The Definitive Guide to Computer Science in K-12

by Vicki Davis

ABOUT THE AUTHOR

Vicki Davis @coolcatteacher writes the Cool Cat Teacher Blog and hosts the show Every Classroom Matters. Vicki is a full time computer science and business teacher at her school in Camilla, Georgia.

TABLE OF CONTENTS

FIRST
www.firstinspires.org

ABSTRACT

The United States is leveling up computer science. With $4 billion in funding for states, the Computer Science for All Initiative wants every student to learn computer science. In 2013, the United Kingdom became the first country to mandate computer science education for all students. Today, countries, politicians, and parents realize that computer code has become the world's first global language.

As a result, schools are scrambling. Should we hire computer programmers and help them learn to teach? Should we train teachers to program? With countless websites boasting an easy, online solution for programming — what works?

Never fear. Whether you're a teacher or administrator, if you're working to bring computer science and coding into the classroom as a full course or just as stand-alone units, there are incredible resources for every school and need. Offline and hands-on approaches can teach computational thinking too.

First, let's discuss the obstacles to making progress on adding computer science into your curriculum. Second, we'll discuss some excellent tools, apps, and approaches for all ages. Finally, we'll discuss how to develop a natural audience for your student work. All three of these are important parts of helping the students in your school succeed.

"Every maker of video games knows something that the makers of curriculum don't seem to understand ... Kids who do not like school will tell you it is not because it's too hard. It's because it's — boring."

— SEYMOUR PAPERT FROM HIS 1980s BOOK "MINDSTORMS"

TIP

Free online resources can help you learn how to teach your students.

6 OBSTACLES TO ADDING COMPUTER SCIENCE TO YOUR CURRICULUM

1. Get past the FEAR.

Fear is "false evidence appearing real." When curriculum directors and principals look at computer programs, they feel overwhelmed. They don't understand how students do this. They mistakenly think computer science is too hard to do.

Ask a beginning calculus student to look at a car and ask her if she can design one. She will think she can't. But if she takes one class at a time and learns how she could create a car, one day she may. Computer science isn't new. A body of best practice is here. Any school can become quite advanced if they just take it one step and one class at a time.

2. Computer science is too hard.

Computer science is, by nature, challenging. As Seymour Papert said in "Mindstorms," his 1980s book on teaching programming,

> "Every maker of video games knows something that the makers of curriculum don't seem to understand. You'll never see a video game being advertised as being easy. Kids who do not like school will tell you it is not because it's too hard. It's because it's — boring."

Challenging doesn't mean hard. Challenging doesn't mean impossible. In my classroom, we have programmed video games in Scratch. We've programmed LEGO® MINDSTORMS®. Last year, my students programmed and released their first apps for the iPhone using Crescerance. We started simply, with Hour of Code. Every activity and lesson took us deeper, and even more, it helped me as the teacher learn a little more. Free online resources helped me learn how to teach my students. We've won awards and accomplished great things one step at a time.

None of those tasks was easy for me or my students to learn. But after making the journey once, I become a far better guide. And as I've found other teachers who have made the journey before me, it has become easier. Computer science is not too hard, it is just new. You have to learn and learning isn't easy. But it is not too hard — not for teachers open to learning a little something new.

3. I am alone in my journey.

Isolation is a lie in computer science. As you select a curriculum, also look for a community to help guide you on your journey. Often the best teacher of teachers is other teachers, not a book. Likewise, when done well, some of the best teachers of students are other students. You, as the teacher, are not required to be an expert in everything.

For example, I had a student programming in Scratch. She wanted her game's character to stay still and the background to move. I told her I didn't know how but that she could look at the Scratch resources. She not only mastered the moving background but she taught three other students and me.

4. Computer science takes students away from "real" classes.

With digital resources and communities, students and teachers journey forth to meet the challenge. Most importantly, students move from rote memorization that they'll do in many other subjects, to the highest levels of Bloom's Taxonomy as they create. When you do assignments that you cannot find the answer to on Google, you become a learner on a journey. As you put your hands on objects to fulfill a mission, you not only find novel answers, but you find out what education is.

Tony Wagner and Ted Dintersmith, in their book "Most Likely to Succeed" write,

> "Now, adults need to be able to ask great questions, critically analyze information, form independent opinions, collaborate, and communicate effectively. These are the skills essential for both career and citizenship."

I've found that after my students program apps, that they are better term paper writers. When they solve problems in coding, they are also able to solve more problems in science. Effective computer science, even in small doses, will help students be better learners in other classes. Even better, there are options to use computer science in math, science, history, and other courses.

DID YOU KNOW?

Effective computer science, even in small doses, will help students be better learners in other classes.

5. I need to hire a new person to "do" computer science.

Unless you want to teach Java or Python, you don't necessarily need to hire a new teacher. Even then, there are so many online courses, teachers have a wealth of ways to level up.

Look for your innovators and learners on your campus. You may already have someone ready to innovate but who has never been given the resources or opportunities. A baby doesn't start off running a marathon. Because the United States (and other countries) are just waking up to the need for more computer science, teachers are just now in demand. Not all of this demand can be met by teachers coming out of education school.

I would argue that most of the demand will need to be met by teachers who level up and learn, not those who have been "specially trained" to teach computer science. With less than 10 percent of U.S. schools in the past offering any form of computer science, exciting opportunities are open for teachers who are ready to spend their summers learning Java, Python, or how to teach programming.

Any school can add computer science to their curriculum, so let's talk about the resources to get you there.

6. I don't have time in the schedule.

Some schools make a time of enrichment during the day. Just like students rotate through music and art, they rotate through computer science related topics. But many schools lack the time for these programs during the school day and implement after-school programs, such as those offered by *FIRST*®, which is present in over 10,000 schools across the United States. After-school programs have been found to be particularly good at closing the opportunity gap because children of color are more likely than others to participate.

DID YOU KNOW?

The outcomes of after-school STEM programs are certainly worth a look as one evaluates research.

CONSIDERATIONS AS YOU BUILD YOUR CURRICULUM

As you make your decisions for integrating computer science in all age classrooms, consider communities, creation spaces, hands-on projects, and a try-before-you-buy approach to some tools.

Look for communities.

As you look at the apps, curriculum, and tools you'll use, look for those with strong communities behind them. A great place to start is the Computer Science Teachers Association (CSTA). Membership is free and their resources and newsletter have been invaluable to me on my journey. *(See their Computational Thinking in K-12 teacher resources guide.)* Follow educators like Sylvia Martinez, Alfred Thompson, and Lou Zulli (these three are my go-to computer science geeks).

Also, tap into the expertise of volunteers and parents in your class. Some parents and companies support schools as robotics league coaches. Additionally, MAKE Magazine has online projects using all kinds of STEM concepts. The robust community there will give you lots of ideas for using what you have.

You never know what you'll find online — I recently found a project where two people had made a robot that can solve a Rubik's cube just over a second. It used webcams and 3D printed items as well as an Arduino board. Often a simple idea spurs incredible invention and innovation.

"Now, adults need to be able to ask great questions, critically analyze information, form independent opinions, collaborate, and communicate effectively. These are the skills essential for both career and citizenship."

— TONY WAGNER AND TED DINTERSMITH, IN THEIR BOOK "MOST LIKELY TO SUCCEED"

"The mathematics that most enables students to be successful in college courses is not high school mathematics, but middle school mathematics, especially arithmetic, ratio, proportion, expressions, and simple equations."

— NATIONAL CENTER ON EDUCATION AND THE ECONOMY QUOTED IN "MOST LIKELY TO SUCCEED"

DID YOU KNOW?
When programming combines with hands-on projects, the dynamic duo of computer science and computational thinking unleash incredible results.

Build spaces for creation and performance.

The Maker Movement also has aspects of computer science, design thinking, and programming embedded in its DNA. Some schools are putting in STEAM labs, others call them MakerSpaces, and still others call them a FAB Lab.

Whatever you call it, this blending of woodshop, programming, and artistry is an incredible place for self-expression. Some libraries are renaming themselves Learning Commons and integrating Makerspaces there. Others don't stop there and add performance spaces where students sing and robots rally.

Look for hands-on projects and experiences.

As you look at your curriculum focus on hands-on experiences where students are measuring, calculating, and making. This work builds upon a 2013 study by the National Center on Education and the Economy quoted in "Most Likely to Succeed,"

> "The mathematics that most enables students to be successful in college courses is not high school mathematics, but middle school mathematics, especially arithmetic, ratio, proportion, expressions, and simple equations."

When programming combines with hands-on projects, the dynamic duo of computer science and computational thinking unleash incredible results.

Sample before you commit.

Last November I went on a "tasters' tour" where we went to different restaurants and had small tastes of lots of incredible food. Think of Code.org as a "tasters' tour" for coding. Take students into different experiences and see what they love (and what your teachers enjoy). This was where I started my journey with Scratch, which can make some pretty advanced video games.

Use what you have.

If you have iPads, select tools for the iPad. If you have computers, then use them. When you get ready to invest, there are lots of hands-on options, but select projects that can be used and reused if possible. You don't have to buy anything new.

Let's explore some options for different ages.

COMPUTER SCIENCE OPTIONS BY AGE LEVELS

These ages are generalized. I know teachers who have used more advanced tools with younger ages. Likewise, I've seen very advanced high school projects using Scratch. If you can program it, use it. These are only guidelines. When a tool spans multiple age ranges, I've moved it to the youngest age available.

The Youngest Students (through age 8)

- Code.org offers free 1-day workshops to introduce computer science to grades K-5. Additionally, they offer Code Studio, with courses for ages 4-18 that has trackable units for students and teachers.
- *FIRST*® LEGO® League Jr. introduces STEM concepts to students aged 6-10 using LEGO® elements, in part, as a tool. With the introduction of the new LEGO® WeDo 2.0 system, now the youngest students get exposure to programming using an intuitive drag-and-drop environment.
- Tynker offers courses and a free 6-hour introduction to programming for kids. Tynker's hour of code lessons include excellent play for summer or classroom play. Just set the dial at the top of the page for the grade level. *(For example, puppy adventure is my favorite game for beginning pre-readers.)* You do have to create a free account to be able to play the free games. Help kids or set up their accounts for them. Lots of free play and learning here.

- Project Lead the Way (PLTW) has K-12 STEM options already in more than 8,000 schools across the United States. They include courses for all levels including engineering, biomedical science, computer science, and general STEM.
- iPad Apps: Several iPad apps teach coding. Note that they often require additional purchases to learn higher levels. Kodable includes a curriculum based on 20-minutes a week per student and no prior programming experience. Daisy the Dinosaur is a nice app for younger children, but it still requires some basic reading.
- ScratchJr is a tool on iPhone/iPad, Google Play, and Kindle for young children aged 5-7 to learn to program their own interactive stories and games.
- Lightbot also has programming games for many platforms for ages 4-8 and 9+. Teachers can play some of these games using the web version.
- Robot Turtles is a board game that teaches computational thinking.
- VoiceThread is a tool for uploading sound, pictures, optionally video, and gathering comments from several people. The online version is easy and now they have a mobile version for iPads.

Middle Students (age 8-12)

- *FIRST*® LEGO® League gives students (grades 4-8) a real-world problem to develop a solution. Teams build, design, and program a LEGO® MINDSTORMS robot related to a specific challenge.
- Scratch can be used for many ages of students. It is often at the middle level, but can be used with younger or older students. While Scratch is created and supported by MIT, Harvard has released a free introductory computing curriculum Using Scratch. This is one of my favorite platforms because it can be used with so many other things including LEGO MINDSTORMS EV3 and Raspberry Pi kits like the Hummingbird.
- This year's Hour of Code Games included Star Wars, Minecraft, and Frozen themed games. Star Wars is a great way to introduce Scratch and Minecraft can be played for just about anyone. While younger kids often choose the Frozen game, it is best for those learning basic geometry in middle school. The ages are there for a reason! There are lots of other games on Code.org as well.
- Project GUTS (Growing Up Thinking Scientifically) is a STEM program for middle school students funded by the National Science Foundation. Some free resources are available that you can use including an 8-week free online course for educators.
- Bootstrap puts computer science into the middle school Math curriculum using the WeScheme program that just requires a Gmail Account. If a school has Google Apps for Education, then you should be good to go, just test it first.
- Globaloria is a project-based computer science/video game making curriculum online. Like other sites, they have free coding courses available to get you started. This site includes many higher level courses as well.
- Hopscotch is a fantastic tool to help you program and share games instantly. This is a favorite with students. If you want to buy extra features, they are available for purchase.
- Move the Turtle is a simple game where students learn to program motion. Much like the Frozen game at Code.org, this game can work with math and geometric concepts for middle school.
- Gamestar Mechanic lets students make simple games and share them with each other.
- CS Unplugged has computer science concepts and lessons that you can teach without a computer. You can include these in physical education classes or other subjects.
- Kodu Game Lab is a simple way to program and share a video game made by Microsoft.
- The ShowMe is a tool for creating videos for/by students. This is essentially an online interactive whiteboard.
- Lure of the Labyrinth is a virtual game where students must learn or use their math skills in order to level up and appear on the leaderboards. It reinforces pre-algebra content taught in middle school.

TIP
See a full list of real-world hardware that can access Scratch.

TIP
Remember to play the games before you teach kids.

TIP
The ages are there for a reason! There are lots of other games on Code.org as well.

Older Students (13+)

- *FIRST*® Robotics Competition is for students ages 14-18. Students can program in C++, Java, or using National Instruments' LabVIEW® graphical programming environment, a tool used by professional engineers. *FIRST*® Tech Challenge is somewhat less intensive, and extends down to students as young as 12. Students build robots powered by Android technology, and program using Java or MIT App Inventor.

- Snap! is a reimplementation of Scratch by the University of California at Berkley which adds higher-level programming. Snap! runs in your browser and with the additional capabilities, lets schools use this tool for a "serious introduction of computer science for high school or college students" using the Scratch interface so many know.

- Code.org's Code Studio has a JavaScript Tools for High School App Lab to teach high schoolers how to code in JavaScript.

- CodeHS includes computer science courses and AP Computer Science, helping schools build a four-year computer science pathway into their curriculum. They give you the Full Intro to Computer Science Course and AP Computer Science A course for free with some limitations.

- Codea for iPad lets you create games and simulations on the iPad for the iPad. Codea for iPad was used to create Cargo-Bot, a computational game for the iPad that you can download for free. Cargo-Bot can be used by younger ages, but Codea would be best for older students.

- Scriptkit boasts that it is the first drag and drop programming app for the Ipad. The app says it is free, however, to create your own scripts, it does require a purchase of "edit mode." Test this one before buying but it may meet the needs of some schools who want to do more advanced work but don't have computers.

- Crescerance has the MAD-Learn platform (I use this in my school) to program mobile apps for Apple, Android, and Windows Mobile.

- MIT App Inventor lets students build mobile apps using the Android Platform. They often have contests on their site as well.

- Tutorial sites like Kahn Academy, Udemy, and Lynda.com now have coding courses you can select as well. Sometimes teachers use these websites to learn the basics before going through full-scale professional development courses.

- Visme is an online presentation and infographic creator — high school reports on steroids.

Hardware for All Ages

- LEGO MINDSTORMS EV3 – Some robots come pre-built, but not this one. This robot comes with a programmable brick (think of it like the brain). Students are able to add all kinds of movement, sensors, and even create their own decorations for the robots they build. This robotic option is highly customizable.

- Sphero and Ollie are programmable robots that go everywhere.

- Dash and Dot from Wonder Workshop – Dash moves, but Dot has other features of listening and making sound. These robots often work together.

- Raspberry Pi is a favorite of the maker movement. You can find lots of devices, sensors, and tools made with this handy mini-computer.

- Arduino boards are another type of board that is inexpensive with a robust forum and lots of ideas from MAKE Magazine.

- The Hummingbird Robotics Kit from Carnegie Mellon's CREATE lab is an easy-to-use programmable tool that lets you use Scratch, Snap!, CreateLab, Ardublock, or even Java, Python, and other tools. Additionally, follow the Arts & Bots research by Carnegie Mellon University as they explore the combination of craft materials, robotic components, and programming.

- Android phones make a great hardware platform for young programmers, since many students already own one of these devices. MIT App Inventor makes programming this platform easy and accessible, with advanced students moving onto programming Java using the Android Studio. Android devices are the platform of choice for *FIRST* Tech Challenge and Technovation, a programming challenge for girls.

- SparkFun has resources and add-ons for building robotics as well as their own competition for Arduino, the SparkFun Redboard.

- The BBC micro:bit is making the news since in the United Kingdom it will be given free to every child in year 7 of their school program. Because this small programmable computer is about to be used by so many children in the United Kingdom, many programming tools are making themselves compatible with this pocket-sized computer.

- Make games to reinforce learning on Kahoot and enhance classroom instruction.

AUDIENCE FOR STUDENT WORK

Students need an audience. They are tired of doing work for the teacher's wastebasket. This is truer as students create games, activities, and other work. Showcase student works with these ideas.

Innovation Day

Instead of having a one-hour "genius hour," some schools are taking a whole day for students to innovate and create. At the end of the day, students display, demonstrate, or compete to show their works. (See CUE's guide to Innovation Day.)

Online Gallery of Work

I display my student's work on our school wiki, but other schools have online galleries, student blogs, or other ways to share student work.

Enter Competitions, Contests, and Challenges

Contests and challenges are great ways for students to participate. Whether it is one of the *FIRST* robotics programs, the Microsoft Imagine Cup, or another competition, students love to compete, invent, and celebrate their achievements publicly.

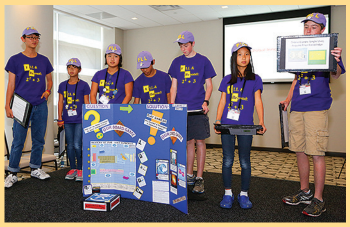
Share on Social Media

Give students an audience. Share their best work on Facebook, Twitter, or where parents see it. Some even use Periscope to live stream events where students share their work. Students can create short videos or screencasts of their work that you can share far and wide if you do not want to share live.

Share in Online Forums or Send in to Magazines

There are so many ways to share student creations: online forums, magazines, ebooks. Students have a wealth of ways to demonstrate their knowledge and help other students and teachers do what they have done.

GLOBAL INNOVATION AWARD



Students present their creative ideas to improve learning in the 21st century in the 2015 FIRST LEGO League Global Innovation Award presented by XPRIZE.®

TIP
Computer science isn't just fun, it is life changing.

BE A DIFFERENCE MAKER

Computer science is easier, more approachable, and teacher and student-friendly than ever. All we need is the mindset to take the resources and ideas here and make it happen.

The most innovative resource in the classroom is an innovative educator. As we open our minds to computer science, we will open up opportunities for students. Computer science isn't just fun, it is life changing. We can do this.

And remember, as you try new things, educators who care, share. Share what you learn with other educators and spread the word. Let's level up together.

www.firstinspires.org