provide to improve the student's understanding of the topic?

PRELIMINARY ANALYSIS

Method

3 researchers independently coded responses to **Topic 2: Variables and Assignment** using the following coding scheme:

- Question 1: type of teaching knowledge (CCK, KCT, KCS, Horizon, missing)
- Question 2: type of teaching knowledge; teacher opinion of definition accuracy (accurate, incomplete)
- Question 3: correct or incorrect
- Question 4: number of responses given; type of responses (identify response, rationalize response)

Questions 1 and 2 could receive multiple codes. Researchers discussed and resolved discrepancies.

Participants (2016-2017)

Course	n	Avg. # Tasks Completed	Avg. Years with TEALS	Avg. Years Teaching CS	# Studied CS
Intro to CS	16	2.80	0.50	0.63	1
AP CS A	13	4.60	1.50	2.25	3

Although AP and Intro teachers have similar years of total teaching experience (~11 years), AP teachers appear to be more experienced with CS. On average, they have done more teaching tasks related to variables and assignment, worked with TEALS longer, and taught CS longer.

Number of Unanswered Items

Response	Question 1	Question 2	Question 3	Question 4
I don't know	2	2	-	3
Missing	1	-	1	4

Q1: What is most important for your students to learn about variables and assignment?
Q2: Is this an accurate definition of variable? If not, what is misleading, how would you improve it?

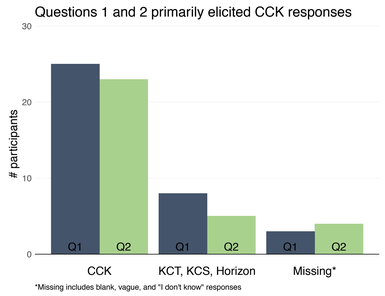
A variable in programming is a name that refers to information the program can use later.

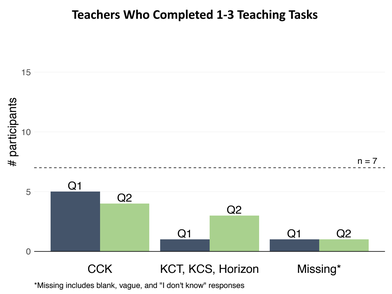
- Most participants (n = 25) said the definition of variable was inaccurate.

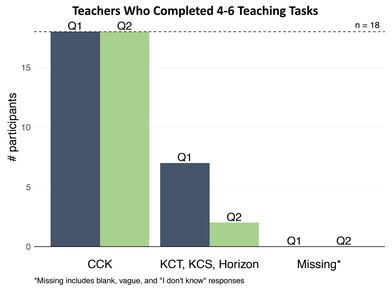
"Not the best. A variable stores information that takes up memory."

"I would say a variable is a programming tool that has a name (so that it can be referred to) and a type (so that you know what kind of data it holds) for storing and accessing data."

- Q1 elicited more examples of KCT and Horizon. Q2 elicited more examples of KCS.
- KCT, KCS, Horizon only mentioned by teachers who completed some related teaching tasks.





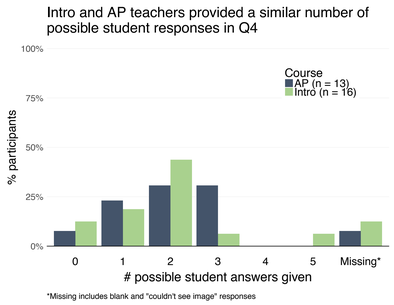


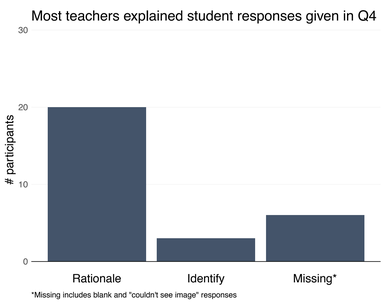
Q3: What does the sprite say when the script is clicked?
Q4: What incorrect responses do you expect to see and why?

- Most participants (n = 24) answered Q3 correctly.
- Participants who answered Q3 incorrectly (n = 2) said they've done 0 or 2 teaching tasks.
- Teachers listed eight different possible incorrect student responses: x, 5, 7, 10, 15, 17, 22, 24.

Students might say 5, because it is the first x they see. They might say "x" because the block says "say x".

5 - doesn't understand how variables update 12 - correct answer 15 - doesn't grasp linear nature of reading code 22 - doesn't understand how variables update





This material is based upon work supported by the National Science Foundation under Grant No. 1348866. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References:
Baxter, J. A. (1987). *Teacher Explanations in Computer Programming: A Study of Knowledge Transformation* (Doctoral Dissertation). Stanford University, CA, United States.
Bender, E., Hubwieser, P., Schaper, N., Margaritis, M., Berges, M., Ohrndorf, L., ... Schubert, S. (2015). Towards a Competency Model for Teaching Computer Science. *Peabody Journal of Education*, *90*(4), 519–532.
Gal-Ezer, J. & Stephenson, C. (2010). Computer science teacher preparation is critical. ACM Inroads, 1(1), 61-66.
Lapidot, T. (2005). *Computer Science Teachers' Learning during their Everyday Work* (Doctoral Dissertation). Technion University, Israel.