

**TRUMBULL PUBLIC SCHOOLS**  
**Trumbull, Connecticut**

**Computer Programming: Coding**  
**Grades K-2**

**Draft for Pilot 2016**  
*(New Course)*

**Curriculum Writing Team**

Lindsay Carley, Technology Integration Specialist  
Jenell Cunningham, Technology Integration Specialist  
Kimberly Greco, Technology Integration Specialist  
Todd Richard, Technology Integration Specialist  
Carol St. John, Technology Integration Specialist  
Brenda Windsor, Technology Integration Specialist

Jonathan S. Budd, Ph.D. Director of Curriculum, Instruction & Assessments

Christina Hefele, Director of Digital Learning

*Draft for Pilot 2016*

## CORE VALUES AND BELIEFS

The Trumbull School Community engages in an environment conducive to learning which believes that all students will **read and write effectively**, therefore communicating in an articulate and coherent manner. All students will participate in activities **that present problem-solving through critical thinking**. Students will use technology as a tool applying it to decision making. We believe that by fostering self-confidence, self-directed and student-centered activities, we will promote **independent thinkers and learners**. We believe **ethical conduct** to be paramount in sustaining the welcoming school climate that we presently enjoy.

Approved 8/26/2011

## INTRODUCTION & PHILOSOPHY

Computer science drives innovation in the U.S. economy and society. Despite growing demand for jobs in the field, it remains marginalized throughout the U.S. K-12 education system. Many 21<sup>st</sup>-century jobs will require students not just to have computer literacy skills, which focus more on using existing technologies (e.g., word processing, spreadsheets), but also to be able to create and adapt new technologies and solve real-world problems.

Computer coding – the creation of instructions in language that a computer understands – expands learners’ understanding of the basic language that facilitates gaming, instructional applications, and many other functions of daily life. Some of the fundamental skills students acquire while learning to code include logical thinking, problem solving, persistence, and collaboration. The earlier we introduce children to coding, the more comfortable they will be when presented more in-depth learning opportunities in middle school and high school. The introduction of computer science is increasingly necessary for community improvement, students’ future opportunities, and local and national innovation.

Trumbull Public Schools acknowledges the importance of providing these opportunities to our students at an early age. It is also our belief that we want to teach these technological skills not in isolation, but aligned to our core curriculum areas. The goal of the elementary coding program is an eight- to ten-week mini-course that teaches the fundamentals of computer science through our core curriculum. Computer coding will be taught in a progressive sequence leading toward complex computer coding languages. At each grade level, computer programming, or coding, will begin with a basic foundation of what computer coding is, why it is important to learn, and where it is used. Children in grades K-2 will be exposed to various applications of computer coding, be introduced to basic coding vocabulary, and create a simple app to help teach the alphabet or basic numeracy. Classroom activities will facilitate the development of unplugged and plugged-in computer programming activities. In order to enhance learning and reinforce skills and rudiments, technology through the use of computers and iPads will be integrated. Original designs and creations will be highlighted prominently throughout. Essential Questions will be applied through these original creations and culminate in a final project at each grade level featuring newly-gained computer coding skills.

As students progress through higher levels of computer coding, they will begin to find ways to use applications to improve their environment, provide support for others, and encourage creativity.

*Draft for Pilot 2016*

Lessons will teach students logical reasoning, algorithmic thinking, and structured problem-solving – concepts and skills that are valuable in any discipline. Students need to first think about what they are trying to do and then plan it out. They have to reason, communicate, plan, and develop a hypothesis. They then will test ideas on the computer to see if they will work.

The curriculum aligns practices, standards, and assessments among the Trumbull Public Schools, the Connecticut State Board of Education, the Computer Science Teachers Association, and the International Society for Technology Education.

## **PROGRAM GOALS**

The following standards derive from the 2016 International Society for Technology in Education Standards.

ISTE Empowered Learner (Standard 1c)	Students use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.
ISTE Innovative Designer (Standard 4a)	Students know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts, or solving authentic problems.
ISTE Computational Thinker (Standard 5d)	Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.
CCSS.ELA-Literacy.W.CCRA.6	Students use technology, including the Internet, to produce and publish writing and to interact and collaborate with others.

# Grades K-2: Coding: The Fundamentals of Computer Science

## Unit Essential Questions

- What are the fundamentals of computer programming?
- How do I create a program?
- How can I use computer programming to complete a task?
- How is computer programming useful in real life?

## Scope and Sequence

Each number below corresponds to one class session.

1. Each student will list the steps necessary to move a character around a map, arrange directions to reach a predetermined goal, and predict results given a list of steps.
2. Each student will predict moves necessary to get a teammate from start to finish, and will convert movements into symbolic instructions.
3. Each student will use the computer mouse to drag and drop information. Recalling and applying rules of pair programming, each student will use pair programming to complete collaborative tasks without a computer. Each student will also identify situations where the rules of pair programming are not followed, and will arrange puzzle pieces in the proper order.
4. Each student will decompose large activities into series of smaller events, arranging sequential events in their logical order.
5. Each student will create a program to draw a shape using sequential steps.
6. Each student will combine different motion blocks into programmed sequences, following sequential steps verbally and visually.
7. Each student will program a character to start when a green flag is touched, will use the end block to signify the end of a program, and will choose a new character based on a previously written story.
8. Each student will use numbers on motion blocks, use repeat and repeat forever blocks to make a program repeat, record sounds and add them to projects, and create speech bubbles for characters based on the previously written story.
9. & 10. Each student will apply concepts from prior lessons to animate a previously written story the student has created as part of Writing Workshop.

## Assured Assessment

The animated story will apply concepts from prior lessons to bring a student's previously written story to life. Each student will use basic programming blocks and concepts taught to design the setting, create the characters, sequence the story, narrate the story, and develop cause and effect actions driven by computer commands. Students will publish and share their final projects.

## Time Allotment

- Approximately ten class sessions of 40 minutes each

*Draft for Pilot 2016*