

Code Together: Teaching Computer Science to Teachers and Students



Julia Orr
Museum Educator for Out of School Time
Intrepid Sea, Air & Space Museum
12th Avenue and West 46th Street
New York, New York 10036

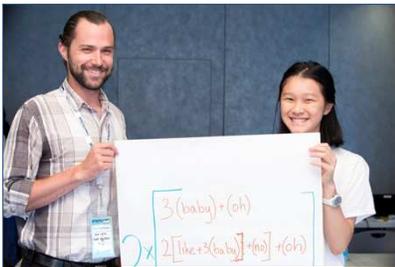


What is Code Together?

At the Intrepid Sea, Air & Space Museum, we've taken the rich history of navy codes (Semaphore, Morse, etc.) and brought it up-to-date with computer programming. The Code Together program partners teachers and students to learn computer science in 15 hours of professional development (over 5 sessions). As teachers learn to code, they also learn to work with students as colleagues and mentors.

At the end of the five workshops participants will:

- Gain knowledge of basic coding concepts, tools and skills
- Increase their interest in computer science
- Improve their confidence in their ability to explain and use computer coding



Challenges

As with many Professional Development courses, the biggest challenges are tardiness and attendance. To encourage participation in the full PD, we award teacher participants:

- A \$750 stipend (based roughly on the per-session pay rate for teachers in NYC)

- A Raspberry Pi 3 kit

Student participants receive:

- A Raspberry Pi 3 kit
- An opportunity to apply for a paid internship with Code Together

Intrepid Museum is a location-challenged museum, and 15 hours is not a lot of time to lay the foundation of computer programming, survey many different programming languages and tools, and complete at least two projects. Tardiness can affect a participant's ability to understand and complete all projects. To combat tardiness, we instituted a make-up work policy. If a participant is more than 15 minutes late, they will have a make-up project to work on at home and return before the next session. This ensures that participants are making up for topics and time they miss.

Why Code Together?

Partnering Teacher & Students



To foster relationship building between teachers and students in Code Together, we pre-assign teachers and students to share a table each day and use several ice breaker and group activities related to coding. These activities encourage everyone in a group to participate and share ideas. Some of these activities include:

- Discovering computational thinking by playing 20 Questions or SETS
- Pseudocode—practicing using and understanding conditional statements by creating rules for games
- Programming a person—giving clear instructions

Our success in relationship building is verified by observing teachers and students assisting each other with debugging, generating ideas for creating digital games and creating genuine relationships to take back to their schools.

Coding & Thematic Learning

Thematic learning humanizes computers. Rather than using code to create purposeless assignments, we show how programming can complement other disciplines' projects and standards. Some of the projects we use to teach computer programming in Code Together include:

- Building with Scratch to create
 - Digital exhibits,
 - Narrative stories using artifacts, and
 - Informative stories about *Intrepid's* history
- Writing in Python to
 - Improve ELA skills by creating MadLibs-like games
 - Practice using random number theory and probability by creating card and dice games
- Using affordable mini-robots (Ozobots) and coding using markers
- Flying drones by programming them with Tynker

We have a website that shares lesson plans and new ideas with teachers who attend Code Together:

<https://sites.google.com/view/codetogether>



Evaluation

Code Together is funded through 100kin10, and as part of a Networked Improvement Community (funded and organized through 100kin10), we are analyzing the best methods of professional development learning in engineering and computer science.

Our project's aim is that **teachers increase their sense of self-efficacy in integrating Computer Science or Engineering Design Process concepts into core subject areas in their teaching contexts.**

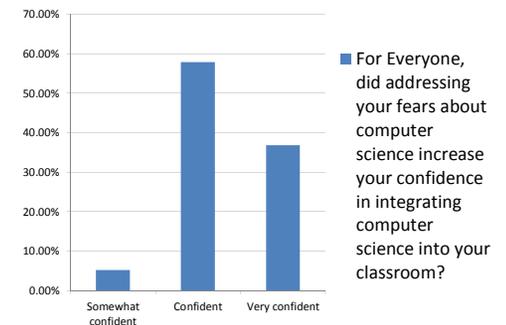


Table 1: Preliminary results from Question 1 (n=19)

To compare programs, we currently analyzing:

1. Whether [individual project's change idea] increases participants' confidence in integrating computer science into the classroom
2. Participants' confidence in their abilities to:
 - Integrate CS/Engineering standards into my curriculum
 - Engage all students in CS/Engineering lessons
 - Incorporate knowledge from CS/Engineering professional development into my classroom.
 - Sustain and improve upon what I have learned about CS/Engineering in my classroom
3. Participants' satisfaction with the instructional resources related to computer science available to them