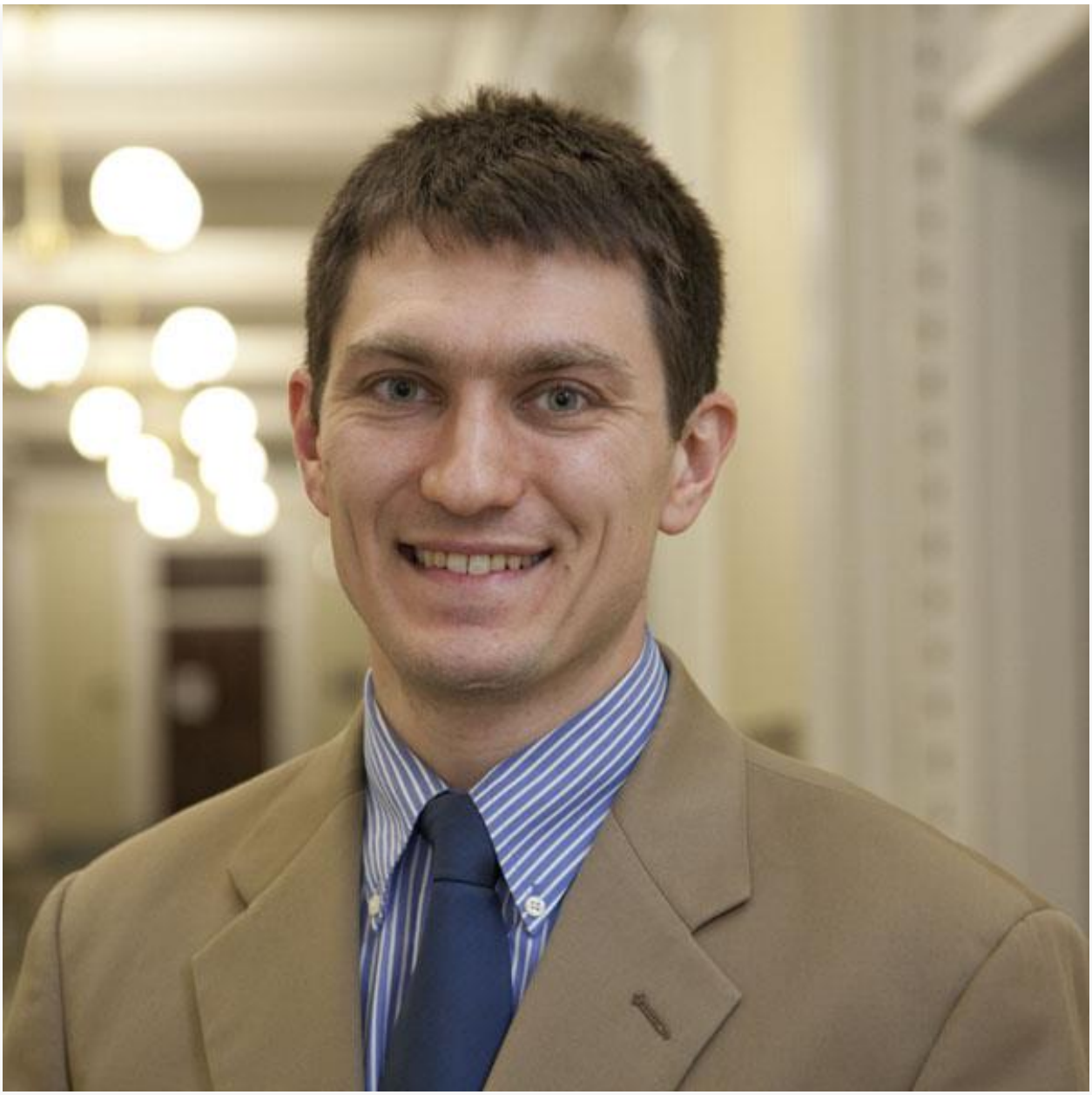
Teacher Leader Pathways



Making Computer Science Happen: One Teacher's Story

BAKER FRANKE

Former Department Chair, University of Chicago Laboratory School. Currently, Education Program Manager, Code.org

When Baker was a computer science teacher at the University of Chicago Laboratory School, the school "sort of" had computer science. There was one introductory elective course that typically had about ten students in it. Believing that all students in the school needed to learn some computer science, Baker set out to make that happen. Here are five key steps that he took to bring computer science to all students.

1. **Make AP Computer Science an official course:** Baker's first step was the addition of AP Computer Science A as an official high school course offering at the school. He taught the course himself and about 10-12 students took it each year. This was the only course that Baker taught at that time. The rest of his time was spent monitoring the computer lab. This allowed Baker the opportunity to plan for broadening CS at his school. For help with arguments for why bringing computer science to a school is important, go to the Why Computer Science Education tool.
2. **Enlist the school counselors:** Baker quickly recognized the key role that the school counselors play in guiding students toward, or away from computer science classes. He took each of the counselors out to coffee and explained to them, "what CS is and what it is for." See the Promoting Computer Science tool for resources and information to help make the case for CS in your school.
3. **Set the stage for making computer science a requirement:** In Baker's school, he felt that the only way to get the students to take more CS would be to make it a requirement. He developed some arguments about why every high school student should take computer science before leaving high school and then set out to convince all of the necessary stakeholders, including the school's Board of Trustees and the principal. Learn from the Recommendations for Building Strong Partnerships and Steps for Starting Partnerships tools
4. **Answer the scheduling "zero-sum problem":** While his stakeholder audiences were receptive, Baker knew he had to answer what he referred to as the "zero-sum problem": where CS would fit in the school schedule. He decided to focus on 9th grade and after examining the students' schedules he found that due to a required freshman class that occupied the equivalent of 1 ½ regular classes in the schedule, there was a hole in the schedule (the remaining ½ class that most students used for study hall). He took that available time and filled it with a new, required computer science course. In this case, the ½ class was scheduled for 45 minutes, two times a week. In his high school, he solved the "zero-sum problem" by sacrificing a period that for most students was free time.
5. **Don't forget to get the students on board:** With all of these challenges solved, Baker taught the course for the first time and came up against some unanticipated resistance – from the students. Baker found that requiring CS meant many students weren't particularly pleased to be in the CS classroom. Many of them felt that they were at a disadvantage being in the same room with others who had more experience than they did. The resistance was so great, that he completely changed the course mid-year. Baker found that he had to begin the course with material that was familiar to no one, yet still engaging for all. This

meant not starting the course with programming, but with other topics in computing with which almost all students, regardless of background, were equally unfamiliar. This leveled the playing field for engagement with the material and allowed for a much more equitable ramp up to more sophisticated material.