Programming Each Other Challenge

Part of the *A World In Motion* Series

Developed by Education Development Center, Inc.

Funded by SAE International

Field Test Draft 2

Teacher Guide

TABLE OF CONTENTS

Reproducible masters and visual aids are shown in *italics*.

A Need for STEM Education

For many years the United States has been a leader in the science and technology fields. However, recent studies suggest that student preparedness in science and technology, and how our country educates our youth in these fields, is not keeping up with other developed areas of the world.

Here are some alarming statistics:

- Thirty-two percent of eighth-graders scored below basic, 34% scored at or above basic, 32% scored at or above proficient, and only 2% scored at or above advanced in the National Assessment of Educational Progress (NAEP) science test.[1]
- Only 2% of high school seniors scored at the advanced level in NAEP science.[1]
- In the United States, only 5% of bachelor's degrees awarded are in engineering, versus 20% in Asia.[2]
- In 2013, 37% of the doctoral degrees in science and engineering fields in the United States were awarded to temporary residents.[3]
- Among U.S. bachelor's degree students entering science, technology, engineering, and math (STEM) fields between 2003 and 2009, nearly 48% had left these fields by spring 2009.[4]
- Natural sciences and engineering fields account for a much larger proportion of all bachelor's degrees awarded in China than in the United States. In 2010, these fields accounted for 44% of all bachelor's degrees awarded in China, compared with 16% of all bachelor's degrees awarded in the United States.[5]

Introducing SAE International's Pre-Professional Programs

Philosophy and Approach

SAE International is a global association committed to being the ultimate knowledge source for the engineering profession. By uniting more than 138,000 engineers and technical experts, we drive knowledge and expertise across a broad spectrum of industries. We act on two priorities: encouraging a lifetime of learning for mobility engineering professionals, and setting the standards for industry engineering.

---

[1] The Nation's Report Card. (2015). *2015 Science Assessment.* Retrieved from https://www.nationsreportcard.gov/science_2015/ - acl/chart_loc_1?grade=8

[2] National Science Board. (2010). *Science and Engineering Indicators 2010.* Arlington, VA: National Science Foundation. Retrieved from https://wayback.archive-it.org/5902/20160210151754/http://www.nsf.gov/statistics/seind10/

[3] National Science Board. (2016, January). Chapter 2: Higher Education in Science and Engineering. *Science and Engineering Indicators 2016* (NSB-2016-1). Retrieved from https://www.nsf.gov/statistics/2016/nsb20161/ -/report/chapter-2/graduate-education-enrollment-and-degrees-in-the-united-states

[4] Chen, X. (2013). *STEM Attrition: College Students' Paths Into and Out of STEM Fields* (NCES 2014-001). Washington, DC: National Center for Education Statistics, Institute of Education Sciences, U.S. Department of Education. Retrieved from https://nces.ed.gov/pubs2014/2014001rev.pdf

[5] National Science Board. (2014, February). *Science and Engineering Indicators 2014 Digest.* Retrieved from https://www.nsf.gov/statistics/seind14/index.cfm/digest

To build a solid foundation for a well-prepared STEM-literate workforce, SAE has developed programs to engage students at the pre-professional education level. We are the only organization of our kind to offer a full continuum of STEM engagement opportunities for students in kindergarten through graduate-level education. Our educational programs are designed to provide a *Total STEM Solution* for student engagement, participation, and achievement in STEM (K–16), starting with an early focus on scientific literacy at age 5.

The 2011 National Research Council (NRC) report *Successful K–12 STEM Education: Identifying Effective Approaches in Science, Technology, Engineering, and Mathematics* recognized the important foundation laid in elementary school. The report set out three goals for U.S. education:

- Expand the number of students who ultimately pursue advanced degrees and careers in STEM fields, and broaden the participation of women and minorities
- Expand the STEM-capable workforce, and broaden the participation of women and minorities
- Increase science literacy for all students

SAE has the longitudinally research-based programs to achieve these goals and build the STEM workforce of the future—and, even more notably, we have the tools to build the STEM-literate society of the future.

Research shows that by the time students enter the fourth grade, one-third have lost interest in science. By eighth grade, almost 50% percent have lost interest in science or have deemed it irrelevant to their education or future plans.[6] While nearly 28% of high school freshmen proclaim interest in a STEM-related field each year, by graduation, more than 57% will have lost interest.[7] SAE has the tools to correct these dreadful statistics and to build the scientifically literate society of the future.

SAE International STEM programing is designed to provide resources to move students through the STEM pipeline with a mission of ultimately engaging them in STEM careers. The following diagram overlays SAE's educational programing (including *A World In Motion*) with an intent to engage and build a foundation for future STEM learning.

A World In Motion

Our programing starts with *A World In Motion* (AWIM) (K–8). Supporting the educational process outlined in NRC's 2012 release of *A Framework for K–12 Science Education: Practices, Cross-Cutting Concepts and Core Ideas*, SAE's AWIM program is designed to offer age-appropriate design challenges built around the AWIM *Engineering Design Experience*. This process provides the structure needed to deliver teacher-led (and STEM industry volunteer-assisted) instruction for all students.

AWIM is designed to engage all students in the traditional classroom setting, and it is aligned with Common Core and Next Generation Science Standards. This design does not require classroom

[6] Murphy, A. (2011, August 29). STEM Education—It's Elementary. *U.S. News & World Report.* Retrieved from https://www.usnews.com/news/articles/2011/08/29/stem-education--its-elementary
[7] My College Options & STEM Connector. (2012). *Where Are the STEM Students?* Retrieved from http://www.dailyherald.com/assets/pdf/DA127758822.pdf

teachers to "do more" in the classroom; rather, it provides the structure they desperately need to deliver high-quality STEM instruction that separates itself from one-off "classroom experiments" that might be engaging but do little for long-term learning and engagement. This highly structured teacher-led experience prepares students for the next step in their educational experience through SAE International's pre-professional programming.

Collegiate Design Series

SAE's *Collegiate Design Series* (CDS) provides world-class student-led, hands-on learning experiences to develop and train the next generation of mobility engineers. Students participate in a diverse menu of competitions that includes Aero Design, Clean Snowmobile Challenge, Formula SAE, Formula Electric, Baja SAE, and Supermileage. All competitions within CDS prepare undergraduate and graduate engineering students in a variety of disciplines for future employment in mobility-related industries by challenging them with a hands-on team engineering experience that also requires budgeting, communication, project management, and resource management skills—the top skills most valued by today's innovative organizations.

SAE's CDS program participation is embraced by the *Top 100 Engineering Degree Producing Colleges/Universities*, as reported in the American Society of Engineering Education's *Profiles of Engineering and Engineering Technology Colleges*. Presently, 100% of the Top 50 participate in one or more of SAE's CDS programs.

A Total STEM Solution

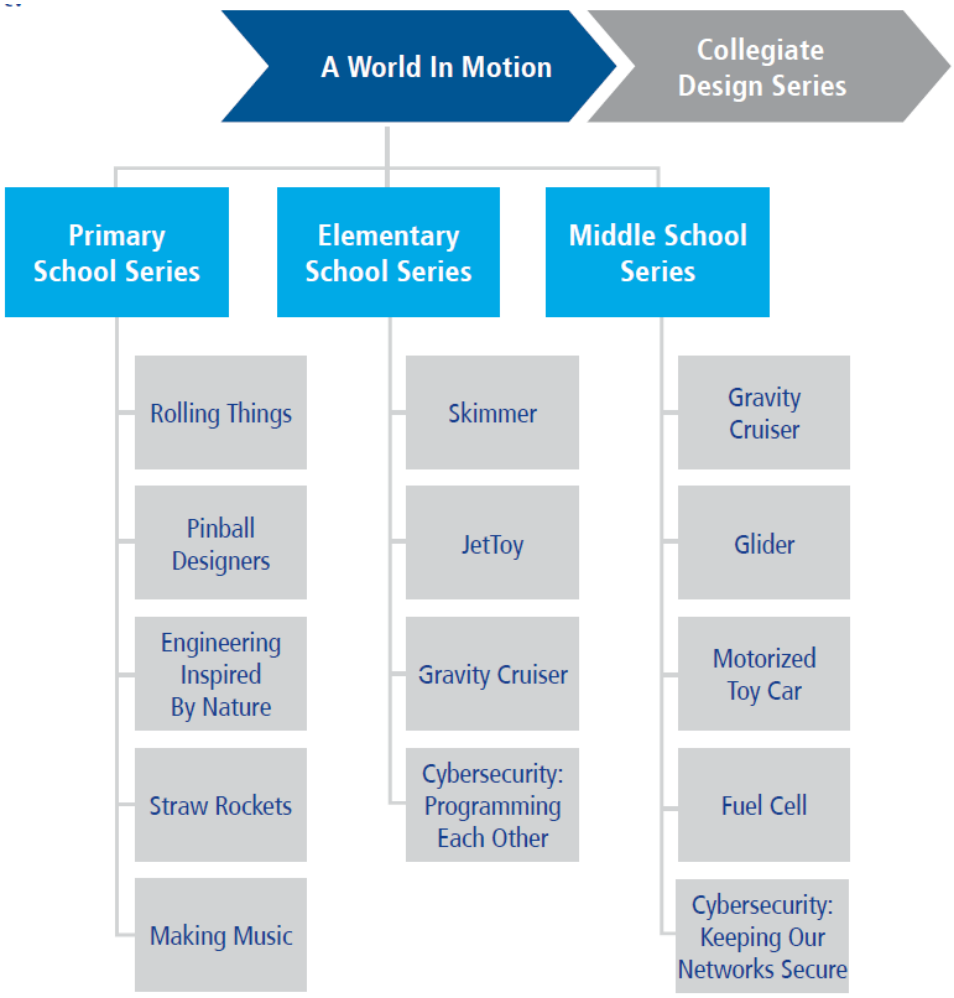
We realize that many diverse professionals are needed for a strong society as a whole; however, all members of our society must be scientifically literate. For the long-term success of our communities, we must focus on developing our science and technology workforce at an early age and then continue to foster and build it. SAE International's *Total STEM Solution* is building the next generation of science and technology professionals *as well as* the strong and scientifically literate society in which that future STEM workforce will exist.

The Scope of *A World In Motion*

To succeed in the society of tomorrow, all children need STEM education that prepares them to understand and apply concepts (i.e., become literate) in science, mathematics, and technology. In addition, students must learn to solve complex problems, communicate clearly, raise and resolve questions, assimilate information, and work cooperatively toward common goals.

To help our students acquire a deep understanding of scientific, mathematical, and technological phenomena, today's educators can no longer simply present students with scientific information and teach them rote processes. Instead, teachers must provide abundant opportunities for direct, hands-on experiences with materials and tools, which foster students' competence and confidence in their abilities to explore, conjecture, and reason logically. Having students gather and manipulate pertinent information encourages them to learn about the world around them, particularly when their school activities grow out of real-life problem situations; their skills are further stimulated and developed through the interactive, cooperative processes of discussing, reading, and writing about direct experiences.

SAE developed AWIM as an opportunity for students and teachers to explore science, mathematics, and technology in an engineering design context.



Primary School Series

At the primary school level, the student experience should focus on building a foundation for STEM learning and creating excitement about science. It should include the following:

- Guided opportunities to question ideas and define problems
- Using literature to facilitate questioning of concepts and ideas
- Play and guided experimentation for investigation
- Building physical models
- Manipulating variables
- Collecting, recording, and analyzing data
- Building tables and graphs
- Making predictions
- Designing solutions
- Pair-share and group discussions
- Communicating ideas
- Turn-and-talk strategies (partner interaction)
- Sharing and interpreting (whole group)
- Presenting a solution

The primary school challenges all include a literacy component: an age-appropriate story that features real-world science concepts. Each primary school challenge is briefly described below.

	Students explore the story *The Three Little Pigs Sledding* Adventure while they study toy cars and car performance. Launching the cars from ramps, the students investigate the effects that different ramp heights and car weights have on distance traveled, measuring and recording data gathered through variable testing.
	After reading *Malarkey and the Big Trap*, students design a homemade pinball game and explore the behavior of the different components (the pinball, ball traps, and bumpers). Students test the launch ramp to explore how launch position affects the behavior of the pinball and then learn how to optimize their games to make them more challenging and interesting.
	Students investigate methods in which seeds are dispersed in nature through the story *Once Upon a Time in the Woods*. The story leads the students to further explore how seeds are dispersed by the wind. Using the designs found in nature, students develop paper helicopters and parachutes, then perform variable testing to improve their performance.
	Students explore the early life of Dr. Robert Goddard while reading his biography, *The Rocket Age Takes Off*. After investigating Goddard's early trials and tribulations in creating the first liquid-fueled rocket engine, students begin to uncover the work necessary to optimize a design with the goal of creating a straw rocket that flies the farthest and highest.
	Students read about the sounds of nature in *Sleep Soundly at Beaver's Inn*, explore sound and vibrations, and learn how the human eardrum works. After collecting information through hands-on lessons, students engineer a musical instrument according to specific criteria.

More information on the Primary School Series can be found on the AWIM website (http://www.awim.org/curriculum/primary/).

Elementary School Series

To address the steep decline in students' interest in science in the upper elementary school grades, AWIM created a set of challenges designed to not only excite students about science, but also to continue building a foundation of the science concepts introduced in the Primary School Series. The challenges in the AWIM Elementary School Series shift the student learning experience away from the children's literature focus at the primary level to incorporate a more structured research focus. Students explore AWIM's *Engineering Design Experience* while engaged in formal variable testing, data gathering, and decision making based on analysis. The student experience should include the following:

- Guided opportunities to question ideas and define problems
- Use of all stem subjects (and beyond) to solve design questions
- Play and guided experimentation for investigation
- Building physical models
- Manipulating variables
- Collecting, recording, and analyzing data
- Building tables and graphs
- Making predictions
- Designing solutions
- Working as an engineering design team
- Communicating ideas
- Presenting a solution

The Elementary School Series begins with the *Skimmer Challenge*, where students construct paper sailboats and test the effects of different sail shapes, sizes, and construction methods to meet specific performance criteria. Friction, forces, and the effect of surface area are some of the concepts students encounter in this challenge.

Next is the JetToy Challenge, which is a bit more complicated. Students make balloon-powered toy cars that meet specific performance criteria (e.g., the car travels far, carries weight, or goes fast). Jet propulsion, friction, air resistance, and design are the core scientific concepts students explore in this challenge.

The series moves on to Cybersecurity: Keeping Our Networks Secure (which is designed to be used at the upper elementary or middle school level), bridging the elementary school and middle school challenges. In this challenge, students investigate how the Internet is structured to fend off both physical and electronic attacks. Students participate in physical simulations of the movement of information over the Internet and then use code-making devices (scytales and cypher wheels) to explore how data on the Internet are made secure.

The series wraps up with the Gravity Cruiser Challenge (which is also designed to be used at the upper elementary or middle school level), where student teams design and construct a vehicle that is powered by gravity. A weighted lever connected to an axle by string rotates on its fulcrum; as the weight descends, it causes the axle attached to the string to rotate, propelling the cruiser forward. Concepts explored include potential and kinetic energy, friction, inertia, momentum, diameter, circumference, measurement, graphing, and constructing a prototype.

The main concepts covered throughout all of these challenges align with the Next Generation Science Standards and the Common Core Mathematics and English Language Arts Standards. Extension ideas are provided within the lessons for those students and classes looking to enrich the experience beyond the scope of the detailed curriculum in these manuals.

More information on the Elementary School Series can be found on the AWIM website (https://www.sae.org/learn/education/elementary-curriculum).

Middle School Series

AWIM Middle School Challenges focus on student STEM learning at the traditional seventh and eighth grade levels. Building on the research and analysis skills fostered in the previous levels, these AWIM challenges expand to include market research and data gathering beyond controlled scientific

testing. Students research market preferences through a survey-based approach and move beyond designing independent solutions based solely on performance criteria. This shift allows students to investigate the added level of design decisions to meet market and customer needs. The student experience should include the following:

- Guided opportunities to question ideas and define problems
- Use of all stem subjects to solve design questions
- Consumer/market research and targeting customers (social studies connection)
- Planning and preparing written proposals, oral presentations, and manuscript content (English language arts connection)
- Guided experimentation for investigation
- Building physical models
- Manipulating variables
- Collecting, recording, and analyzing data
- Building tables and graphs
- Making predictions
- Designing solutions
- Working as an engineering design team
- Communicating ideas
- Presenting a solution

In the Gravity Cruiser Challenge, students focus on understanding the relationships between the "sweep" of the lever arm, the number of winds the string makes around the axle, and the distance the gravity cruiser travels. They investigate how the diameter of the wheels, the diameter of the axles, and the amount of weight placed on the lever affect the gravity cruiser's speed and distance.

The Motorized Toy Car Challenge asks students to develop new designs for electric gear-driven toys. To meet a specific set of design requirements, students must write proposals, draw sketches, and work with models to develop a plan. Force and friction, simple machines, levers and gears, torque, and design are the core concepts covered.

In the Glider Challenge, students explore the relationship between force and motion and the effects of weight and lift on a glider. The activity culminates in a book-signing event, where each design team presents its prototype, and the class presents its manuscript of glider designs. Students learn the importance of understanding consumer demands and the relationships between data analysis and variable manipulations.

In the Fuel Cell Challenge, student teams design a toy car that uses a Proton Exchange Membrane fuel cell to power the electric motor. Teams explore the elements of electrical currents, "green" design, and transformations of energy as they develop their products.

Finally, in Cybersecurity: Keeping Our Networks Secure, students investigate the architecture of the Internet and how it was designed to withstand both physical and electronic attacks. Students explore a number of physical models that simulate the movement of information through the Internet, and then investigate the two basic components of securing data and systems: encryption and authentication.

More information on the Middle School Series can be found on the AWIM website (https://www.sae.org/learn/education/middle-school-curriculum).

The Science Process Skills

All AWIM challenges are designed to have students actively engage (independently and within groups) in procedures that involve the process skills associated with critical thinking, project management, communication, inquiry and analysis, and teamwork and collaboration. The design of AWIM addresses many process standards as students delve into learning experiences aligned to age-appropriate content standards. The different components of all AWIM challenges are outlined in the table below.

Components Common to All AWIM Challenges (K–8)				
Critical Thinking	**Project Management**	**Communication**	**Inquiry and Analysis**	**Teamwork and Collaboration**
▪ Assess a new situation ▪ Formulate questions for scientific analysis; generate and evaluate ideas ▪ Gather and interpret data ▪ Synthesize information and make predictions ▪ Integrate and apply learning	▪ Define goals and establish objectives ▪ Maintain portfolios or design logs of work ▪ Identify and set priorities ▪ Organize and present data ▪ Propose and test solutions	▪ Design a team name, logo, and slogan ▪ Develop and produce drawings and diagrams ▪ Prepare written and oral reports ▪ Prepare presentation aids and materials ▪ Deliver oral presentations	▪ Conduct formal testing ▪ Measure and record outcomes ▪ Make qualitative and quantitative observations ▪ Establish relationships among variables and make predictions ▪ Use the scientific method of experimentation, questioning, and trial/variable testing	▪ Participate as a member of a team ▪ Practice cooperation and compromise to reach group consensus ▪ Assign team member roles and responsibilities ▪ Understand group dynamics ▪ Evaluate team and individual performance

The STEM Industry Volunteer

The AWIM curriculum is unique in incorporating STEM industry volunteers into the classroom setting to assist the teacher. These volunteers not only serve as a content expert for the teachers (GRG, 2007), they also communicate the needs of the STEM workforce directly to students and provide a face for STEM careers. Students become problem solvers and communicators by learning from professionals who practice these skills daily in a STEM field, which facilitates students seeing themselves as future scientists or engineers. Empirical research, covered later in the paper, supports the success of this AWIM model.

SAE International is distinctive in that its membership base supports education through the practical application of STEM concepts, linking the classroom setting to industry and the requirements of the future STEM workforce. More information on being an AWIM volunteer can be found on the AWIM website (https://www.sae.org/participate/volunteer).

Overview

Introducing *A World In Motion*

Educating Children for Tomorrow's World

To succeed in the society of tomorrow, all children need an education that prepares them to understand and apply concepts in science, technology, engineering, and mathematics (STEM). In addition to becoming literate in these disciplines, students must also learn to solve complex problems, to communicate clearly, to raise and resolve questions, to assimilate information, and to work cooperatively toward common goals.

Today's educators can no longer succeed by presenting students with scientific information and teaching them rote processes. To help students acquire a deep understanding of scientific, technological, engineering, and mathematical phenomena, teachers must provide students with abundant opportunities for hands-on experience with materials and tools. In this way, students become competent and feel confident in their abilities to explore, conjecture, and reason logically, and to gather and manipulate information to arrive at useful knowledge about the world around them. These abilities are nourished and nurtured when school activities grow out of real problem situations, and they are further stimulated and developed through the interactive, cooperative processes of discussing, reading, and writing about direct experiences.

SAE International has developed the *A World In Motion* (AWIM) challenges as an opportunity for students and teachers to explore STEM.

What Is Programming?

Programming is the act of writing a series of instructions that control the operation of a computer or other machine. Simple programs can look similar to instructions for building furniture, walking to school, or baking a cake. Programs can consist of lots of parts pieced together to solve fairly complex problems.

Good programmers harness a particular set of skills, called **computational thinking** (CT) skills, when developing and writing programs. Jeannette Wing, currently the Avanessians Director of the Data Sciences Institute at Columbia University, and former Corporate Vice President of Microsoft Research, is credited with elevating the discussion in 2006 with an essay titled "Computational Thinking," where she defined CT as follows:

> Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science.[8]

[8] Wing, J. M. (2006, March). Computational thinking [Viewpoint]. *Communications of the ACM*, *49*(3), 33–35.

Her primary goal was to promote the idea that computer science provided essential problem-solving skills to fields beyond those that had historically been linked to computer science. In the decade since, many educators have worked to make computer science more accessible to students across grades K–12, and a clear definition of CT has emerged. Specific standards related to CT from the Computer Science Teachers Association are detailed below. This unit focuses on components of CT in a number of ways:

- Encouraging students to write **algorithms**—detailed step-by-step instructions that are clear, concise, and unambiguous. Computer language often forces a certain level of precision on a programmer, but good programmers have a keen ability to clearly lay out the steps required to perform a task before they even begin to write code. Attention to detail and the ability to test algorithms even before committing to specific code can save a programmer a lot of time and anxiety when testing programs.
- Challenging students to break down complex problems into smaller parts (often called *problem decomposition*). Large, complex problems can often be broken down in a number of ways, and a good computer software architect will take into account the resources available (including the processing power of the computers and the speed of the machinery that is part of the solution) to seek an optimal solution that performs the task in the least amount of time or for the lowest cost.
- Preparing students to think *abstractly* by stripping away details of a problem that are not relevant and focusing on the core concepts. Some programmers start by stripping away even essential details in order to get a handle on a small, easily understood problem and its underlying solution. They will then restore details and adjust the complexity of their program systematically.
- Asking students to consider how using **automation** to solve a problem (e.g., with a computer or a robot) may differ from how a human may solve it. Humans will tire of performing tedious tasks long before a robot would, so some humans may choose ways to solve problems that wouldn't be the most efficient for a robot. Further, humans have a certain flexibility of thinking that cannot always be easily programmed into a computer.
- Extending students' solutions to include *parallelization*, a process by which multiple resources (e.g., several robots instead of just one) are used to carry out a task. When more than one part of the solution can be carried out at the same time, students will be faced with more than one way to break down the tasks. By considering the different ways that these tasks can be accomplished in parallel, they start to think about how to make their algorithms more efficient.

Other key aspects of CT, such as analyzing data and developing models and simulations, are touched on throughout the unit. The backbone of the unit—creating a comprehensive algorithm for sorting socks—can be thought of as a metaphor for the process of manipulating and analyzing data. Other activities, such as making sandwiches, may require students to develop a model (using index cards instead of bread, coloring with a marker instead of spreading peanut butter, etc.) to test their code.

What Is the Programming Each Other Challenge?

In this challenge, students receive a letter from a fictional publisher, CodeWorks Publishing, asking for their help with a book the company is publishing to introduce young children to the idea of programming. The book tells the story of an absent-minded programmer and the funny things that happen when he gives poorly thought-out instructions to his robots. The publisher asks students to write sample programs to help a robot accomplish everyday tasks—both "absent minded" programs and well-thought-out ones—for the book.

Throughout the unit, students look at the basic challenge of programming a robot to sort socks. The complexity of the task increases with each successive activity: Students start with a small number of identical socks, and then the number increases and different colors are added. Each incremental increase in complexity is designed to introduce specific programming concepts, such as *loops* (repetition of sections of code), *conditionals* (branches in code, typically based on an "if-then" statement, where different steps happen depending on a particular observation), *variables* ("buckets" that temporarily hold pieces of data for the program to interact with), and *general error checking* (what to do if a program encounters an unexpected type of data). Each activity also includes opportunities for students to explore and apply the concepts in different ways.

Teacher Question 1: Does this analogy for variables work for you? How could we improve it?

Introduction to the Unit

Objectives for the Programming Each Other Challenge

The primary objective of this challenge is to engage students in CT skills. Specifically, by the end of the unit, students should be able to do the following:

- Explain the need to write clear, concise instructions
- Recognize patterns in instructions and use shorthand (loops) to shorten a list of steps
- Use conditions within instructions and explain how these conditions redirect instructions
- Identify the *input* and *output* of a program
- Explain the distinction between a variable and its value
- Decompose problems into smaller parts
- Decompose problems in a way that leverages parallelization, and explain why it is important to do so

Correlation with the 2017 Computer Science Teachers Association (CSTA) K–12 Computer Science Standards

This unit addresses the following CSTA K–12 CS Standards:

		ACTIVITIES									
STANDARD		1	2	3	4	5	6	7	8	9	10
Algorithms											
1A-AP-08.	Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. (P4.4)	X	X	X	X	X	X	X	X	X	
1B-AP-08.	Compare and refine multiple algorithms for the same task and determine which is the most appropriate. (P6.3, P3.3)		X	X	X	X	X	X	X		
Variables											
1A-AP-09.	Model the way programs store and manipulate data by using numbers or other symbols to represent information. (P4.4)		X	X	X	X	X	X	X		
1B-AP-09.	Create programs that use variables to store and modify data. (P5.2)						X				
Control											
1A-AP-10.	Develop programs with sequences and simple loops to express ideas or address a problem. (P5.2)			X	X	X	X	X			
1B-AP-10.	Create programs that include sequences, events, loops, and conditionals. (P5.2)				X	X	X	X			
Modularity											
1A-AP-11.	Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. (P3.2)	X	X	X	X	X	X				
1B-AP-11.	Decompose (break down) problems into smaller, manageable subproblems to facilitate the program						X				

	ACTIVITIES									
STANDARD	1	2	3	4	5	6	7	8	9	10
development process. (P3.2)										
1B-AP-12. Modify, remix, or incorporate portions of an existing program into one's own work to develop something new or add more advanced features. (P5.3)			X	X	X	X				
Program Development										
1A-AP-12. Develop plans that describe a program's sequence of events, goals, and expected outcomes. (P5.1, P7.2)		X	X	X	X	X	X			
1B-AP-13. Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences. (P1.1, P5.1)			X	X	X	X	X	X		
1A-AP-14. Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops. (P6.2)	X	X	X	X	X	X	X			
1B-AP-15. Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. (P6.1, P6.2)	X	X	X	X	X	X	X	X		
1B-AP-16. Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. (P2.2)								X	X	
1A-AP-15. Using correct terminology, describe steps taken and choices made during the iterative process of program development. (P7.2)			X	X	X	X	X			
1A-AP-17. Describe choices made during program development using code comments, presentations, and demonstrations. (P7.2)							X	X	X	X

Source: Computer Science Teachers Association. (2017). *CSTA K–12 Computer Science Standards, Revised 2017.* Retrieved from http://www.csteachers.org/standards

Correlation with the 2013 Next Generation Science Standards

This unit addresses the following NGSS K–12 Standards:

	ACTIVITIES									
STANDARD	1	2	3	4	5	6	7	8	9	10
3-5 Engineering Design										
3-5-ETS1-1. Define a simple design problem reflecting a need or a want that includes specified criteria for success and constraints on materials, time, or cost.	X	X	X	X	X	X				
3-5-ETS1-2. Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.		X	X	X	X	X	X	X		
3-5-ETS1-3. Plan and carry out fair tests in which variables are controlled and failure points are considered to identify aspects of a model or prototype that can be improved.			X	X	X	X	X			

These materials also address, in part, the following crosscutting concepts:

- *Cause and Effect: Mechanism and Prediction:* Events have causes, sometimes simple, sometimes multifaceted. Deciphering causal relationships, and the mechanisms by which they are mediated, is a major activity of science and engineering.\
- *Systems and System Models:* A system is an organized group of related objects or components; models can be used for understanding and predicting the behavior of systems.

And the following Science and Engineering Practices:

- *Using Mathematics and Computational Thinking*

Source: NGSS Lead States/Achieve. (2013). *Next generation science standards: For states, by states.* Washington, DC: The National Academies Press.

Teaching the Challenge

To facilitate student learning, use the information in this section to organize your classroom. In this section, you will find techniques and tips for facilitating discussions, building student teams, using Science Notebooks, and assessing student learning, as well as information for obtaining basic sets of classroom materials.

Differentiating for Lower Grade Levels

This unit is written primarily for upper elementary school students (grades 4–6), though students in earlier grades can certainly make use of these materials with some tailoring. In general, lower-level classes can focus on the "Present the Activity" sections and choose to skip most of the "During the Activity" tasks. Writing tasks, including Science Notebooks, may need adjustment for lower reading levels.

Integrating Science and Literacy

The Programming Each Other Challenge allows students to develop important literacy skills, including speaking and representing their experiences and ideas through writing, drawing, and diagrams. By engaging in structured conversation with peers about their observations, plans, and conjectures, and keeping a Science Notebook of their actions and results, students are learning and practicing skills of both science and literacy.

Discussions

Talk is critical to conceptual learning. Just as children read to learn, they talk to learn. At the same time, they are building vocabulary, developing ways to express themselves, and learning to share and debate—all important skills. Talk takes place as children work with their peers in small groups, and you should hear a steady hum of talk as children engage in the hands-on activities in these challenges.

As you engage with small groups, be careful not to become the source of right answers or right ways of doing things. Avoid answering any of the students' questions directly. Encourage students to learn from their peers or from their own experiences, or suggest that you and they can work together to find out (particularly if you do not know the answer yourself!). When they ask, "How do I do this?"

ask them, "How could you figure this out for yourself?" or suggest, "Maybe Juan could help you." They will then learn how to rely on themselves and one another.

You may also want to encourage or challenge the groups with questions and comments such as the following:

- "I wonder what would happen if . . . ?"
- "How did you do that?"
- "Could you do it again?"
- "Another group did it this way. I wonder if you could?"

Students learn a great deal in both literacy and science when they come together to talk about their work as a large group. This allows them to practice an important part of scientific inquiry: comparing and debating their findings, new ideas, and conclusions just as scientists do. Frequent whole-class discussions help children see the relationship between specific activities and the challenge as a whole. They are also an important assessment tool.

In classrooms where whole-class discussions are already taking place in literacy, science, and/or mathematics, the skills of both teachers and students can easily be adapted to a computer science discussion. However, many teachers shy away from such discussions because they and their students would need to learn new skills and strategies to make these discussions successful. Following are some helpful tips for establishing and facilitating discussions in your classroom.

SETTING THE STAGE

Have students meet in a circle on the floor, or some other configuration where they can all see one another's faces, rather than sitting "audience style." They will pay more attention to one another this way.

Have physical props—one set of materials—to help focus the discussion and support students' description of a phenomenon they have observed or a conclusion they have reached. If students struggle with communicating their ideas, offer them the materials to use at that time.

Develop an explicit set of norms and expectations for the discussions (e.g., Don't talk when someone else is talking. Stay on focus. Listen to the speaker). The skills needed to follow these norms and expectations will need to be taught and practiced if students have never used them before. (Video-recording children's discussions and sharing the results with them can help students develop their discussion skills.)

WHEN TO HOLD DISCUSSIONS

Hold numerous whole-group discussions throughout the unit:

- At the beginning, to identify students' prior knowledge
- Near the beginning, to discuss why they are doing what they are doing
- In the middle, when they have completed a task or are developing a new idea
- At the end, when final conclusions need to be drawn

FACILITATION STRATEGIES

Consider using the following approaches to facilitate discussion among your students:

- Start with a productive question or comment.

- Use *wait time* (time between when a question is asked and when an answer is given, and after an answer is given and the discussion continues).
- Use strategies that allow students to rehearse what they might say: turn and talk, or quick write.
- Redirect student responses so that the talk is not always directed at you.
- Intervene to keep the discussion on topic.

REMINDING STUDENTS OF WORK IN PREVIOUS ACTIVITIES

This unit is intended to be taught on successive days. However, the activities can also be taught with classes that meet only periodically (once or twice a week). There may also be events that lead to lengthy periods between activities (e.g., testing, school breaks, weather-related cancellations). In these cases, you may need to remind students of their prior work. The following strategies may help jog students' memories:

- Quickly recap what students did in their last AWIM activity. Then, toss a small, soft ball to a student and ask a review question. If the student answers the question correctly, they choose whom to throw the ball to next. If the student answers the question incorrectly, they throw the ball back to you, and you pick the next student.
- Draw a tic-tac-toe grid on a piece of chart paper or a whiteboard. Split the class into two teams. Ask the first team a question about the last activity. If the team answers the question correctly, they fill in a space on the tic-tac-toe grid. Then ask the second team a question, and so on.
- Break the content from the last activity into four or five important concepts that you want students to remember (these could be represented as either pictures or words). Make each concept into a jigsaw puzzle, with five or six pieces per puzzle. (Paper plates work great for this!) Jumble the pieces in each puzzle, and give one puzzle to each team of students. When students have assembled their puzzles, they share what their puzzle reminded them of from the previous activity.

If students are not familiar with these discussion strategies, they will need to be taught prior to the start of the unit. The activities in this challenge assume that students have these skills; they do not include time or guidance for teaching them.

Visual Representations

Considered a 21st century skill, *interpreting and conveying ideas visually* is a crucial aspect of students' literacy development. It involves having students create and explain their ideas and experiences through different types of representations, including drawings, symbols, graphs, charts, pictures, and images.

Science Notebooks

ROLE OF SCIENCE NOTEBOOKS

The Science Notebook is the student's record of their work. It is chronological and includes writing, drawings, diagrams, charts, and graphs: whatever is needed to have a full record of their work. The notebook icon at right appears at locations when students should add records to their notebooks. Student Notebook contents will include the following:

- Investigations students undertake
- Results of those investigations
- Questions they ask themselves, other students, or the teacher

- Challenges they encounter and how they addressed them
- Their own ideas, discoveries, and reflections

THE FORM OF THE NOTEBOOK

If you choose to obtain the Programming Each Other Challenge Classroom Materials Kit, Science Notebooks will be included. Otherwise, you will need to provide students with notebooks. Some possible formats for this notebook are as follows:

- Bound notebook with any separate pages glued or stapled in
- Papers held together with brads so that students can add each new page
- Clamp binder
- Three ring-binder (but choose one that doesn't open easily)

NOTEBOOK STRATEGIES

If students have not had to keep Science Notebooks in the past, they will need guidance on how to keep their notebooks. Provide students with the following:

- Models of what a notebook entry should look like
- A checklist of important elements
- Explicit instructions on various notebook strategies, such as observational drawing, graphing, bulleted lists, and diagrams
- Reminders to record regularly
- Regular feedback on their use of notebooks

AUTHENTIC USE

A Science Notebook is a record of work for the student, not a project for the teacher. Students need to realize that it is part of, and necessary for, their work. Below are some instructional strategies that can help facilitate this learning:

- Ask students to use their notebooks to find evidence for the ideas they are discussing.
- Encourage students to look back at previous tasks to help them think about new ones.
- Remind students to use their notebooks to gather evidence when they are getting ready to share a conclusion.
- Facilitate looking at previous notebook entries by asking questions during class.
- Refer to reflections and ideas students have written.

Student Teams

Forming Teams

Before teaching the challenge, plan how to divide the class into teams.

Studies show that girls often stay in the role of notetaker, particularly in science activities. Encourage girls to participate in the hands-on activities, and watch to see that they participate equally in writing and testing the activities. In some cases, same-sex team groupings may be appropriate to encourage equal participation and discussion.

Discuss with students strategies for working together, especially if they are not accustomed to working in teams. Young students often have trouble negotiating the different roles and responsibilities involved in the challenge. Emphasize to students that every student should have the opportunity to engage in the investigation equally.

Managing Student Teams

In addition to the suggestions given earlier, consider the following ideas when planning how to organize and manage student teams:

- Design the teams so that each member brings something different. For example, try to balance energy levels, ability to get along with other students, and reliability in getting work completed.
- Help students develop teamwork skills.
- Assign the role of "manager" to one student in each team for each activity (or have teams self-select a manager, as long as each student gets a turn at the role). The manager will lead the team in discussions, making sure that everyone has an opportunity to share their thoughts and observations. They will also decide which student on the team should share the group's work with the class. Changing this role for each activity supports the quieter students who may have trouble being heard during team discussions.
- Be prepared to rearrange teams as necessary. As you observe teams while they work, remind students that they need to share responsibilities.
- Accent the positive by commending students whenever you see them demonstrating good teamwork skills.
- Build in opportunities for teams to share what they have learned. Students can learn a lot from one another and, ideally, will begin to use one another as resources.
- Visit each team as often as possible, and make notes on your conversations. Having regular conversations about what students are doing, listening and responding to their questions, and talking about their observations can be a rewarding exchange for both you and your students.

Student Assessment

The exploratory nature of the challenges invites the use of a variety of assessment techniques. Here are some assessment opportunities and strategies that you may want to adopt:

- As students are testing their instructions and programs, observe how they carry out their testing. Daily monitoring can reveal how careful students are in reading and writing instructions and how attentive they are in keeping good records.
- Gauge students' understanding through their participation in class discussions and the work of their teams.
- Student work on the reproducible masters found throughout the challenge can help you assess their thinking.
- A rubric for assessing student work on the final project is included in the challenge.
- At the end of the challenge, ask students to write letters to their parents or guardians about what they did, which will give them an opportunity to reflect on their experiences. In addition, parents and guardians will appreciate getting such thoughtful letters from their children.

Implementation Ideas

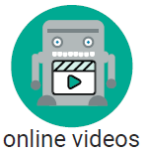
Refer to this section for ideas on materials and classroom management.

Materials

The Programming Each Other Challenge uses very few materials that are not already available in your classroom. The AWIM kit for this unit provides the socks that you and students use to help create their algorithms while learning the basics of programming. Other materials include such things as index cards, markers, and chart paper.

Reproducible Masters and Visual Aids

This unit includes two kinds of blackline masters: reproducible masters and visual aids. Reproducible masters are worksheets. Instructions are included for how many to provide (1 per student, 1 per team, etc.). Students will either fill out these worksheets and hand them in, or insert them into their Science Notebooks so they're available as needed for later activities. Visual aids help describe a specific concept or activity visually. They can be reproduced as posters and hung on the wall, or projected during a lesson. (You can also make individual copies for teams or students.)



Online Videos and Supporting Technology

There are numerous online videos suggested for use throughout this unit. Links to these videos are provided on the AWIM website. Also on the website are links to supporting information and simulations for students. The icons shown at right notify you of when online videos and other supporting technology is required in the unit.

Classroom Management

Most classroom management issues in a challenge like this one typically center on student involvement, grouping issues, and organization. One of the biggest considerations is finding a place where students can safely test their programs. If there is insufficient space in the classroom, corridors outside classrooms, the cafeteria, and the gym can be good testing areas. Always keep safety in mind when students are doing independent work.

Consider the following ideas when planning how to organize and manage the classroom:

- Include students in making rules for working on the challenge and working in teams. List the expectations for students, and keep the list visually accessible at all times.
- Establish clear rules for testing outside the classroom to avoid disturbing other classes.
- Provide ample room for testing—a hallway, cafeteria, or another large room is ideal. Schedule activities during times when the space is not being used by other students.
- Facilitate students' efforts, and help them maintain their focus on the clearly stated expectations.

Activity Calendar

In most cases, the *Programming Each Other Challenge* should run between 10 and 13 (45-minute) sessions. Activity 9 can last from one to two sessions, if you'd like students to have more time to work on their presentations in class. If you limit students' presentation time to 5 minutes, you can likely complete Activity 10 within one session.

Some time to review prior activities is built into the activities, which are intended to be taught on successive days. If you will teach this unit less regularly (e.g., once a week), plan accordingly for extra review time.

Skills Emphasized in This Unit

Problem Solving

Each activity in this unit asks students to write instructions for a variety of tasks, including doing the steps in a dance, guessing a number, and making a sandwich. In each case, the problem they are solving is how best to write these instructions so that another person (or a robot) will follow them to produce the correct results. In the process, students will test their instructions and correct them to achieve their goal.

Computational Thinking

Writing Algorithms

In Activities 2–5, students are tasked with writing clear sets of instructions for increasingly complex problems. Throughout these activities, students instruct a "robot" (the teacher or another student) to do the task of sorting socks. Each successive activity increases the complexity of the task by adding different numbers or colors of socks to the mix. Students will recognize the need to write clear instructions, discover the usefulness of repetition, find patterns in their instructions to identify where loops can save programming steps, and use statements that trigger different tasks depending on specific conditions they can test (such as the color of two socks being the same).

Abstraction

In Activities 2–6, students are presented with several options for problems. For each, they need to make sense of what the task entails, focus on the key aspects of the task, and write instructions that can be followed by someone else. They will abstract the key elements of the problem from the broader descriptions they are given.

Students also engage in abstract thinking in other ways, such as identifying the similarity of their instructions for very different problems (for instance, some key statements in their instructions for folding socks may be nearly identical to key statements in their instructions for making 10 sandwiches).

Students are introduced to the concept of a *variable*, which itself is an abstraction: The instructions tell the robot to operate on the variable, when in practice it is actually operating on whatever the variable holds at that particular time. So, for instance, when the robot is asked to compare the sock in its left hand with the one in its right hand (or "compare sock1 to sock 2"), the instructions might state, "Compare *left-hand-sock* with *right-hand-sock*." However, each time the robot encounters this statement in the course of following the instructions, different socks will actually be in the robot's hand.

Problem Decomposition

Throughout Activities 2–6, students are challenged to break down larger problems into smaller parts. Particularly in Activity 6, students break down a large, complex problem in more than one way.

Added to the sock-sorting activity is the ability to have three robots sorting at the same time (allowing for parallelization). The availability of multiple resources affects the way the problem can be broken down (e.g., each robot can take one-third of the socks, or one robot can sort, one can pair, and one can fold). Students compare the various algorithms and discuss which may be more efficient to solve the problem.

Additional Teacher Resources

Teacher Tips

Throughout the unit, you will see sidebars of information titled Tips from the Field. Some of these tips were offered by teachers who field tested the materials; others provide additional information or are items that we think can help ease your way through the activities.

Background

This section, which appears at the end of each activity, provides additional information regarding the content of the activity. It is intended for the teacher, as it provides more detail about certain information or direct connections to programming and other disciplines. None of the information in this section is necessary to teach to students, but it may help prepare you for discussions or shape your language around the content in the activity.

Glossary

The words in **purple boldface** that appear throughout the unit are defined in the Glossary, which is provided for your convenience. Similar to the Background, the Glossary is there primarily for your own benefit. It is not necessary to ensure that students understand or memorize the terms in the Glossary—it is more important that students understand the general concepts explored in the activities and can explain them *in their own words*. Use your best judgment as to how explicit you wish to make vocabulary a part of students' projects or any assessment.

Teacher Question 2: Did you find this Overview helpful? Is there additional information you wished you had found here?

1 Introducing the Programming Each Other Challenge

Introduction

What Students Do in This Activity

This activity sets the goal for the challenge. Students are asked to learn about programming because an author is writing a book to introduce young children to the idea of programming. The book is about an absent-minded programmer and the funny things that happen when a robot he is programming is given poorly thought-out programs. The challenge for students comes from the publisher, CodeWorks Publishing, who is looking for students to write sample "absent-minded" programs and companion well-thought-out programs to be included at the end of their book.

Rationale

This activity sets the context for the work students will do over the course of the challenge. Although the task comes from a fictional publishing company, it presents an engaging way for students to see how programmers think about their tasks. Students are further motivated by the real-world and hands-on characteristics of this challenge.

Activity at a Glance

Present the Activity

- Review the "Letter from CodeWorks Publishing" (Reproducible Master 1).
- Introduce the challenge: Student teams will create two "programs" for a book about an absent-minded robot programmer—one program that is full of "bugs" and one that works.
- Briefly go over what students need to do for their final projects (Visual Aid 1).
- Show and discuss a video that models "bad" programming instructions.

During the Activity

- Students get into their challenge teams.
- Students view the video a second time, and note any problems they notice on "A Better Set of Instructions" (Reproducible Master 2).

Share and Interpret

- Teams share the issues they found in the video.
- Students compare the challenge in the video to a computer program, and discuss programming.

Reflect and Summarize

- Students share their reactions to the CodeWorks Publishing challenge and note their thoughts in their Science Notebooks.

Time

1 (45-minute) session

Materials

For the teacher

- 1 copy of "CodeWorks Publishing Idea Planner" (Visual Aid 1)

For each team of four

- 1 copy of "A Better Set of Instructions" (Reproducible Master 2)

For each student

- 1 copy of "Letter from CodeWorks Publishing" (Reproducible Master 1)

Preparation

- Photocopy Reproducible Master 2 for each team and Reproducible Master 1 for each student.
- If you haven't already done so, determine how you will divide the class into four-person teams. (For more on forming teams, see *Student Teams* in the Introduction to the Unit.)
- As part of learning about programming, students will explore how writing clear instructions is important. To introduce the ways in which instructions can go wrong, you will show a video based on an internet challenge called the Exact Instructions Challenge. Choose the video that you want to show from the following possibilities (or you can select a similar video of your own):
  - **Exact Instructions Challenge | How to Make a Perfect Sandwich** (https://youtu.be/BAGu3Cp8KK8) (by SuperDuper. In this video (22:55), two young girls in England try to write instructions for their parents to make a ham sandwich. Their mother, who is Spanish speaking, translates the instructions into Spanish. The first daughter's instructions (0:00–3:24) do not result in an edible sandwich. The parents then try the second daughter's instructions (3:25–6:36), which also do not work. After reworking her directions, Daughter 1's second try is again a failure (6:37–11:32). Daughter 2's second try (11:33–15:44) is a success.
  - **Exact Instructions Challenge (FAIL!!!)** (https://youtu.be/edeVxRITK0c) by TodayWithTray. In this video (7:29), a very young girl tells her parents how to make a bowl of cereal with cheese, olives, and milk (0:00–3:37). In the second part, her older brother tells his parents how to make a sandwich. This video is notable because the children do not write the instructions.
  - **Exact Instructions Challenge | Ramen Edition** (https://youtu.be/UNbzNyZcm-Y?t=50s) by Josh Darnit. In this video (8:05), two children write instructions for their father describing how to prepare ramen noodles. In the first segment, the younger child

NOTE

Some of the suggested video segments include instructions for two tasks. You may want to show only portions of those videos for one task, or have students concentrate on one task when they write instructions later in the activity.

creates two sets of instructions that the father cannot follow correctly (0:50–2:40). In the second part, the father follows the older child's instructions (2:41–3:56). The younger child then fails again (3:57–4:38). Finally, the entire group follows instructions sent to them by others.

- **Exact Instructions Challenge!! | The Holderness Family** (https://youtu.be/CT8Z1LlooCM) by The Holderness Family. The two children in this family write instructions for tooth brushing and making toast. The first child's instructions for brushing teeth do not work (0:55–2:08). The second child's instructions for making toast (2:09–3:59) also fail. The second child then gives instructions to her mother for brushing her teeth (3:59–5:09). In the last segment, the mother follows the younger child's instructions for making toast (5:10–5:44).

Classroom Activity

Present the Activity

1. **Pass out a copy of "Letter from CodeWorks Publishing" (Reproducible Master 1) to each student.**
   Read the letter out loud to students (or ask student volunteers to read it to the class). Explain that an author at CodeWorks Publishing is developing a book for young students that explores computer programming through everyday tasks. The main character is an absent-minded programmer who writes programs for a robot that never turn out quite right. CodeWorks would like to include some examples of "absent minded" programs along with their working counterparts at the end of the book. Over the next several weeks, teams of students will each choose a task and write both an absent-minded and a working program for it.

2. **Discuss the letter with the class.**
   Use some or all of the guiding questions below to spark discussion. Feel free to supplement with your own questions, based on what your students seem especially interested in.

   - *What does CodeWorks Publishing want you to do?*

     Students should recognize that CodeWorks Publishing wants them to create two robot "programs" to be included as examples in their book—one that is absent-minded and one that works.

   - *What requirements does CodeWorks Publishing have for the absent-minded program?*

     The letter explains that they would like students to describe how the absent-minded program can go wrong.

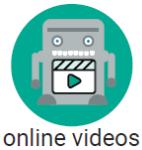
   - *What requirements does CodeWorks Publishing have for the well-thought-out program?*

The letter explains that they would like students to describe how the well-thought-out program fixes the problems found in the absent-minded program and what working programming techniques it uses.

3. **Briefly project "CodeWorks Publishing Idea Planner" (Visual Aid 1), and go over the steps that students will carry out during this unit.**

   Explain that it might seem like a lot right at first, but students will revisit these steps during the unit to remind them of their ultimate goal. As the class progresses through the unit, each planning step will make more sense.

4. **Explain that the author was inspired by the Exact Instructions Challenge on the internet. Show the video you chose.**


online videos

5. **Ask students, *"What happened when the parent tried the instructions prepared by the child?"***

   Students will have noticed that the child's instructions were not very precise, and the parent ended up making a mess of the task they were supposed to do.

6. **Explain that programming may seem difficult at first, but it is really just an organized way to think. Tell students that over the next couple of weeks, they will work in teams to meet the challenge presented by CodeWorks Publishing.**

   Explain that they will explore how to think like a programmer by breaking down everyday tasks into sets of instructions that can easily be followed by someone else.

During the Activity

7. **Break up the class into the teams you pre-selected.**

8. **Pass out copies of "A Better Set of Instructions" (Reproducible Master 2) to each team. Explain that you will show the video again so that students can think about what parts of the child's instructions might have led to problems.**

> ***tips from the field***
>
> *Younger students might find it easier to get started if the first two lines in the Steps column of Reproducible Master 2 are filled in before you copy them for students.*

9. **Have students write instructions on their worksheet—both the instructions they observe and a "better" set of instructions.**

   Ask teams to make sure that each team member writes down at least one step.

> ***tips from the field***
>
> *Stop the video a few times during the second showing to give students time to write.*

Share and Interpret

10. After teams have finished, ask them to share their thoughts about where the child went wrong.

Keep track of students' responses on the board or on a piece of chart paper. Invite them to discuss any differences in what they noted.

11. Ask students how the Exact Instructions Challenge was like a computer program.

12. Discuss what students already know about programming.

Ask questions such as:

- *What does a programmer do?*
- *What kinds of things use programs?*
- *How does a machine, such as a robot, know what to do?*
- *What are* data*?*
- *What do you think the words* input *and* output *mean?*

Give students some time to think, and then ask them to share their ideas with the class.

13. Explain that in the next activity, students will begin to learn more about writing down tasks in a way that is clear enough for a robot to follow.

Reflect and Summarize

14. Ask students to share their reactions to the CodeWorks Publishing challenge.

Remind students that they will work in teams to create two programs for the CodeWorks Publishing book about the absent-minded programmer.

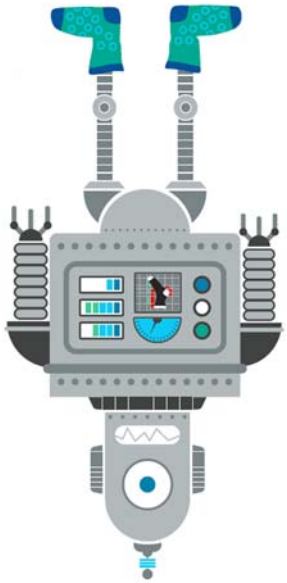
15. Have students include their reproducible masters and note their thoughts in their Science Notebooks.

Reproducible Master 1

Letter from CodeWorks Publishing

Dear Students:

We need your help! CodeWorks Publishing is working with a famous author, Al Gore Rythim, who is writing a book to teach young children about programming. The main character is an absent-minded programmer who keeps writing programs for his robot that *aren't quite right*. The programs look okay to the programmer, but when the robot tries to follow them, things go hilariously wrong! Our CodeWorks team wants to include some examples in this book of "absent-minded" programs (ones with lots of "bugs" in them) and of the same programs, but "cleaned up" so they will work. And that's why we're coming to you! We need your help to write some programs that will be fun for kids to read about.



We suggest that your team takes the following steps to help us:

- Learn how to break down a task into steps.
- Learn how programmers write steps that you have to do over and over.
- Think about how a program would change if you had to do a task a different way (say, instead of taking your usual route to school, you have to

make a detour, or instead of making just one sandwich, you need to make 25 sandwiches for all of your friends).

- With your team, choose an everyday task—such as following a recipe or carrying out a simple chore—that you'd like to write programs for. The task should be interesting and complex enough to provide both an absent-minded program that can go wrong and a real program that can go right.
- Write your absent-minded program and describe how it goes wrong.
- Write your working program and describe how it fixes the problems of the absent-minded one.
- Describe the working programming techniques that you use in your real program.
- Present your examples to "CodeWorks executives."

Good luck! We are very excited to hear your ideas.

Pascal B. Asic
President, CodeWorks Publishing

Reproducible Master 2

A Better Set of Instructions

As you watch the video a second time, work with your team to write down the steps given by the child. Then look at each step: How could you improve it?

Steps	Better Steps
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

Visual Aid 1

CodeWorks Publishing Idea Planner

Throughout the challenge, you will be expected to do the following:

1. Collect all the work you do over the course of the project.
2. In your Science Notebook, keep track of any thoughts or ideas you have about how what you've done relates to your CodeWorks Publishing challenge.
3. Make sure that your programming task is interesting and complex enough to provide both an absent-minded program that can go wrong and a working program that will go right.
4. Complete the tasks below:
   a. Write an absent-minded program and describe how it goes wrong.
   b. Write a working program and describe how it fixes the problems of the absent-minded program.
   c. Describe the real programming techniques that you used in the working program.
   d. Present your two programs to the class.