The CS Principles Project[1]

Owen Astrachan, Duke University
Amy Briggs, Middlebury College

Abstract

The Computer Science Principles project is part of a national effort to reach a wide and diverse audience of students and share with them the deep and rich intellectual and practical contributions of computing. Members of the CS Principles community are building a new course—designed to be an introductory college-level course for everyone—a course rich in computer science content and exciting in its pedagogy. The course provides an academic foundation for understanding these contributions and for emphasizing the intellectual, practical, and creative aspects of the field of computer science.

A new introductory computing course for everyone

The Computer Science Principles project is part of a national effort to reach a wide and diverse audience of students and share with them the deep and rich intellectual and practical contributions of computing. Members of the CS Principles community are building a new course—designed to be an introductory college-level course for everyone—a course rich in computer science content and exciting in its pedagogy. The course provides an academic foundation for understanding these contributions and for emphasizing the intellectual, practical, and creative aspects of the field of computer science.

A collaborative effort

In fall 2008 a group of computer scientists and educators convened in Atlanta under the auspices of an NSF-funded conference, *Computational Thinking and Fluency in the 21st Century*, to discuss the future of computer science education and to identify emerging models. It was widely agreed that students require increasing skills in computing across all disciplines and that a new high school course on a national

[1] © ACM, 2012. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in ACM Inroads, VOL 3, ISS 2, (June 2012), http://doi.acm.org/10.1145/2189835.2189849

scale would be an important step towards developing and providing access to these skills.

The CS Principles Project has since developed as a collaborative effort involving computer science educators and the College Board with support from the National Science Foundation. Professional organizations including the Computer Science Teachers Association and the Association for Computing Machinery have played prominent roles as well in supporting the development of the course and its place within the high school CS curriculum (*see* Reforming K-12 Computer Science Education … What Will Your Story Be?, *this issue*) and the larger CS 10K project (*see* Transforming High School Computing: A Call to Action, *this issue*).

Support for the project from the computer science education community has led the College Board to commit to making the CS Principles course a new Advanced Placement offering in the coming years [1]. In 2009 the College Board convened a commission of 10 computer science educators to lead the course development. That core group has worked closely with an advisory group of 20 college faculty who have greatly assisted in refining the curriculum framework. The curriculum framework is not a daily, weekly, or monthly set of lesson plans, but rather it is a set of standards and learning objectives from which lesson plans and activities can be built. The advisory group and the commission have worked together in an iterative process to design the framework and learning objectives on which the CS Principles course and exam are being implemented.

The CS Principles curriculum

The CS Principles curriculum is specified through a set of Computational Thinking Practices and Big Ideas that identify the content, practices, thinking, and skills central to the discipline of computing and computer science.

Computational Thinking Practices

1. **Connecting Computing**

   Developments in computing have far-reaching effects on society and have led to significant innovations. These developments have implications for individuals, for society, for commercial markets, and for innovation. Students in a CS Principles course will study these effects and connections, and learn to draw connections between different computing concepts.

2. **Developing computational artifacts**

   Computing is a creative discipline in which the creation takes many forms, ranging from remixing digital music to generating animations to developing websites to writing programs and more. Students in a CS Principles course will engage in the creative aspects of computing by designing and developing interesting computational artifacts as well as applying computing techniques to creatively solve problems.

3. **Abstracting**

   Computational thinking requires understanding and applying abstraction at multiple levels ranging from privacy in social networking applications to logic gates and bits to the human genome project and more. Students in a CS Principles course will use abstraction to develop models and simulations of natural and artificial phenomena, use them to make predictions about the world, and analyze their efficacy and validity.

4. **Analyzing problems and artifacts**

   The results and artifacts of computation, and the computational techniques and strategies that generate them, can be understood both intrinsically for what they are as well as for what they produce. They can also be analyzed and evaluated by applying aesthetic, mathematical, pragmatic, and other criteria. Students in a CS Principles course will design and produce solutions, models, and artifacts and will evaluate and analyze their own computational work as well as the computational work that others have produced.

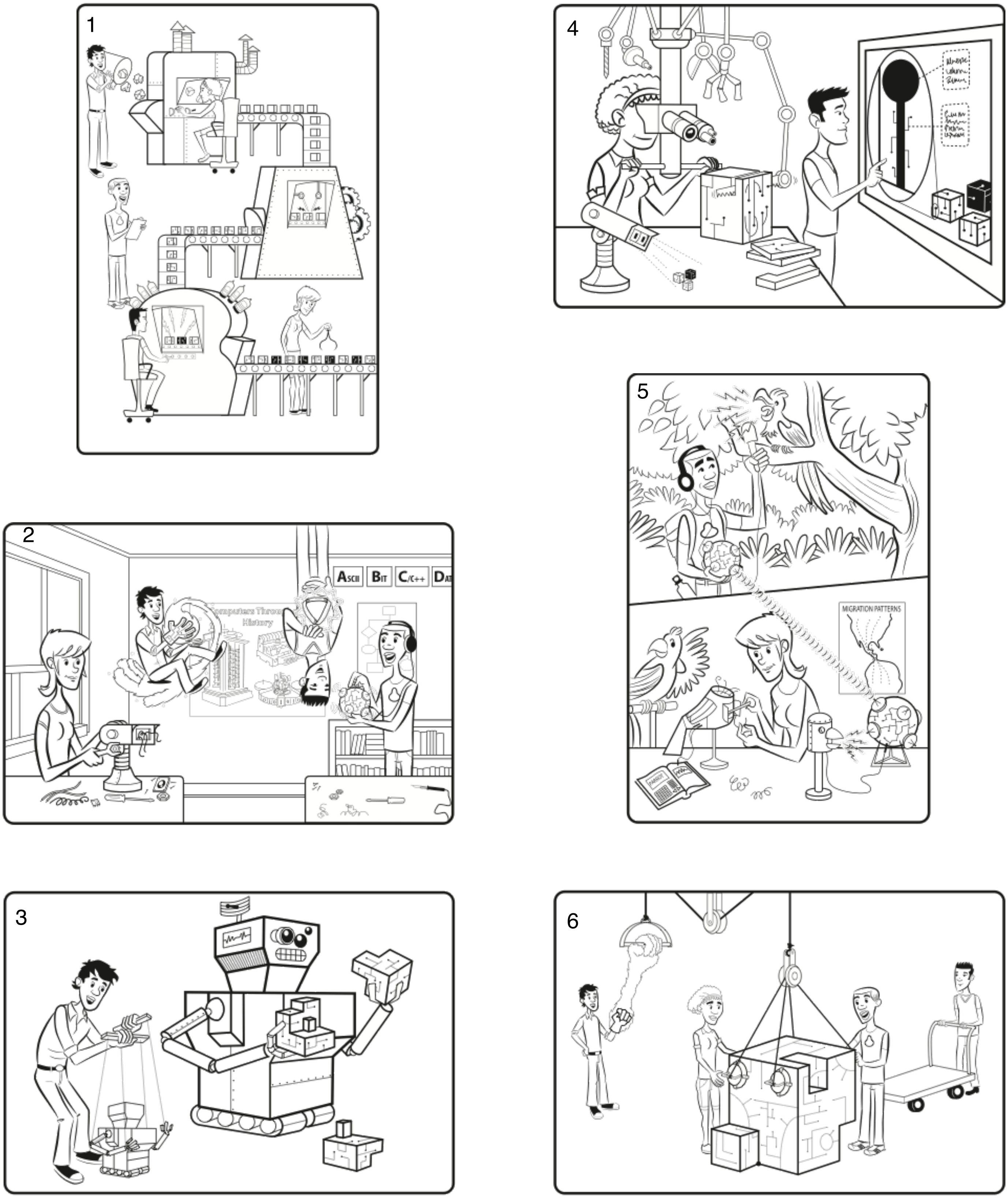
5. **Communicating**

   Students in a CS Principles course will describe computation and the impact of technology and computation, will explain and justify the design and appropriateness of their computational choices, and will analyze and describe both computational artifacts and the results or behaviors of such artifacts. Communication will include written and oral descriptions supported by graphs, visualizations, and computational analysis.

6. **Collaborating**

   Innovation occurs through the work of individuals, teams, and the collaborations among them. Students in a CS Principles course will collaborate with others in the production of computational artifacts, in extracting information and knowledge from data, and in studying the global impacts of computing.

See the accompanying figures for illustrations [2] of these six Computational Thinking Practices.



Illustrations of Computational Thinking Practices: 1. Connecting computing; 2. Developing computational artifacts; 3. Abstracting; 4: Analyzing problems and artifacts; 5. Communicating; 6. Collaborating

Illustrations are licensed under Creative Commons by Tim Piotrowski for ctillustrated.com

Big Ideas of Computer Science

I. Computing is a creative activity.

Creativity and computing are prominent forces in innovation; the innovations enabled by computing have had and will continue to have far-reaching impact. At the same time, computing and computer science facilitate exploration and the creation of knowledge. This course will emphasize these creative aspects of computing. Students in this course will create interesting and relevant artifacts with the tools and techniques of computing and computer science.

II. Abstraction reduces information and detail to facilitate focus on relevant concepts.

Everyone uses abstraction on a daily basis to effectively cope with our world. In computer science, abstraction is a central problem-solving technique. It is a process, a strategy, and the result of reducing detail to focus on concepts relevant to understanding and solving problems. This course will include examples of abstractions used in modeling the world, in managing complexity, and in communicating with people as well as with machines. Students in this course will learn to work with multiple levels of abstraction while engaging with computational problems and systems.

III. Data and information facilitate the creation of knowledge.

Computing enables and empowers new methods of information processing that have led to monumental change across disciplines, from art to business to science. Managing and interpreting an overwhelming amount of raw data is part of the foundation of our information society and economy. People use computers and computation to translate, process, and visualize raw data, creating information. Computation and computer science facilitate and enable a new understanding of data and information that contributes knowledge to the world. Students in this course will work with data using a variety of tools and techniques to better understand the many ways in which data is transformed into information and knowledge.

IV. Algorithms are used to develop and express solutions to computational problems.

Algorithms are fundamental to even the most basic everyday tasks. Algorithms realized in software have affected the world in profound and lasting ways. The development, use, and analysis of algorithms is one of the most fundamental aspects of computing. Students in this course will work with algorithms in many ways: they will develop and express original algorithms, they will implement algorithms in some language, and they will analyze algorithms both analytically and empirically.

V. Programming enables problem solving, human expression, and creation of knowledge.

Programming and the creation of software have changed our lives. Programming results in the creation of software, but it facilitates the creation of more general computational artifacts including music, images, visualizations, and more. In this course programming will enable exploration as well as being the object of study. This course will introduce students to the concepts and techniques used in writing programs and to the ways in which programs are developed and used by people; the focus of the course is not programming *per se*, but on all aspects of computation. Students in this course will create programs, translating human intention into computational artifacts.

VI. The Internet pervades modern computing.

The Internet and the systems built on it have had a profound impact on society. Computer networks support communication and collaboration. The principles of systems and networks that helped enable the Internet are also critical in the implementation of computational solutions. Students in this course will gain insight into how the Internet operates, will study characteristics of the Internet and systems built upon it, and will analyze important concerns such as cybersecurity.

VII. Computing has global impacts.

Computation has changed the way people think, work, live, and play. Our methods for communicating, collaborating, problem-solving, and doing business have changed and are changing due to innovations enabled by computing. Many innovations in other fields are fostered by advances in computing. Computational approaches lead to new understandings, new discoveries, and new disciplines. Students in this course will become familiar with many ways in which computing enables innovation, and will analyze the potential benefits and harmful effects of computing in a number of contexts.

These 6 Computational Thinking Practices and 7 Big Ideas are paired to specify the set of student learning objectives for the course. Refinements to the curriculum framework and learning objectives are ongoing, aided by data from a series of pilot offerings.

Pilot offerings of CS Principles

Five pilot sites were selected to offer the CS Principles course during 2010-11 and participate in the first research phase of the project (*see* The First Five Computer Science Principles Pilots: Summary and Comparisons, *this issue*). The pilot sites each developed and implemented a recruitment plan for increasing enrollment of women and underrepresented minorities, and each course instructor contributed to course evaluation activities such as pre- and post- surveys.

During 2011-12 the pilot group comprised 10 high schools paired with 10 colleges. Pilot sites were chosen according to criteria including geography, ability to enroll a diverse population of students, and ability to partner with another institution (*see* Computer Science Principles at Newbury Park High School, *this issue*). In addition to again participating in surveys and submitting activity logs, piloters further contributed to the research component of the project by administering prototype test items to their students.

The pilot phase of CS Principles will continue in Fall 2012 with the addition of a portfolio component to the prototype assessment. Creativity and collaboration are key components of the student experience that the portfolio-based assessment will support. Portfolio assessments are currently used as part of the Advanced Placement Studio Art courses. Portfolio assessments typically include collecting and evaluating authentic student work, and facilitate assessment of student performance that goes beyond multiple choice and essay questions. The pilots and early adopters are using several different tools, methods, and languages for students to use in understanding the big ideas and computational thinking practices that together comprise the curriculum framework for the course. Portfolio assessments can facilitate valid measures across these different tools in a way that is not possible with more traditional forms of assessment.

Stay tuned

At this point many high schools and colleges around the country are offering their students a version of CS Principles. Early adopters of the course are sharing their own curricular materials and conducting professional development workshops. As the College Board continues to develop and test its assessment for the eventual deployment of this course as an AP offering, the CS education community will continue to shape this important new course and its pedagogy.

See the project website csprinciples.org and its growing repository of course information and links to related resources [3].

Acknowledgments

The authors gratefully acknowledge support for this work from the National Science Foundation, grant 0938336.

References

[1] http://collegeboard.org/csprinciples
[2] http://ctillustrated.com
[3] http://csprinciples.org

Author information:

Owen Astrachan
Department of Computer Science
Duke University
Durham, NC 27708
ola@cs.duke.edu

Amy Briggs
Department of Computer Science
Middlebury College
Middlebury, VT 05753
briggs@middlebury.edu

Categories and Subject Descriptors: K.3.2 [Computers and Education]: Computer and Information Science Education – Computer science education, Curriculum

General Terms: Experimentation, Human Factors, Design, Management, Measurement

Keywords: Computer science education, Computational thinking, CS 10K, CS Principles