

3 powerful words can unlock computer science success

By Janice Mak

This article originally appeared as a blog post. To access the online version and related links visit www.bit.ly/2PzajwF

I was reminded recently of the three powerful words that capture the everyday miracles of learning and teaching. It happened as my sixth graders were just beginning to create ordered and unordered lists during their first lessons in HTML. All at once, I heard a chorus of voices exclaim with joy, "I did it!"

There you have it. Three powerful words. I did it.

It struck me in that moment that this is what learning is all about. It's about being able to do something that I was not previously able to do. It's about surprising myself with what I am capable of doing. It's about stretching beyond what is currently possible and continuing to push the boundaries of what is possible. That is empowerment.

The most incredible thing is that pretty soon these "I did it's" will imperceptibly lead to a shift in mindset to "I can do it." And, as a computer science educator, this shift is precisely what is most needed to expand participation in computing. This shift will allow students who previously could not envision themselves as computer scientists to do computer science in creative and innovative ways to solve problems in personally relevant ways.

But as I reflect on these inspiring moments, I have to wonder if I have students who still cannot say those three words. I think of Abby, one of three eighth grade girls in my class of 37 students who on the second day of school approached me in tears saying she didn't think she could be successful in computer science because "everyone already knew" more than she did (or so she thought).

At that moment, I realized that there are barriers to computer science hidden from sight — barriers that are in the minds, hearts and perceptions of my students. I also realized a tremendous commission — to teach in a way that could overcome these barriers.

Asking myself these questions brings to mind reading I've done on universal design (UD) and the parallels between UD in the physical world with architecture and [Universal Design for Learning](#).

I love that in the [10 things to know about UD](#) the first thing to know is "Universal Design strives to improve the original design concept by making it more inclusive." In teaching, the same

applies. As I strive to provide these "I did it" experiences to all of my students, it's about making my lesson more inclusive without taking away any rigor. It's also about understanding that UD "aspires to benefit every member of the population by promoting accessible and usable products, services and environments."

In the classroom it's about using strategies and approaches that benefit all of my students, leading to an inclusive learning environment where every student has access to these "I did it" experiences.

Here are some ways you can do it in your classroom or school:

Build pair programming into your computer science classes.

[Pair programming](#) is an approach where two programmers work together at one workstation. One, the driver, writes code while the other, the navigator, has the big picture in mind and coaches the driver as the code is being entered. I have my students switch roles of "driver" and "navigator" every 5-10 minutes, which allows them to collaborate, communicate and problem-solve together.

The National Center for Women and Information Technology (NCWIT) has [resources](#) to support the integration of this collaborative learning structure. This [student-friendly video](#) models, demonstrates and defines pair programming.

Give students opportunities to see computer scientists at work in the real world.

This [video](#) from Code.org shows how computer science is changing everything. [Careers with Code](#) and its [magazine](#) contain a diverse representation of computer scientists along with ideas and actions that students can take to propel their careers forward.

[The Girls in IT: The Facts Infographic](#) from NCWIT provides a strong case for why we need to ensure broad representation in

computing, the deterrents that exist and what we can do to overcome barriers to diversity.

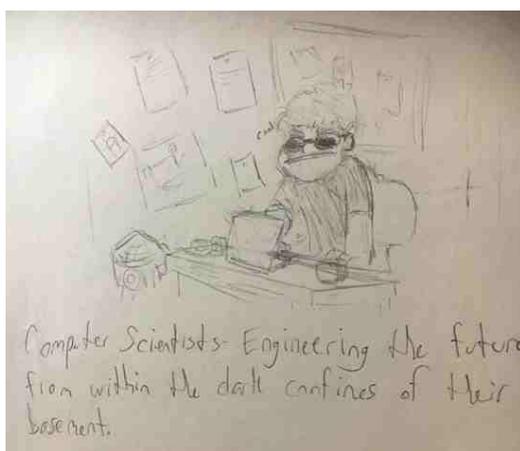
Technolochicas aims to inspire and raise awareness among Latinas and their families about the opportunities that exist in technology. **Mission Possible**, **Life in Code**, and **What's your Coding Super Power?** from NCWIT are three downloadable posters that ignites students' thinking about all the possibilities and applications that come with the field of computer science, inspiring them to make a difference in the world around them.

Reaching out to various women in computing groups at local universities and industry and inviting them to my classroom as guest speakers has been an incredible way to bring their stories to life.

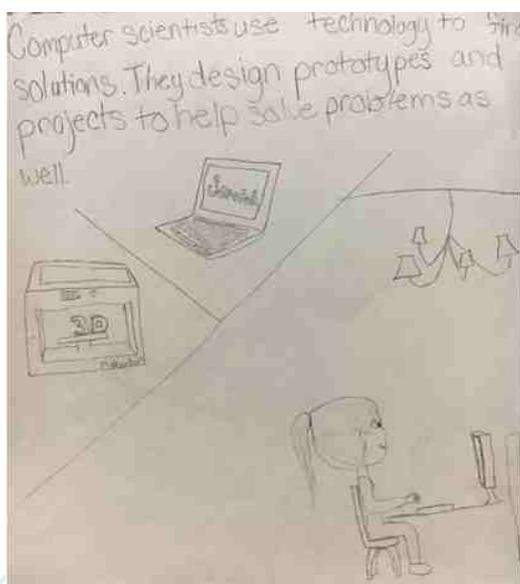
We held a districtwide event to **highlight the personal journeys of women in IT** through the lens of their passion, inspiration, perseverance and advocacy. I really want my students to view themselves as computer scientists who can impact the world through their creative solutions.

That's why I like to have my students draw what they picture a computer scientist looks like at the beginning and the end of a semester. Here is what I love to see:

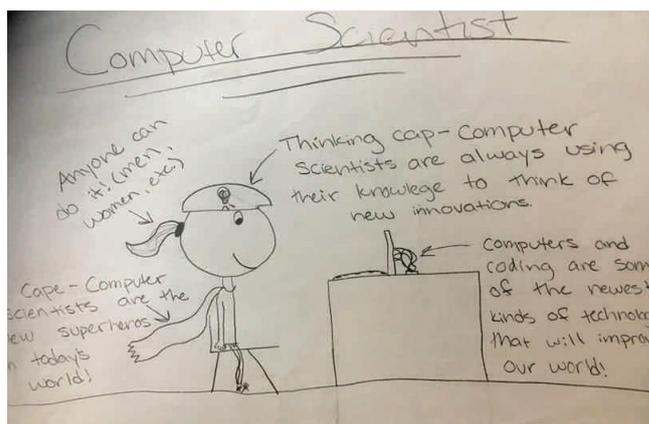
Students shifting their perception of computer scientists from this image:



To this image:



Or to this one, where computer scientists are today's superheroes and anyone can do it!



Allow students to solve authentic problems.

I really like the concept behind **Technovation** and the **Verizon App Challenge**, which provides students with opportunities to identify a problem in their community, research existing apps that address a similar problem and design an innovative app to solve this problem. The cross-curricular connections abound in this project as well as the opportunities for my students to communicate, collaborate and solve problems in teams, thereby creating the "artists" who are only limited by the power of their imaginations.

These types of projects address the **ISTE Standards for Educators**, which call on teachers to "challenge students to use a design process and computational thinking to innovate and solve problems." And they allow students to address the **ISTE Standards for Students** by "using a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions."

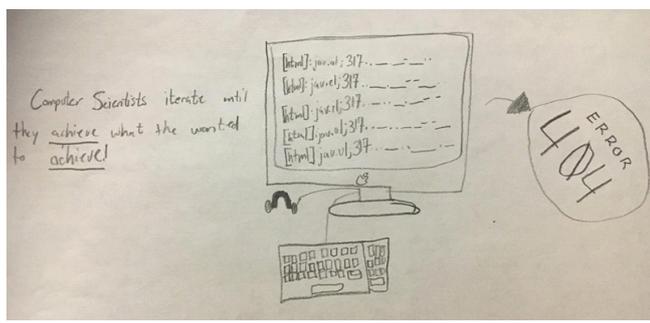
Explicitly teach and model perseverance along with a growth mindset.

With computer science, the opportunities for students to "exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems" (ISTE Standards for Students 4d) come naturally as they "develop, test, and refine prototypes as part of a cyclical design process" (ISTE Standards for Students 4c).

What happens when their program does not work the first time? They need to go back and debug it, rising up from setbacks to continually test and refine their computational artifacts (**Core practice 6, K12 Computer Science Framework**).

I've also integrated debugging by having my students create debugging challenges in **Scratch** for their classmates to solve. This involves creativity, collaboration, computational thinking and communication. Students first need to come with an idea for an animation or drawing and then use computational thinking (ISTE Standards for Students: Computational Thinker)

to create the program to execute their idea. Then, they must strategically decide which piece of code they will intentionally manipulate to create a debugging challenge for others to solve. Or, as one of my students wrote, "Computer scientists iterate until they achieve what they wanted to achieve!"



Create by connecting to the physical world.

Using physical computing really helps my students connect input to output in a tangible way. By creating kinetic sculptures with [Hummingbird Robotics Kits](#), they connect what they program to the physical world through sensors.

They collaborate in teams to design their kinetic sculptures and each of their creations are as unique as they are. Made from materials such as cardboard, paper clips, scraps of fabric and styrofoam, they brought their creations to life with programming. Throughout the process, they kept a design journal with entries that focused on what they accomplished, challenges they faced and how they plan to overcome their challenges.

Here is a reflection from a team member:

If you have an imagination, a computer, and a HummingBird robotics set, you can make anything you want to make. HummingBird Robotics is a series of wires, lights, vibrators, sensors and motors. Throughout all of our series of making the disco room, we ran into a few problems. For instance, making the motor work with the record player was a challenge, because we needed to have a pencil lead in the disc so it will work. Also, we were going to make a whole series of notes on the Piano, and time it to every second to make the song work. Also, our LED lights were ripped, and my group tried to fix it.

Raspberry Pi is another great way to connect to the physical world. The foundation's mission is to bring the power of digital making to people all over the world. The founders believe, as do I, that the complex problems facing society today and in the future will be solved by people who understand and shape our digital world through innovative solutions.

So, where does one start? With the basics such as:

- Make music using Sonic Pi.
- Build and destroy towers in Minecraft.
- Program Picamera with Python 3 to take selfies.
- Operate a motor.
- Light up LEDs with a button.

After learning the basics, it's then a matter of combining, remixing and synthesizing different skills to digitally make an innovation. Here are some helpful guides at www.projects.raspberrypi.org/en to get started:

- [Getting started with and setting up Raspberry Pi](#)
- [Getting started with Sonic Pi](#)
- [Getting started with Picamera](#)
- [Physical computing with Scratch](#)
- [Getting started with Minecraft Pi](#)
- [Physical computing with Python](#)

Provide students with the opportunity to individually reflect on all the Raspi tools in their toolbox and encourage them to dream. That's how I and a team of educators created [selfie-taking roaming robot](#) that tweets out random messages celebrating great work being done in classrooms. Because, in addition to students creating and innovating for themselves, it is critical that they see me, their teacher, modeling what it is to take risks, be creative and be a lifelong learner.

So, whatever happened to Abby, you may ask? I'm happy to report that she did not drop my class. She actually ended up creating multiple computational artifacts including animations, games, art and even a kinetic robotic sculpture with sensors. In fact, midway through the semester, she told me how when her mother started posting links to her projects on Facebook, she rolled her eyes and responded with, "Mom, it's no big deal!" There you have it, the everyday miracles of teaching — witnessing students from diverse backgrounds journeying from "I can't do it" to "I can do it!"

Here's to a year full of empowering our students with many more choruses of "I did it's!"

Janice Mak is an educator from Phoenix, Arizona, and the recipient of the NCWIT Educator Award and Presidential Award for Excellence in Math and Science Teaching. She is co-chair of [CSForAZ](#), a Code.org affiliate, and serves on the [K-12 Executive Council for NCWIT](#), [CSTA Arizona](#) board, [Arizona State Board of Education](#) and [Arizona K12 Center](#) board. You can read more about her classroom adventures on her blog www.supercodingpower.blogspot.com/ and follow her on Twitter [@jmakaz](https://twitter.com/jmakaz).

Copyright 2017, International Society for Technology in Education (ISTE). Reprinted with permission. ISTE affiliates have special reprint permissions. To support our work, consider becoming an ISTE member. Visit iste.org/join for more information.