

Promising New Pedagogical Approaches for Teaching High School Computer Science

Prepared by Chris Stephenson

Executive Director, Computer Science Teachers Association

Summary

Computer science education has traditionally been characterized by an unfortunate focus on the technology—development of integrated development environments for example—rather than on concepts and teaching. However, the field is now in the midst of a transformation as researchers are grappling directly with curriculum and teaching issues. This focus is both the result of the current enrollment crisis (the precipitously low number of students choosing to study computer science) and of a profound recommitment to engagement and accessibility (especially for under-represented student populations). But more broadly, these changes reflect the distinct shift in computer science's import to society and how the field itself thinks about the need to engage students.

Framework for Curriculum and Pedagogy

To discuss the current state of computer science education in high schools, it is important to clarify the terminology. A *curriculum* is a specified set of objectives or student learning outcomes for a course or series of courses. The curriculum specifies the set of concepts and skills that the students must learn, that is, what will be taught. The term *pedagogy* refers to the teaching strategies that will be used to communicate the concepts to the students and ensure their mastery, that is, how the concepts will be taught.

The publication of the *ACM Model Curriculum for K-12 Computer Science* in 2005 represented the efforts of the computer science education community to provide a curricular framework that identified the primary concepts that must be taught in K-12. It is a bi-level structure consisting of: 1) a set of concepts to be interwoven in all subject areas in K-8; and, 2) a set of three formal high school courses focusing exclusively on computer science concepts. Since its publication, this document has provided a rigorous framework upon which many researchers, schools, and states have constructed comprehensive curricular and assessment guidelines.

New Approaches for Computer Science Education

At the same time, a number of new pedagogical approaches to computer science instruction are having considerable success at revitalizing teaching and learning at both the secondary and post-secondary level. The following sections provide a brief description of four of the most promising practices: **the Exploring Computer Science approach, gaming, media computation, and storytelling with Alice**. (All of the software and curricula described here are provided free of charge.)

Exploring Computer Science:

The *Exploring Computer Science (ECS)* curriculum is a year-long, project-based college-preparatory course for high school students. Tapping into students' fascination with technology, ECS introduces computational thinking—the problem-solving, logical thinking at the heart of technology innovation and creation. It also focuses on developing core problem solving and analytical skills. A range of instructional strategies are woven throughout the curricular materials to highlight this instructional design, including: role playing, jigsawing activities, pair and small-group collaboration, structured tinkering, multiple solutions, utilizing manipulatives, modeling simulations, English language learner considerations, multimedia presentations, journal reflections, and interdisciplinary connections. There are six instructional units included in the *Exploring Computer Science* curriculum: Human-Computer Interaction, Problem Solving, Web Design, Programming, Robotics, and Applications in Computing. These topics are aligned with K-12 computing curricular standards and include daily lesson plans for teachers. Each instructional unit begins with an introduction to a final unit project so that students have a sense of the purpose and applications of the skills they learn throughout the unit. Professional development that also is based on hands-on inquiry is a critical component part of this curriculum.

Gaming:

We can build upon students' fascination with computer gaming (84% of children ages 8 - 17 years old own at least one computer video game and play games for a minimum of one hour per day) to create engaging environments for computer science concepts. In introductory courses, computer gaming can be used as a motivational technique to improve recruitment and retention in computer science. In such a course, the teaching and learning of fundamental concepts can be embedded in hands-on problem-solving and design activities focused on designing and building a game. Students can develop a more thorough understanding of the algorithms, data structures and other topics in computer science while engaged in active learning.

Game Maker software (light edition) provides an environment that is intuitive and easy to use and that introduces students to programming using an objects-first approach. Game Maker is particularly suitable for novice programmers because it allows them to develop games without a previous programming background. It includes a "drag-and-drop" interface that is used to define objects and their actions. Objects live in rooms and react to player input (mouse, keyboard) and to other objects (collisions).

Media Computation:

Media Computation is an approach to introducing computing that emphasizes the manipulation and creation of multimedia. Students in Media Computation classes write programs that manipulate pixels and samples to create images, sounds, and animations. Media Computation materials have now been developed for use with Python, Alice, and Java, and for introductory computing (APCS and undergraduate CS1) and data structures classes. Classes using Media Computation address the same concepts as more traditional versions, but using media as the focus. For example, students iterate over all the pixels in an image to compute a negative, or find the largest sample value in a sound in order to maximize volume. This approach addresses the concerns about "relevance" that have been identified as a factor influencing students' (especially female students) perceptions of computing. We present evidence, drawn from studies at Georgia Tech, Gainesville College, and University of Illinois-Chicago, that Media Computation has played a role in improving student success rates, especially among underrepresented groups. A common finding in these studies is that students spend more time on assignments, beyond what is required, because they want the media produced to be engaging (e.g., "cool").

Storytelling with Alice:

Alice has been used as the basis for the creation of an innovative curriculum for teaching introductory computing courses. Alice uses the context of animation to excite students about the possibilities of computing. In Alice, students author and implement their own 3-D animated movies and design and build their own 3-D interactive games as they learn to program. Alice has been used in hundreds of colleges, universities, and K-12 institutions to increase the comprehension of basic computer science topics of novice programmers. The Alice programming environment allows students to create programs using a "drag-and-drop" editor. Because the Alice environment does not allow students to make syntactic errors, programming is reduced to the development and implementation of appropriate algorithms, an essential programming skill regardless of language. Further, the Alice software allows students to test their algorithms visually. The Alice curriculum is designed to exploit these features of the Alice software for the purpose of learning to program. Alice has been shown to be effective in college (both in attracting students to computing as well as helping to retain "at risk" computing majors). A new study is demonstrating its effective impact in high and middle school.

URLs:

Exploring Computer Science Curriculum: <http://csta.acm.org/Curriculum/sub/ExploringCS.html>

Game Maker: <http://www.yoyogames.com/>

Media Computation: <http://www.mediacomputation.org>

Alice: <http://www.alice.org>