

Welcome to Computer Science for All & SEPjr

In September 2015, Mayor Bill de Blasio announced the [Computer Science for All](#) Initiative, an initiative that will provide computer science (CS) education to every public school student by 2025. In an effort to support this goal in elementary schools, the CS4All team is proud to offer the [Software Engineering Program Jr.](#) (SEPjr), a comprehensive CS program for kindergarten through fifth-grade students that includes high-quality curriculum, training, and ongoing teacher and school leader support.

The goals of SEPjr are to (1) increase the number of elementary students, particularly from traditionally underrepresented groups, who learn computer science, (2) develop students' computational thinking and problem-solving skills in real-world contexts, and (3) enable students to engage in computer science instruction in each year of elementary school. The SEPjr curriculum is designed to provide *all* students in the school with deep exposure to CS concepts and practices through a "[sequence](#)" of CS experiences that build on each other from year to year, thus offering a truly rich elementary school CS experience that exceeds the CS4All goal of a single meaningful unit.

Why is Computer Science Important?

Computer Science is everywhere! Not only is computer science required for most modern careers and fields of study, it is also fun! Learning the basics of computer science prepares students for a world that is increasingly dominated by technology.

Computer science is also where many jobs are. Over 50% of all STEM (science, technology, engineering, and math) jobs deal with computing. Students with computer science degrees are some of the highest-paid college graduates, and computer programming jobs are growing at two times the national average—but there aren't enough graduates to fill these jobs!

When you enter a computer science classroom you may find students:

- Working together to solve problems
- Writing code and adapting existing code to their own projects
- Working with teachers and peers to troubleshoot code
- Building physical prototypes as part of the design process
- Participating in unplugged activities (that do not use technology), to introduce them to computer science fundamentals

Computer science gives students exposure to:

- Robots & Robotics
- Block-based and text-based programming
- Data and data science
- Physical computing, including Maker Events
- Game design and development

About PS 181Q Computer Science Units

This school year, students will focus on the Computer Science Units below.

Computer Science Fundamentals (All Grades) – The Computer Science Fundamentals Unit will provide the foundation for student understanding around computational thinking and computer science concepts. Concepts will include but are not limited to sequence, debugging, repeat/loops, events and conditionals, with upper elementary learning about variables and functions. Students will develop computational thinking skills such as decomposition, pattern matching, abstraction and automation.

Robots – (All Grades) Programmable robots provide students an opportunity to demonstrate understanding of what was learned in the CS Fundamentals Unit. Beginning with virtual robots and then moving to physical robots, the unit will include activities such as programming robots to complete mazes, obstacle courses, tell stories and navigate maps.

Problem Based Learning (Grades 3–5) – Beginning with a challenging, problem or question the Project Based Learning Unit will provide students an opportunity to integrate computational thinking and practices into other content areas through digital storytelling, animation and game design. Students will be given the opportunity to investigate, collaborate and create solutions while having an authentic experience of the design process including testing, debugging, iterating and remixing. Content connections can correlate to Black History Month, Women’s History Month, Poetry Month and National Inventors Month. Through presenting projects students will be able to form a better understanding of computer science and other content areas. As well as develop communication skills and teamwork.

Physical Computing (Grades 3–5) – The The Physical Computing Unit will provide students an opportunity to design, create and build physical devices such as game controllers, sensors and simple machines with microcontrollers, which students can then program using specific applications as a solution to specific challenges.

Resources for Parents/Guardians

Computer Science Standards:

- [CS4ALL Concepts & Practices](http://blueprint.cs4all.nyc)
(blueprint.cs4all.nyc)
- [K12 Computer Science Framework](http://k12cs.org)
(k12cs.org)
- [CSTA Standards](http://csteachers.org/page/standards)
(csteachers.org/page/standards)
- [ISTE Standards](http://iste.org/standards) (iste.org/standards)

Computational Thinking:

- [Computational Thinking w/ Scratch](http://scratched.gse.harvard.edu/ct/)
(<http://scratched.gse.harvard.edu/ct/>)
- [Google Course: Computational Thinking for Educators](http://www.google.com/courses/learning-computational-thinking-for-educators/)
- [ISTE: Computational Thinking for All](http://tinyurl.com/SEPjrNYC)

Books:

- [Mindstorm by Seymour A. Papert](http://www.mindstorms.com/)
- [Code by Charles Petzold](http://www.amazon.com/Code-Charles-Petzold/dp/0130264643)
- [Student Reading List](http://www.studentreadinglist.com/)

Unplugged Activities:

- [CS Unplugged](http://csunplugged.org/) (csunplugged.org)
- [Computer Science In A Box](http://www.computer-science-in-a-box.com/)

Online Activities:

- [Hour of Code](http://hourofcode.com/us/) (hourofcode.com/us)
- [Blockly Games](http://blockly-games.appspot.com/)
(blockly-games.appspot.com)
- [Scratch](http://scratch.mit.edu/tips/) (scratch.mit.edu/tips)
- [SEPjr Curriculum](http://tinyurl.com/SEPjrNYC)
(tinyurl.com/SEPjrNYC)