DOI:10.1145/2366316.2366327

Aman Yadav and John T. Korb

Education

Learning to Teach Computer Science: The Need for a Methods Course

A multipronged approach to preparing computer science teachers is critical to success.

Computer science is a crucial driver of innovation and productivity in our technology-rich society and there is a strong demand for computationally educated workers. Yet, there is a shortage of computer science undergraduates and computationally educated students. Furthermore, computer science plays only a minor role in high school curricula in the United States. Exposing students to computer science in K–12 education could be a natural pipeline for the field, but the U.S. is facing a very different reality: only 15,000 high school students take the Advanced Placement Computer Science (APCS) exam annually, less than 10% of U.S. high schools offer an APCS course, and nationally there are only 2,000 teachers qualified to teach the APCS course.

The National Science Foundation has started an ambitious effort, the CS10K Project, to have computer science taught in 10,000 schools by 10,000 highly qualified teachers by 2016. However, teaching quality computer science courses not only requires substantial understanding of the specific subject matter, but also a solid background of pedagogy. These two areas of knowledge, referred to as "content knowledge" and "pedagogical knowledge" are important to someone who hopes to become an effective teacher. However, knowledge of just the content and just the pedagogy is insufficient. It is also important to understand ways of teaching the particular subject matter, that is, to have *pedagogical content knowledge*. *Pedagogical content knowledge* is a kind of knowledge that "goes beyond the knowledge of subject matter per se to the dimension of subject matter knowledge for teaching.... [and it includes] the ways of representing and formulating the subject that make it comprehensible to others."[13]

Prospective teachers gain knowledge in these three areas from three kinds of courses: content courses in their discipline (for example, mathematics and physics), pedagogical courses that provide broad educational training (such as learning theories and classroom management), and methods courses (pedagogical approaches to teaching a specific discipline). For example, a mathematics methods course weaves together knowledge of mathematics with knowledge about how children learn mathematics, how the classroom environment influences that learning, and the teachers' role in student learning.[1]

ARISTOTLE

"Teaching is the highest form of understanding."

In this column, we argue that in order to meet the CS10K Project goals, high school CS teachers must have in-depth computer science knowledge as well as strong pedagogical content knowledge, developed through a computer science methods course.

Learning to Teach

Learning to teach can be conceptualized around four main ideas—learning to *think* like a teacher, learning to *know* like a teacher, learning to *feel* like a teacher, and learning to *act* like a teacher.[7] These knowledge systems are developed with a comprehensive understanding of the subject matter to be taught as well as ways of teaching that subject matter, that is, pedagogical content knowledge. Teachers with in-depth pedagogical content knowledge understand ways of representing and formulating the subject matter—using powerful analogies, illustrations, examples, explanations, demonstrations, and so forth—to make it understandable to students.[13] These teachers also know which topics students find easy or difficult to learn, which ideas (often misconceptions) students bring with them to the classroom, and how to transform those misconceptions. In addition, teachers understand how students develop and learn as well as how to teach diverse learners.

A methods course is typically where prospective teachers are introduced to this skill set and learn about "pedagogical ways of doing, acting, and being a teacher."[1] This knowledge is developed within the context of learning and teaching a particular subject area. Transforming Ball's statement about mathematics to computer science implies that a computer science methods course is about how computer science is learned and taught, and about how classrooms can provide an environment for learning computer science.

A Computer Science Methods Course: Learning to Teach CS

Recently, there has been discussion within the CS education community concerning the need for teacher licensure programs to address the critical need to adequately prepare high school CS teachers.[8] At Purdue University, we offer a CS Methods Course as part of our computer science supplemental licensure program.[a] The methods course builds upon existing course work in computer science and education to prepare secondary STEM education majors to be effective high school computer science teachers. Prospective teachers take the course as a final step before their student teaching experience.

Our CS methods course develops students' pedagogical content knowledge through experiences that allow them to think and act like computer science teachers. Specifically, the methods course trains prospective computer science teachers to combine pedagogical principles with computer science content to improve the learning experience for their students. Prospective teachers are expected to understand computer science concepts as well as to have a basic understanding of how children learn and how to promote effective learning strategies.

The specific activities and assignments in the methods course help prospective teachers understand and address computer science classroom situations. The course involves reading, discussing, and reflecting on papers that describe pedagogical practices of teaching computer science principles.

Teachers with in-depth pedagogical content knowledge understand ways of representing and formulating the subject matter.

Reading materials for the course are selected from a variety of sources, including ACM SIGCSE, *Reflections on the Teaching of Programming*,[3] and *Guide to Teaching Computer Science*.[9] For example, we use Wu's Castle[6] and image processing[4] to illustrate ways of teaching arrays and loops.

The course includes a laboratory component in which students review and evaluate various pedagogical tools, such as DrJava, Alice, Scratch, Light-Bot, Greenfoot, and *CS Unplugged*[2] activities. The lab also includes a review of topics covered in the AP Computer Science (Java) course, including GridWorld.[b]

Prospective CS teachers develop lesson plans for teaching computer science topics, chosen in consultation with the course instructor. The lesson plans detail student learning goals, how the teacher (and students) will accomplish those goals, a corresponding laboratory activity, and methods of assessing whether or not students have met the learning goals. The prospective teachers use one of their lesson plans for a micro-teaching experience during which they "teach" their peers/course instructors (who role-play as students) and practice their teaching skills in a supportive environment. In addition, students in the course also complete a "theory in practice" component (that is, field experience visits), which involves observing local high-school computer science classrooms.

Students keep two types of journals: a learning journal and a reflection journal. In their learning journal, students record their reactions to the assigned readings and class activities using a set of guided questions. In their reflection journal, they think and reflect upon their visits to the school as well as their micro-teaching experiences. These prospective CS teachers benefit from their reflections because it helps them develop a clear and more complete understanding and insight into computer science teaching. Specifically, reflections have been found to be very effective in developing a teacher's thinking, since "having to express oneself to others, so that others truly understand one's ideas, reveals both the strengths

a The Purdue Computer Science Teaching Supplemental Licensure Program prepares education majors to teach computer science in secondary schools. For more information and related coursework see http://cs4edu.cs.purdue.edu/license.

b See http://apcentral.collegeboard.com/apc/public/courses/teachers_corner/4483.html.

and the holes in one's thinking."[12]

The multifarious experiences built into our methods course, such as classroom observations and micro-teaching combined with reflections provide opportunities for prospective CS teachers to *think*, *know*, *feel*, and *act* like a teacher. The activities used in our methods course also meet NCATE (National Council for Accreditation of Teacher Education) requirements of planning and delivering lesson plans, observing secondary CS teachers, and evaluating their own practice of teaching. The many experiences (field experience, observation, micro-teaching, reflections, and so forth) provided in our methods course equip prospective CS teachers with the ability to use a broad range of pedagogical approaches and computing tools when they begin teaching a computer science class (ISTE[c]).

We have presented one framework for teaching a computer science methods course; however, this model is not meant to be prescriptive. Other researchers have also been involved in teaching computer science methods courses and have presented a similar framework for their methods course (see, for example, Lapidot and Hazzan[11]). CS educators interested in teaching a methods course must be aware of contextual factors (for example, student background, state teaching standards, and teaching licensure requirements) that are relevant to their course. Regardless of the framework, a quality methods course builds on students' comprehensive understanding of computer science concepts and a thorough knowledge of pedagogy.

Conclusion and Future Directions

To meet the CS10K goal, we need a multipronged approach to preparing computer science teachers, both at the preservice and inservice levels. A CS methods course is a necessary component to prepare not only preservice, but also inservice teachers to teach rigorous computer science courses at the high school level. A methods course geared toward the needs of inservice teachers could help them meet the challenges of a CS classroom, such as creating and maintaining a productive lab environment and assessing open-ended student work (programs), while giving them opportunities to grow their pedagogical content knowledge and enhance their overall computer science knowledge.

Preparing CS teachers is an important step to increase the role, availability, and recognition of computer science in high schools.

Typically, high schools cannot afford a full-time CS teacher (for example, due to low enrollment), but instead must rely on teachers whose primary licensure is in another content area, such as mathematics. This situation can lead to teachers with insufficient pedagogical content knowledge to teach CS courses. Furthermore, these teachers face isolation in their school because there are no other teachers with whom to brainstorm ideas to meet the challenges of a CS classroom. The combination of rapidly changing computer technology and busy workday schedules of teaching multiple content areas also makes it difficult for teachers to stay current in the field.

Historically, inservice teachers have gained knowledge through university courses as well as professional development opportunities, including workshops. However, professional development approaches that rely heavily on "one and done" workshops are known to have limited success.[5] Hence, there is a critical need to provide inservice CS teachers with opportunities for in-depth and continual training so they can gain sufficient knowledge necessary to offer rigorous computer science courses.

These approaches to professional development need to be sustainable, scalable, and fit well within teachers' busy schedules. One potential way to meet this need and provide an accessible means for inservice teachers to expand knowledge and develop skills to teach computer science (that is, pedagogical content knowledge) is through online courses. An online methods course would allow teachers to continue their learning and communicate with their peers about topics and techniques they use when teaching computer science. We believe such a course has the potential to support inservice CS teachers and develop their pedagogical content knowledge.

Preparing CS teachers is an important step to increase the role, availability, and recognition of computer science in high schools. And given that CS teachers need various kinds of knowledge and skills to be successful in the classroom, a CS methods course is a necessary and important step to meet this goal. C

c See http://www.iste.org/Libraries/PDFs/NCATE_ISTE_csed_2002.sflb.ashx.

References

1. Ball, D.L. Breaking with experience in learning to teach mathematics: The role of a preservice methods course. *For the Learning of Mathematics 10*, 2 (Feb. 1990), 10–16.
2. Bell, T., Witten, I.H., and and Fellows, M. *Computer Science Unplugged: An Enrichment and Extension Programme for Primary-Aged Children.* Computer Science Unplugged, Canterbury, New Zealand, 2006.
3. Bennedsen, J., Casperson, M.E., and Kolling, M. *Reflections on the Teaching of Programming.* Springer, 2008.
4. Burger, K.R. Teaching two-dimensional array concepts in Java with image processing examples. In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education.* ACM, New York, 2003, 205–209.
5. Dickinson, D. and Caswell, L. Building support for language and literacy in preschool classrooms through in-service professional development: Effects of the Literacy Environment Enrichment Program (LEEP). *Early Childhood Research Quarterly 22*, (2007), 243–260.
6. Eagle, M. and Barnes, T. Wu's castle: Teaching arrays and loops in a game. *SIGCSE Bulletin 40*, 3 (Mar. 2008), 245–249.
7. Feiman-Nemser, S. Teacher learning: How do teachers learn to teach? In *Handbook of Research on Teacher Education: Enduring Questions in Changing Contexts (Third Edition)* M. Cochran-Smith, S. Feiman-Nemser, and D.J. McIntyre, Eds. Routledge, New York, 2008, 697–705.
8. Gal-Ezer, J. and Stephenson, C. The current state of computer science in U.S. high schools: A report from two national surveys. *Journal for Computing Teachers*, Spring 2009.
9. Hazzan, O., Lapidot, T., and Ragonis, N. *Guide to Teaching Computer Science.* Springer, 2011.
10. Horstmann, C. *Java Concepts.* Wiley, Hoboken, NJ, 2010.
11. Lapidot, T. and Hazzan, O. Methods of teaching computer science course for new prospective teachers. *Inroads—The SIGCSE Bulletin 35*, 4 (2003), 29–34.
12. Rodgers, C. Defining reflection: Another look at John Dewey and reflective thinking. *Teachers College Record 104*, 4 (Apr. 2002), 842–866.
13. Shulman, L.S. Those who understand: Knowledge growth in teaching. *Educational Researcher 15*, 2 (Feb. 1986), 4–31.

Aman Yadav (yadav0@purdue.edu) is an associate professor of Educational Psychology in the Department of Educational Studies and also has a courtesy appointment in the Department of Computer Science at Purdue University, West Lafayette, IN.

John T. Korb (jtk@purdue.edu) is the assistant head of the Department of Computer Science at Purdue University, West Lafayette, IN.