Computer Science Principles at Newbury Park High School

Richard Kick

Abstract

Newbury Park High School in Southern California is one of 10 high schools participating in the 2011-12 pilot of Computer Science Principles. Teacher Richard Kick describes student activities from his course, shares some examples of student work, and reflects on student engagement with the material and the inspiration they draw from it.

Our class runs like a jazz orchestra

Newbury Park High School is a Southern California school that prides itself on being identified throughout the community as "Nice People High School". This atmosphere extends to the Computer Science Principles class that consists of 21 students – 11 females and 10 males. As summarized in the College Board Computer Science Principles website [1], the CS Principles course is designed to "broaden participation in computing and computer science." As a computer science teacher for over three decades at five different institutions, I feel like my entire career has had a similar purpose; to spread the word to high school students about how important and powerful computer science and computational thinking are to both the individual and our global society.

The NPHS Computer Science Principles class runs very much like a jazz orchestra. Everyone in the "band" must have an idea of what "songs" are to be played, what tempos will be set, and what keys will be selected. I, as bandleader, begin each class by indicating the songs for the day. I lay down a tempo, and signal the group to begin to play. I then step aside and allow the band to create and flow in directions that their talents and inspirations move them. At times, I will signal to individuals or sections to take the lead. Other times, I signal the end of an exploration and shift the focus to other members of the band. Each time a new direction is suggested, I step to the side and experience the creativity of the inspired performers.

Our course was structured with a fractal model in mind. The self-similar nature of fractal images [13] inspired the NPHS Computer Science Principles curriculum. From a distant view, the course consisted of the seven "Big Ideas" of computer science as listed in the Computer Science Principles website previously mentioned. Explorations that focus on one particular big idea eventually lead to discussions that reveal, even when greatly magnified, all seven big ideas and their relevance to the discussion. The flow of the course consists of "zooming in" on specific details associated with a particular big idea, followed by guided explorations associated with those details. We conclude with a shift back to broader perspectives that helps students make the necessary connections between the course concepts and their work. Students regularly make connections between specific details of the day and the course "Big Ideas" during classroom discussions, online posts, and journal reflections.

Among the tools chosen for enhanced course communication are Piazza and Google Docs. After talking with several computer science educators and using the tool myself during College Board meetings, the Piazza online educational gathering software was selected to enhance teacher-student communications. It quickly became very clear that the student-student communication facilitated by the software was much more important for student success. Google Docs was selected for producing and sharing student journals, group project documents, and general student progress summaries.

Initially, the novelty of silent classroom communications between students led to some interesting exchanges of student humor. However, Piazza quickly became a highly useful academic device for learning. Google Docs allowed me to gain insights into student learning like I had never experienced before. Students were open with their likes and dislikes, their successes and stumbles, and their goals for future explorations. Weekly journaling has been an important part of the learning process for both the students and the instructor. It was also eye opening to watch a small group edit a shared document. The group agreed on specific tasks for each participant, and everyone began to fill in their portion of the document, each student having a separate color-coded cursor simultaneously racing across the screen of every group member.

Student activities

The major activities that students worked on during their first semester of the course were programming activities provided by Nick Parlante from Stanford university through the website CodeInTheBrowser [2], data discussions with a Google software engineer, teaching others about the Internet by creatively using computational tools, exploring and extracting meaning from large data sets using computational thinking, debating the positive and negative impacts of the use of computing technology on our global society, listening to and discussing current and former student presentations on relevant topics in computer science today, discussing articles from a variety of current e-book and Internet sources that were suggested by both the instructor and the students, and reflecting on student learning experiences through weekly journals shared between each individual student and the instructor.

When programming in the CodeInTheBrowser website, students explored programming concepts, discussed suggested problems, and developed their own extensions to the problems provided. Many students wanted more control over the images and their manipulations. Building off of student suggestions, generalized for loops were replaced by traditional for loops and image pixels were traversed in a wider variety of ways. Student creativity generated a range of outputs that were unimaginable by me at the start of our work together. Many of these results came from students who had never done any type of programming before this course. Indeed, they reflected in their journals that they had very little confidence that they would ever be able to write a program. They were pleased by their increased confidence and programming results.

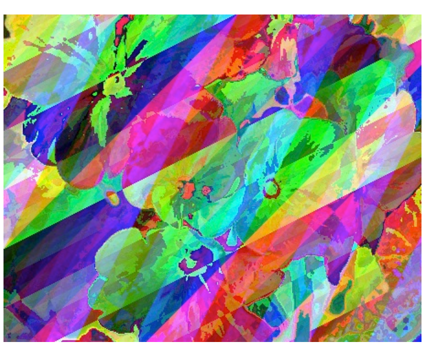


Figure 1: Manipulating images through programming – William

Figure 2: Manipulating images through programming - Eric

Figure 3: Manipulating images through programming – Casey

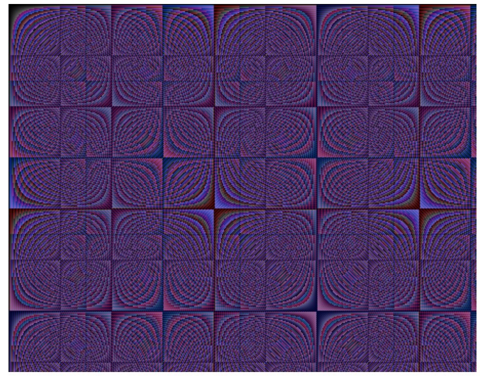


Figure 4: Creating images through programming - Aaron

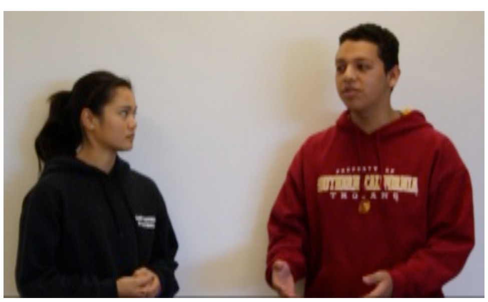
We scheduled a 45-minute virtual visit by a Google software engineer by using Google+ Hangout. Students listened to a short presentation, had the opportunity to ask questions, discussed topics of interest, shared perspectives on what computational tools they are currently using, and discussed tools they would like to use in the future. One student wrote in their journal "My absolute favorite thing about this class is the opportunities we get to talk to people who are currently working in the field of computer science." Another student wrote the following. "Everyone that we have talked to is just so enthusiastic about computers that it really makes me want go into the field of computer science. I feel like the positive attitude about computer science that surrounds this class allows me to enjoy the time in this class and makes me want to continue working in the computer science world."

Figure 5: Talking with a Google software engineer using Google+ Hangout

It is most rewarding to facilitate students in their educational explorations by helping to align their talents with their passions. The group projects seemed to best facilitate that alignment. Students were given the task of educating others about the Internet and how it works. They were allowed the freedom to create any product that they believed would accomplish that task. We had a group brainstorming session that generated a list of possible methods for presenting concepts related to the Internet. Students voted on which project idea to rank as best by moving to a portion of the classroom that was designated for a particular project idea. Students were then told that they had selected their group topic and members by their vote. They then began the process of defining their group goals, assigning tasks to specific students, and planning the steps needed to create their educational products. Ideas ranged from an Internet Rap to a video skit. One student wrote the following about her experience. "While creating our Internet project, [we] are creating lyrics for our rap. I'm normally not a very creative thinker, but having to consistently create new things in class and do group projects has sparked creative ideas."

Several students were proud of the fact that they could extract information from large files of data by writing appropriate programs. However, others argued that they could obtain the solutions more easily in other ways using existing software. Because of this experience, I am suggesting a variation on Astrachan's Law [3]. Give assignments that may be done more easily in other ways besides programming, but don't limit student solutions by requiring only one form for the solution. On one particular assessment this semester, I asked students to "Explain a step-by-step process you can use to modify an image so that the resulting image is a grey-scale version of the original image." Since this was similar to a programming problem we worked on earlier in the semester, the vast majority of the students described the steps they had implemented in their programs to create grey-scale images. One student, however, confidently described the steps they would take to use Photoshop to accomplish the task. Students were pleasantly surprised to learn that the answer was accepted as completely correct.

The technology debate focused on the positive and negative impacts of the use of computing technology in our global society. Students chose whether or not they believed that the positive impacts of such uses outweigh the negative impacts. Some students chose to argue for a side other than what they truly believed in order to balance the size of the groups. After significant research and organization efforts, students held a classroom debate using techniques some had learned as participants in Mock Trial activities. The discussion that followed the debate led to the creation of a video that captured the arguments that they felt best illustrated the two sides.



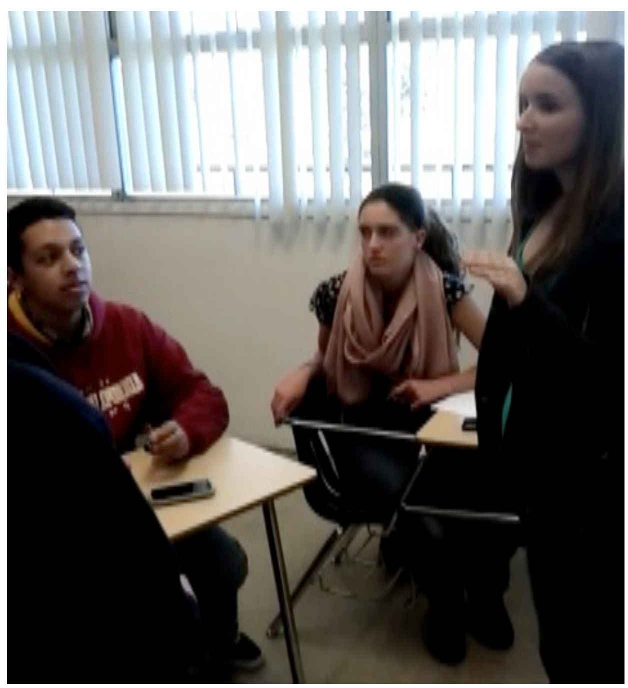


Figure 6: (above) Technology debate - Casey and Andrés

Figure 7: (right) Coordinating the technology debate - Andrés, Emily and Elaine

Of all of the activities that were experienced during the first semester of this course, it is the journaling that I feel was most beneficial for both the students and the instructor in terms of appropriately adjusting the direction of the course. Effectively teaching computer science involves inspiring students to explore topics that are interesting to them, providing examples that are relevant to their lives, and connecting course content with the work people are currently doing in computer science that is positively changing the world. We were fortunate to have several current and former students share their interests in computer science in a way that inspired the Computer Science Principles students. A student wrote this in their journal. "Talking to students who are currently in college learning these things is really awesome."

A Life-Changing CS Course

After reading the student journals throughout the semester, it is clear that what is important to individual students varies dramatically. Their individual life experiences make content more or less meaningful to them. Choosing in advance the best ways for them to explore computing is unrealistic. Instead, I guide them to general topics about which they should be familiar. I ask questions, respond to their answers with interest and enthusiasm, and attempt to inspire students to explore further and share their discoveries and passions with their peers. One student wrote that they appreciated the instructor's "general enthusiasm for everything". My self-analysis leads me to believe that it is probably more accurate to say that the instructor has an enthusiasm for everything for which others display a passion. I appreciate that which inspires my students. I enjoy analyzing what it is that they find interesting and motivating. I look for the beauty as seen through their eyes.

In conclusion, I hope that this article has provided some insights into the experiences my students and I share while piloting the Newbury Park High School Computer Science Principles course. It is my hope that students and teachers across the country will be provided the opportunity to explore their talents and passions during their explorations into computer science, computational thinking, and personally relevant problem solving. One female student summarized her experience with the following. "To be honest, I did not know if this class was the right thing for me or if it would be a waste of my time. I realized, after the first class, that this class was going to be a life-changing event."

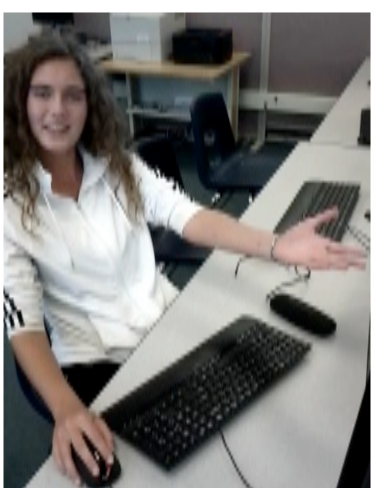


Figure 8: Sharing graphical programming results – Emily

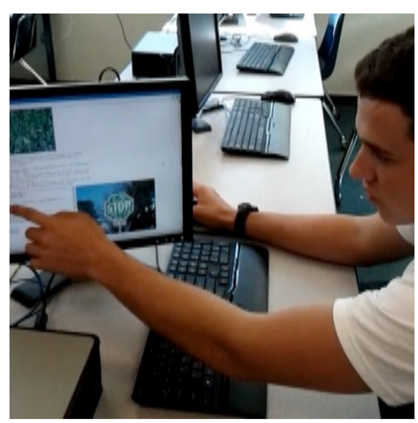


Figure 9: Sharing graphical programming results - Jack

Acknowledgement

Thank you to my Computer Science Principles class for providing me with constant inspiration and information. I teach because that is how I best learn, and I have learned so much from all of my students past and present. Also, thank you to Dr. Leo Michelotti for providing me with an invaluable opportunity to work with an incredible group of computational thinkers at Fermi National Accelerator Laboratory while I explored object oriented three-dimensional graphics programming, and computational problem solving. Of course – this work was supported by NSF grant 0938336 – a big thank you to the National Science Foundation and the College Board for making all of this possible.

References

[1] "CS Principles", http://csprinciples.org/

[2] Parlante, "Code In The Browswer", http://codeinthebrowser.org/

[3] Parlante, "Astrachan's Law", ACM Inroads, Volume 35, Number 4, 2003 December

[4] Astrachan, Briggs, Diaz, Haynie, and Stephenson, "Re-imagining the First Year of Computing", In Proceedings of SIGCSE '10

[5] Dodds, Garcia, Huang, and Rebelsky, "Teaching Tips We Wish They'd Told Us Before We Started", In Proceedings of SIGCSE '11

[6] Bruce, Cupper, and Drysdale, "A History of the Liberal Arts Computer Science Consortium and its Model Curricula", Transactions on Computing Education (TOCE) , Volume 10 Issue 1

[7] Guzdial, Perry, and Rich, "A CS1 Course Designed to Address Interests of Women", In Proceedings of SIGCSE '04

[8] Aiken, Sahami, and Zelenski, "Expanding the Frontiers of Computer Science: Designing a Curriculum to Reflect a Diverse Field", In Proceedings of SIGCSE '10

[9] Astrachan, "A New Way of Thinking About Computational Thinking", Journal of Computing Sciences in Colleges , Volume 25 Issue 1

[10] Wing, "Computational Thinking", Journal of Computing Sciences in Colleges , Volume 24 Issue 6

[11] Guzdial, "Does Contextualized Computing Education Help?", ACM Inroads, Volume 1 Issue 4

[12] Kumar, "A skit-based approach to teaching web and networking protocols", ACM Inroads Volume 2 Issue 4, December 2011

[13] Peitgen, Jurgens, Saupe, "Chaos and Fractals", Springer-Verlag, 2004

[14] Cuny, “Finding 10,000 Teachers”, http://csta.acm.org/Communications/sub/CSTAVoice_Files/csta_voice_01_2010.pdf

[15] Astrachan, Briggs, and Diaz, “AP Computer Science: Principles Project Overview”, http://csta.acm.org/Communications/sub/DocsPresentationFiles/2009APCSgeneral.pdf

Author Information

Richard Kick
Newbury Park High School
Newbury Park, CA 91320
rkick@kickstyle.net

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, pedagogy, CS 10K, CS Principles