

[BASICS]

ADVICE FROM SCHOOL LEADERS TO SCHOOL LEADERS: INTRODUCING COMPUTER SCIENCE IN YOUR SCHOOL

When beginning a CS program, who better to learn from than other school leaders who have already taken those initial steps? We reached out to 25 school leaders in three school districts whose schools were using the Exploring Computer Science (ECS) high school computer science materials and asked them:

"How would you suggest other school leaders introduce computer science into their schools?"

This guide captures their advice. As one school leader told us:

"I think it'd be helpful to have a checklist. In my first year as a principal, every time I learned something new or went through a process, I'd think "this was something every principal has done, why no one has ever written it down?" Something like a step-by-step guide would be helpful."

This is that check list. We gratefully acknowledge that the format is modeled after the "tip sheets" found at csteachingtips.org (a handy resource that school leaders can share with their computer science teachers).

[1]

Commit to bringing CS to your school

First and foremost, you have to want to do it and be willing to make the investment to bring CS to your school.

[2]

Assess knowledge of and interest in CS for stakeholders within and outside school walls

Determine how much students and teachers already know about CS and student interest in enrolling in CS courses.

[3]

Visit schools with CS programs, sit in on classes, and talk to teachers and school leaders

Take advantage of [what others in your shoes have learned](#) about planning for and implementing a CS program.

[4]

Communicate the importance of CS to stakeholders within and outside your school

Many people know little about computer science. Develop 2-3 short, strategic [messages](#) about why learning computer science skills is essential for students today to communicate to your different stakeholders.

[5]

Determine the right size and scope for your own CS program

Once you've seen CS in action at other schools, decide how you want to approach creating a program in your school, and the budget necessary to commit to that plan.

[6]

Identify desired student learning outcomes and create an assessment plan

"Go into it [a new CS program] with your outcomes in place. Analyze your school and think about what it needs... Know what you want to come out of it."

[7]

Review available instructional materials and identify equipment needs

Do your own homework—ask, *"What instructional resources are available to teach K-12 computer science"* and what additional resources are needed to implement those materials?

[8]

Prepare a realistic budget and allocate resources

"Don't overextend beyond what you can fiscally support."

[9]

Consider who will teach your CS classes and what credentials they need

Identify teachers in your school who can sustain the program beyond the first year, or alternately, where to look to hire credentialed teachers.

[10]

Identify a range of supports for the teachers who will teach CS

Have a range of supports ready for whomever will be teaching your CS class(es), whether they be teachers experienced in teaching other subject areas (who may need more content support) or teachers with more CS industry than classroom teaching experience (who will need more instructional support).

[11]

Be flexible and patient

Plan to make adjustments to the program as it grows.

[BASICS]

ADVICE FROM SCHOOL LEADERS TO SCHOOL LEADERS: INTRODUCING COMPUTER SCIENCE IN YOUR SCHOOL

Further Information...

[1]

Commit to bringing CS to your school

One principal explained, *"It's an investment in the infrastructure of the building. It's an investment in the technology itself. It's an investment in the teacher and overall it's an investment in the students...you have to decide that you're going to make it mean something."*

[2]

Assess knowledge of and interest in CS for stakeholders within and outside school walls

Understanding what stakeholders within and outside of your school [know and think about](#) CS is a necessary early step. One school leader suggested surveying the stakeholder community to gather this information as when you begin any program, *"you want to know what people are excited and scared about so you can address their scare factor right away."*

Another leader spoke of the reality of district requirements related to teaching positions and the importance of demonstrating demand. The leader suggested conducting a student CS needs assessment to see *"if it's something the kid are going to be interested in"* noting that in some schools *"you have to have the numbers. You can't have [only] 5 kids say I'm interested...because your [teaching] positions are based off having 28 to 30 kids."*

[3]

Visit schools with CS programs, sit in on classes, and talk to teachers and school leaders

Go and *"see what's happening."* One school leader suggested going to schools with established CS programs to *"meet with the principals and teachers...talk to the students...about how it's working, what are some of the challenges they are facing, how did they get to this point [and] where things seem to be functioning."* You also might invite some parent/guardians and students from your school to join you on these visits. This is not only an opportunity to benefit from others' experience, but also to *"build networks with people for support."*

[4]

Communicate the importance of CS to stakeholders within and outside your school

It is important to create a sense of urgency about introducing a CS program now – not only to families and other stakeholders outside of the school (from whom *"at some point, you are going to need help in funding, whether that's through grants or local businesses here or there"*), but also to those within the school, including teachers of all subjects, and [counselors](#). If there is misunderstanding about what computer science is and is not, it can be perceived as lacking academic challenge. As one school leader explained, *"Sometimes computer classes are the dumping grounds for students who [counselors think] may not necessarily be able to access the academic challenge in other classes."*

[5]

Determine the right size and scope for your own CS program

What is your capacity for developing and sustaining a program short term? Long term? What type of course sequence, or pathway should you consider? As one school leader explained, *"you have to be mindful of [course] programming, the school resources, and the capacity to do it internally."* Another said *"you need to know the end goal, and not just how to get it*

started. If you want let's say two classes of seniors to complete this program, then you need to backwards-map and say okay if I want two classes in the senior year, that means I probably need at least three classes junior year, four classes or five classes sophomore year and maybe ten classes freshman year. So, do I have the resources?"

It is also important to understand your students' access to technology outside of school. *"If you do an assessment and only 30% of your students will have access outside school, you need to check into your community's resources to let your students know where they can go for access" so that they can fully participate in the CS opportunities you create.*

[6]

Identify desired student learning outcomes and create an assessment plan

To clearly communicate the academic value of computer science as a discipline and generate interest in it, you'll need to first identify and define the student learning outcomes you will associate with CS education experiences and determine the evidence (assessments, work products) you will use to measure progress toward those outcomes. In the words of one school leader, *"What's the outcome of a successful program?... What should kids be working on? When you walk into a classroom what does it look like when students are highly engaged in meaningful activity?"*

[7]

Review available instructional materials and identify equipment needs

CS instructional resources vary widely. Some provide detailed information for teachers, like specific instructional approaches, questions and activities to [engage students](#), while others identify broad content topics to cover with the expectation that teachers will find or create lesson activities and student materials themselves. School leaders and teachers must explore these options carefully to select materials that fit the program scope, the desired learning outcomes, and teacher needs. It is also important to continually re-evaluate the materials. As one school leader noted, *"Principals need to make sure whoever is teaching the course understands that the curriculum is ever evolving. They need to decide whether they are going to revise the curriculum every two, three years...are the projects that are taught to students current and reflect the type of technology they interact with, the more recent cases or articles in the news, or anything current?"*

[8]

Prepare a realistic budget and allocate resources

Plan for space, equipment, and infrastructure and inventory your existing resources to see if you already have what is needed to implement an introductory CS course. Your budget should also account for technical support and an administrator or teacher leader to oversee the program. And, be sure to create a plan to sustain these resources beyond initial start-up. As one school leader pointed out, *"When something is broken or doesn't work right, you need to have a designated person to make [any] adjustments otherwise you have really great intentions but if the computers aren't working then neither are the teachers, the students, or whatever the programming is that you thought you designed out."*

[9]

Consider who will teach your CS classes and what credentials they need

According to one school leader, *"You have to have a teacher who knows the content, can engage students, and is excited about both the content and the students."* Ask yourself, do you have teachers in your school who can fill this role, or do you need to look elsewhere to staff your CS program? What are the credential requirements for your state? If you do not already have teachers, school leaders recommended advertising the position(s) widely, including posting to the [Computer Science Teachers Association \(CSTA\) job page](#),

"tweet[ing] out things to people [doing CS programs, like] [Girls Who Code](#)," and reaching out to other schools in the area offering CS programs to ask for advice and recommendations from their current CS teachers. One school leader considered this challenge by speculating, "is it physically possible for me to have a person in the building ...do this? if not, can I get someone that's part time maybe? Maybe I can offer one class. Maybe I have to do something after school as a club."

[10]

Identify a range of supports for the teachers who will teach CS

For all of your teachers, seek out professional development. Opportunities range from summer workshops, to meetup communities. The breadth and depth of supports needed will vary from teacher to teacher, depending on their prior experience with the content and classroom teaching in general. As one leader explained, "In some areas, including CS, you might have someone who is really excited about the content and knows the steps really well but doesn't know how to engage the students." Be ready to provide differentiated [support for the range of teachers who will teach your CS classes](#) through professional development and [instructional guidance](#). Consider [stakeholders within and outside your school community](#) who can help provide the different types of supports your teachers will need to excite students about CS and meet their learning needs.

[11]

Be flexible and patient

Introducing any kind of new program can be a bit rocky at first, and a new CS program is no different. But over time and with careful attention to continual improvement of your CS initiative, you will develop the right program for your school. One school leader expressed the importance of a longer-term program plan with improvement along the way: "let's try to work some kinks out and maybe it doesn't have the best implementation in the first year, but after the first year okay, let's make a few key changes to it and go for it again the second year, and go for it again the third year, and really--let's have a five-year plan instead of a one year see if it works and [then] throw it out the window plan."

ABOUT THE BASICS STUDY:

The Barriers and Supports to Implementing Computer Science (BASICS) study is a three-year exploratory research project funded by the National Science Foundation (#1339256) as part of the CS10K program – an ambitious effort to have 10,000 well-trained computer science (CS) teachers in 10,000 schools. The BASICS study seeks to contribute to this effort in part by informing policy, school, and stakeholder leaders about the supports and barriers to wide-scale introductory CS education in high schools and strategies for addressing them. Over the course of two years, researchers at Outlier Research & Evaluation at UChicago STEM Education | University of Chicago interviewed school leaders in three different school districts whose schools were using Exploring Computer Science (ECS) instructional materials. It is important to note that BASICS is not, in any way, an evaluation of ECS. Rather, the BASICS study focused on the ECS curriculum as an example because it is widely used to teach introductory high school computer science.

Outlier Research & Evaluation (Spring 2017). *School Leaders' Advice for Introducing Computer Science in Schools*. Chicago, IL; Outlier Research & Evaluation at UChicago STEM Education | University of Chicago. Retrieved from <http://outlier.uchicago.edu/basics/findings/Advice-fromSchoolLeaders/>

This work is supported by the National Science Foundation, Study # 1339256

outlier.uchicago.edu/basics/findings