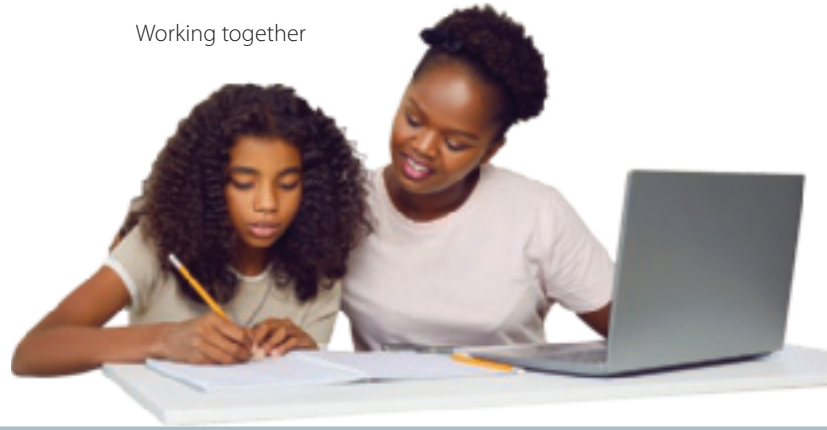


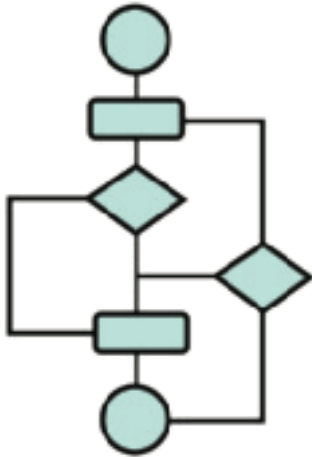
# Designing Computer Programs

Working together



## Teacher Guide

Planning



Data



Programs for learning





# Designing Computer Programs

## Teacher Guide



## Creative Commons Licensing

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.



### You are free:

- to Share**—to copy, distribute, and transmit the work
- to Remix**—to adapt the work

### Under the following conditions:

**Attribution**—You must attribute the work in the following manner:

*This work is based on an original work of the Core Knowledge® Foundation ([www.coreknowledge.org](http://www.coreknowledge.org)) made available through licensing under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. This does not in any way imply that the Core Knowledge Foundation endorses this work.*

**Noncommercial**—You may not use this work for commercial purposes.

**Share Alike**—If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

### With the understanding that:

For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page:

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Copyright © 2023 Core Knowledge Foundation

[www.coreknowledge.org](http://www.coreknowledge.org)

All Rights Reserved.

Core Knowledge®, Core Knowledge Curriculum Series™, Core Knowledge Science™, and CKSci™ are trademarks of the Core Knowledge Foundation.

Trademarks and trade names are shown in this book strictly for illustrative and educational purposes and are the property of their respective owners. References herein should not be regarded as affecting the validity of said trademarks and trade names.

# Designing Computer Programs

## Table of Contents

<b>Introduction</b> .....	1
About this Program: Conceptual Approach and Standards .....	1
Using the Student Reader .....	7
Using the Teacher Guide: Instructional Design and Strategies ...	8
Materials and Equipment .....	14
Pacing .....	16
Lesson 1.1 <b>Computer Systems</b> .....	18
Lesson 1.2 <b>What Is a System?</b> .....	21
Lesson 1.3 <b>How Can You Troubleshoot a Computer Problem?</b> .....	25
Lesson 2.1 <b>Communication Between Computers</b> .....	28
Lesson 2.2 <b>Information Through a Computer System</b> ..	31
Lesson 2.3 <b>Protecting Information on a Network</b> .....	35
Lesson 3.1 <b>Making Data Useful</b> .....	39
Lesson 3.2 <b>Collecting and Storing Data</b> .....	42
Lesson 3.3 <b>Looking at Data</b> .....	45
Lesson 4.1 <b>Collaboration</b> .....	48
Lesson 4.2 <b>Including People’s Ideas</b> .....	51
Lesson 5.1 <b>Designing Programs</b> .....	54
Lesson 5.2 <b>Decision Trees</b> .....	57
Lesson 5.3 <b>Variables</b> .....	60
Lesson 6.1 <b>Computer Code</b> .....	64
Lesson 6.2 <b>Instructions for Actions</b> .....	67
Lesson 6.3 <b>Modification</b> .....	70
Lesson 7.1 <b>Testing and Usability</b> .....	73
Lesson 7.2 <b>Testing and Debugging</b> .....	76
Lesson 7.3 <b>Universal Design</b> .....	79
Lesson 8.1 <b>Respecting Copyrights</b> .....	82

Lesson 8.2	<b>Using Material and Giving Credit</b> .....	85
Lesson 9.1	<b>Using Feedback</b> .....	88
Lesson 9.2	<b>Program Designs and Functions</b> .....	91
Lesson 10.1	<b>Computer Technology and Culture</b> .....	95
Lesson 10.2	<b>Becoming a Computer Programmer</b> .....	98
Teacher Resources	.....	101
	Activity Page Masters .....	102
Appendices	.....	107
	A. Glossary .....	107
	B. Internet Safety .....	110
	C. Strategies for Acquiring Materials .....	112

## **Designing Computer Programs Teacher Guide**

Core Knowledge Computer Science™ 5

# Introduction

## ABOUT THIS PROGRAM

### Core Knowledge Computer Science Conceptual Approach Framework

Grade	Title and Driving Question	Emphasis is on . . .	Overarching Unit Objective
<b>K (ages 5–6)</b>	<b>Computers All Around Us</b> What are computers, and where can we find them?	recognizing	Identify computer devices and parts, and operate them age-appropriately.
<b>1 (ages 6–7)</b>	<b>Helpful Computers</b> What can computer programs do?	using	Experience a variety of age-appropriate programs designed for different purposes.
<b>2 (ages 7–8)</b>	<b>Using Computers</b> How can we use computers and the internet?	using	Use a variety of age-appropriate programs and websites with increasing skill and independence under supervision.
<b>3 (ages 8–9)</b>	<b>Codes and Computers</b> How do programmers build computer programs?	deciphering and evaluating	Model use of symbols, codes, and steps in simple processes and problem-solving.
<b>4 (ages 9–10)</b>	<b>Problem-Solving and Computers</b> What kinds of problems can we solve with computers?	deciphering and evaluating	Model solution design in increasing complexity and involving conditionals and loops.
<b>5 (ages 10–11)</b>	<b>Designing Computer Programs</b> What kinds of computer programs can I develop?	planning and designing	Write, evaluate, and debug code to execute grade-appropriate tasks.

### Skills and Performance Summary

**Grade K**

The goal of the instruction and cumulative experiences in this unit is for young students to build a concept of what makes a thing a computer. Students think about how tools and devices can be helpful for completing tasks. Though students will not specifically study simple and compound machines in science until a later grade level, they are able from everyday exposure to differentiate devices that need a power source and those that do not. Within the class of devices that need a power

source, students can form a further classification of electronic devices—ones that do something with information that is put into the device. Upon recognizing input-processing-output characteristics of devices, students can also begin to classify and use devices for different purposes, including playing games, communicating, and doing schoolwork.

- Grade 1** Building on the recognition established in the previous grade, the goal of the instruction and cumulative experiences in this unit is for young students to begin using computer devices for various purposes, including game play, communication, and schoolwork. Through practice, students understand that the operation of devices has a series of steps in common, in which users input information, the device performs a process, and then a useful output occurs. Inputs include the use of a mouse, keyboard, touchscreen, game controller, camera, and microphone. Output includes screen visuals (text, still images, video), hard-copy printout, and audio. Accessing desired outputs from computers depends first on successful operation of input devices.
- Grade 2** Building on the use of devices established in the previous grade, the goal of the instruction and cumulative experiences in this unit is for young students to continue using with increasing independence computer devices for various purposes, including game play, communication, and schoolwork. With increasing practice in successfully inputting information, students can consider ways to achieve creative outputs that use and combine text, graphics, visual data displays, photos, animations, video, and audio.
- Grade 3** In this grade level, the emphasis begins to shift from use—skillfully inputting information to prompt outputs—to the unseen processes inside computing technology. The goal of the instruction and cumulative experiences in this unit is for students to identify the output of every computer device as a solution to a problem and deconstruct the solution into the necessary steps required. Students discern that computers do not think on their own to solve problems but only execute step-by-step instructions that are built into them. And students begin to dissect messages into the granular parts that make them up and transmit them.
- Grade 4** Building on the concepts of symbols, codes, signals, and step-by-step instructions established in the previous grade, the goal of the instruction and cumulative experiences in this unit is for students to continue articulating steps of increasing complexity to solve problems. Beyond dissecting solutions into granular steps, students think forward in scenarios with conditional branching options (if-then statements) and loops (if-then, repeat prior instruction). Students relate what computing devices do to their programming, and they build understanding that complex computer programming is the outcome of building, layering, and chaining together links of fundamentally simple code.
- Grade 5** In this grade level, the emphasis begins to shift from examination—figuring out how instructions make computing devices do what they do—to creative application of that knowledge. They step forward from the role of computer technology user into a beginning maker role. Students begin to use functional computer language to craft code for executable tasks.



## Note to Teachers and Curriculum Planners

This unit introduces Grade 5 students to increasingly complex concepts related to computer programming. The unit includes a story-based Student Reader and this Teacher Guide. The lesson sessions alternate between those focused on the Student Reader and others that are interactive in nature and do not involve a Student Reader chapter.

This unit can be integrated into your existing science or math curriculum. The lessons can be inserted intermittently, or the entire unit can be taught on consecutive days. The unit can also be offered as an enrichment course.

## Standards: What are the relevant CSTA Concepts for this unit?\*

This unit, *Designing Computer Programs*, has been informed by the K–12 Computer Science Standards put forth by the Computer Science Teachers Association. The CSTA K–12 Computer Science Standards delineate a core set of learning objectives designed to provide the foundation for computer science curricula and implementation at the K–12 level. The CSTA Standards introduce fundamental concepts of computer science beginning at the elementary school level.

Online Resources



For detailed information about the CSTA references, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Grades 3–5 CSTA Standards

Concepts	Subconcepts	Level 1B (ages 8–11) By the end of Grade 5, students will be able to . . .
Computing Systems	Devices	1B-CS-01 Describe how internal and external parts of computing devices function to form a system. (P7.2)
	Hardware and Software	1B-CS-02 Model how computer hardware and software work together as a system to accomplish tasks. (P4.4)
	Troubleshooting	1B-CS-03 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. (P6.2)

\* The K–12 Computer Science Framework, led by the Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and National Math and Science Initiative in partnership with states and districts, informed the development of this work. Authors of the CSTA K–12 Computer Science Standards were not involved in the production of this product, and their endorsement is not implied.

#### Sources:

Computer Science Teachers Association (2017). CSTA K-12 Computer Science Standards, Revised 2017. Retrieved from <http://www.csteachers.org/standards>.

K–12 Computer Science Framework. (2016). *Framework view by grade band*. Retrieved from <http://www.k12cs.org>.

Concepts	Subconcepts	<b>Level 1B (ages 8–11)</b> By the end of Grade 5, students will be able to . . .
<b>Networks and the Internet</b>	<b>Network, Communication, and Organization</b>	1B-NI-04 Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination. (P4.4)
	<b>Cybersecurity</b>	1B-NI-05 Discuss real-world cybersecurity problems and how personal information can be protected. (P3.1)
<b>Data and Analysis</b>	<b>Storage</b>	(continuation) 1A-DA-05 Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data. (P4.2)
	<b>Collection, Visualization, and Transformation</b>	1B-DA-06 Organize and present collected data visually to highlight relationships and support a claim. (P7.1)
	<b>Inference and Models</b>	1B-DA-07 Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea. (P7.1)
<b>Algorithms and Programming</b>	<b>Algorithms</b>	1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. (P6.3, P3.3)
	<b>Variables</b>	1B-AP-09 Create programs that use variables to store and modify data. (P5.2)
	<b>Control</b>	1B-AP-10 Create programs that include sequences, events, loops, and conditionals. (P5.2)
	<b>Modularity</b>	1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program 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)
	<b>Program Development</b>	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)
		1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. (P7.3)
		1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. (P6.1, P6.2)
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)		
1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations. (P7.2)		

Concepts	Subconcepts	Level 1B (ages 8–11) By the end of Grade 5, students will be able to . . .
<b>Impacts of Computing</b>	<b>Culture</b>	1B-IC-18 Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices. (P7.1)
		1B-IC-19 Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users. (P1.2)
	<b>Social Interactions</b>	1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts. (P1.1)
	<b>Safety, Ethics, and Law</b>	1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission. (P7.3)

## Integrated Practices

1. Fostering an Inclusive Computing Culture
  - 1.1 Include the unique perspectives of others . . .
  - 1.2 Address the needs of diverse end users . . .
  - 1.3 Employ self and peer-review advocacy . . .
2. Collaborating Around Computing
  - 2.1 Cultivate working relationships . . .
  - 2.2 Create team norms, expectations, and equitable workloads . . .
  - 2.3 Solicit and incorporate feedback . . .
  - 2.4 Evaluate and select technological tools . . .
3. Recognizing and Defining Computational Problems
  - 3.1 Identify complex, interdisciplinary, real-world problems . . .
  - 3.2 Decompose complex real-world problems . . .
  - 3.3 Evaluate whether it is appropriate and feasible . . .
4. Developing and Using Abstractions
  - 4.1 Extract common features . . .
  - 4.2 Evaluate existing technological functionalities . . .
  - 4.3 Create modules and develop points of interaction . . .
  - 4.4 Model phenomena and processes and simulate systems . . .

5. Creating Computational Artifacts
  - 5.1 Plan the development of a computational artifact . . .
  - 5.2 Create a computational artifact . . .
  - 5.3 Modify an existing artifact . . .
6. Testing and Refining Computational Artifacts
  - 6.1 Systematically test . . .
  - 6.2 Identify and fix errors . . .
  - 6.3 Evaluate and refine . . .
7. Communicating About Computing
  - 7.1 Select, organize, and interpret . . . data sets . . .
  - 7.2 Describe, justify, and document computational processes . . .
  - 7.3 Articulate ideas responsibly . . .

(See *K12 Computer Science Framework* for full text of the Practices integrated into the standards.)

## What Teachers Need to Know

Supportive information on the content standards and the concepts they address is provided throughout the lessons at points of relevance:

**Know the Standards:** These sections, found later in this Teacher Guide, explain what to teach and why, with reference to CSTA and Core Knowledge expectations.

**Know the Science:** These sections provide supporting, adult-level, background information or explanations related to specific computer science concepts or examples.

## A Special Note to Grade 5 Teachers Before Starting This Unit

### Why Study Computer Science in the Elementary Grades?

For many elementary teachers, the suggestion that computer science should be a part of the Grade 5 curriculum may seem questionable. For many teachers, precious instructional time is devoted to teaching core content area fundamentals to students.

The study of computer science in Grade 5, however, is consistent with the Core Knowledge approach to learning. Knowledge builds upon knowledge, and computer science builds upon the foundational understanding that many devices share basic underlying technology. And that technology operates through shared processes that are simple at their core but are combined in increasingly complex ways that can produce unlimited outcomes. Grade 5 students can build on what

they learned in Kindergarten through Grade 4, to recognize that devices are operated by instructional coding. They can also consider step-by-step instructions, if-then scenarios, and simple codes that carry meaning. These building blocks have established readiness for students to put concepts together and make their own instructional codes for simple computing operations.

## USING THE STUDENT READER

### Student Reader



The *Designing Computer Programs* Student Reader includes ten chapters and a student Glossary providing definitions to Core Vocabulary words. Engaging text and images encourage students to draw upon their own experiences and the world around them to understand unit concepts. In addition to Core Vocabulary, the Student Readers include a feature called Word to Know, which provides background information to help students understand key terms and may sometimes include additional informational boxes, such as Think About.

**Read, then explore:** In the CKSci computer program, lessons are sequenced for reading before active engagement. First, students are introduced to concepts through relatable scenarios in the Student Readers. Then students explore the concepts through hands-on investigations or teacher demonstrations, accompanied by active questioning and analysis. The icon shown at left will signal Core Lesson segments that focus on Student Reader chapters.

**Independent reading or group read aloud:** While the text in the Student Readers is written for independent reading, we encourage group read aloud and engagement with the text. The Teacher Guide provides Guided Reading Supports to prompt discussion, clarify misconceptions, and promote understanding in relation to the Driving Question and chapter topics.

### Online Resources



Use this link to download the CKSci Online Resources for this unit, where the specific link to this appendix can be found:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

## Instructional Design

The *Designing Computer Programs* unit is the Grade 5 installment in the CKSci elementary computer science series. To meet CSTA Standards we encourage teachers to complete the corresponding grade-level unit each school year. Each lesson part requires thirty to forty-five minutes of instruction time. The time it takes to complete a full lesson depends on class size and individual circumstances.

Within the Teacher Guide, each Core Lesson is composed of multiple numbered parts, generally four to six. Each segment concludes with a Check for Understanding, providing the teacher with a quick opportunity for formative assessment. At the end of this unit introduction, you will find a blank pacing guide, which you may use to plan how you might pace the lessons. We strongly recommend that you preview the unit in full before beginning and create your pacing guide before teaching the first lesson.

### Features

The unit is composed with several integrated features that support learning for all students and development for teachers.

This Computer Science unit can be completed BOTH in classrooms that have computers and devices for students and those that do not. Look for the icons at left to indicate the types of student interactions recommended within each lesson. Pay attention to those icons most relevant to your classroom situation.

#### Unplugged



There are many opportunities for students to complete foundational exercises relevant to the concepts of computer science without needing access to computers. These activities are designated by the Unplugged icon.

#### Device-Based



Several activities and extensions do require that students work on computers, and those experiences are designated by the Device-Based icon.

#### Online Resources



A few activities and demonstrations recommended in this unit require the use of internet-connected devices. These experiences are indicated by the Online Resources icon and will also include references to the Online Resources Guide for links to the recommended internet resources.

#### Differentiation

Adjustments to instruction appear in the text, indicated by **SUPPORT**, **EXTEND**, and **CHALLENGE** notations.

**SUPPORT**—Reading, writing, listening, and/or speaking alternatives appear for students who are English language learners, have special needs, or read below the grade level. Extra support is suggested for students who struggle to meet targeted expectations.

**EXTEND**—Extensions are suggested for students with high interest or who have already met the performance expectations.

**CHALLENGE**—Additional, relevant, and interesting exercises are suggested for students to explore that exercise math, reading, or science skills/comprehension that pushes beyond the grade level.

<b>Teacher Development</b>	Information in the instructional text, Know the Science boxes, and Know the Standards boxes are provided to support ongoing teacher development with regard to both content and the teaching process.
<b>Monitor Progress</b>	Opportunities for formative assessment appear throughout the instructional support. These instances are most consistently noted in a Check for Understanding that concludes each lesson segment.

## The Core Lessons

- Lesson time: Each Core Lesson part constitutes one classroom session of thirty to forty-five minutes. Some activities and performance tasks will require setting aside a longer block of time.
- Lesson order: The lesson parts are coherently sequenced to build from one to the next, linking student engagement across lessons and helping students build new learning on prior knowledge.

<b>Driving Question:</b> What kinds of computer programs can I develop?		
<b>Overarching Unit Objective:</b> Write, evaluate, and debug code to execute grade-appropriate tasks.		
Lesson and Part	Title	Interaction Type and Learning Objective
<b>Lesson 1, Part 1</b>	<b>Computer Systems</b>	Student Reader Read Aloud: Chapter 1, “Computer Systems” <ul style="list-style-type: none"> <li>• Describe contributions of component parts to the function of a system.</li> </ul>
<b>Lesson 1, Part 2</b>	<b>What Is a System?</b>	Classroom Interactive <ul style="list-style-type: none"> <li>• Articulate the components of a familiar system, and demonstrate how they interact to carry out functions that the individual parts cannot.</li> </ul>
<b>Lesson 1, Part 3</b>	<b>How Can You Troubleshoot a Computer Problem?</b>	Classroom Interactive <ul style="list-style-type: none"> <li>• Demonstrate how components in a system are critical to the overall function of the system.</li> </ul>
<b>Lesson 2, Part 1</b>	<b>Communication Between Computers</b>	Student Reader Read Aloud: Chapter 2, “Communication Between Computers” <ul style="list-style-type: none"> <li>• Conceptualize the transmission of digital messages.</li> </ul>
<b>Lesson 2, Part 2</b>	<b>Information Through a Computer System</b>	Classroom Interactive <ul style="list-style-type: none"> <li>• Classify and organize information based on consistent criteria.</li> </ul>
<b>Lesson 2, Part 3</b>	<b>Protecting Information on a Network</b>	Classroom Interactive <ul style="list-style-type: none"> <li>• Detail effective password usage.</li> </ul>

<b>Lesson 3, Part 1</b>	<b>Making Data Useful</b>	Student Reader Read Aloud: Chapter 3, "Making Data Useful" <ul style="list-style-type: none"> <li>Describe management of data to support revealing patterns, answering questions, and making predictions.</li> </ul>
<b>Lesson 3, Part 2</b>	<b>Collecting and Storing Data</b>	Classroom Interactive <ul style="list-style-type: none"> <li>Identify best methods and use tools for collecting and recording data.</li> </ul>
<b>Lesson 3, Part 3</b>	<b>Looking at Data</b>	Classroom Interactive <ul style="list-style-type: none"> <li>Organize data sets to reveal patterns and highlight relationships.</li> <li>Use data patterns to support claims.</li> </ul>
<b>Lesson 4, Part 1</b>	<b>Collaboration</b>	Student Reader Read Aloud: Chapter 4, "Collaboration" <ul style="list-style-type: none"> <li>Describe the elements of a successful team.</li> </ul>
<b>Lesson 4, Part 2</b>	<b>Including People's Ideas</b>	Classroom Interactive <ul style="list-style-type: none"> <li>Use teamwork, respectful communication, and varying roles to achieve a goal.</li> </ul>
<b>Lesson 5, Part 1</b>	<b>Designing Programs</b>	Student Reader Read Aloud: Chapter 5, "Designing Programs" <ul style="list-style-type: none"> <li>Explain why breaking down a task efficiently and accurately is important for completing a task.</li> </ul>
<b>Lesson 5, Part 2</b>	<b>Decision Trees</b>	Classroom Interactive <ul style="list-style-type: none"> <li>Create a decision tree.</li> <li>Compare the efficacy of two different decision trees.</li> </ul>
<b>Lesson 5, Part 3</b>	<b>Variables</b>	Classroom Interactive <ul style="list-style-type: none"> <li>Recognize the prevalence of variable factors and the sources of variability.</li> </ul>
<b>Lesson 6, Part 1</b>	<b>Computer Code</b>	Student Reader Read Aloud: Chapter 6, "Computer Code" <ul style="list-style-type: none"> <li>Identify computer commands to instruct a computer to do simple tasks.</li> </ul>
<b>Lesson 6, Part 2</b>	<b>Instructions for Actions</b>	Classroom Interactive <ul style="list-style-type: none"> <li>Execute coding fundamentals to complete a task using events, sequences, conditions, and loops.</li> </ul>
<b>Lesson 6, Part 3</b>	<b>Modification</b>	Classroom Interactive <ul style="list-style-type: none"> <li>Decompose a computer task into smaller, isolated tasks (which may themselves be decomposed).</li> <li>Incorporate smaller portions of existing programs to develop something new.</li> </ul>
<b>Lesson 7, Part 1</b>	<b>Testing and Usability</b>	Student Reader Read Aloud: Chapter 7, "Testing and Usability" <ul style="list-style-type: none"> <li>Describe how testing and usability research help programs meet the needs of the most users.</li> </ul>



<b>Lesson 7, Part 2</b>	<b>Testing and Debugging</b>	Classroom Interactive <ul style="list-style-type: none"> <li>• Test and gather data about a solution.</li> <li>• Determine if solutions meet intended criteria and constraints.</li> </ul>
<b>Lesson 7, Part 3</b>	<b>Universal Design</b>	Classroom Interactive <ul style="list-style-type: none"> <li>• Articulate diverse perspectives related to improving computer technology tools.</li> </ul>
<b>Lesson 8, Part 1</b>	<b>Respecting Copyrights</b>	Student Reader Read Aloud: Chapter 8, “Respecting Copyrights” <ul style="list-style-type: none"> <li>• Explain the concept of copyright.</li> <li>• Summarize the responsibilities of media use.</li> </ul>
<b>Lesson 8, Part 2</b>	<b>Using Material and Giving Credit</b>	Classroom Interactive <ul style="list-style-type: none"> <li>• Model the responsible and ethical use of media from outside sources.</li> </ul>
<b>Lesson 9, Part 1</b>	<b>Using Feedback</b>	Student Reader Read Aloud: Chapter 9, “Using Feedback” <ul style="list-style-type: none"> <li>• Recognize communicating with peers about proposed solutions as an important part of the design process.</li> <li>• Value that shared ideas can lead to improved designs.</li> </ul>
<b>Lesson 9, Part 2</b>	<b>Program Designs and Functions</b>	Classroom Interactive <ul style="list-style-type: none"> <li>• Design, test, and revise a computer project.</li> <li>• Present or demonstrate a design, and reflect on choices made to achieve desired goals and what was learned from the experience.</li> <li>• Critique the projects of other students.</li> </ul>
<b>Lesson 10, Part 1</b>	<b>Computer Technology and Culture</b>	Student Reader Read Aloud: Chapter 10, “Computer Technology and Culture” <ul style="list-style-type: none"> <li>• Describe how technology has evolved to improve lives.</li> </ul>
<b>Lesson 10, Part 2</b>	<b>Becoming a Computer Programmer</b>	Classroom Interactive <ul style="list-style-type: none"> <li>• Explore online coding and programming options for students.</li> </ul>

## Activity Pages

### Activity Pages



AP 2.2  
AP 6.2  
AP 8.2

Blackline reproducible masters for Activity Pages are included in Teacher Resources at the back of the Teacher Guide. The icon shown to the left appears throughout the Teacher Guide wherever Activity Pages (AP) are referenced.

Make sufficient copies for your students in advance of each lesson.

Lesson 2, Part 2—Mystery Message (AP 2.2)

Lesson 6, Part 2—Instructions for Actions (AP 6.2)

Lesson 8, Part 2—Copyright Dilemma (AP 8.2)

## Online Resources for Science

---

### Online Resources



For each CKSci Computer Science unit, the Teacher Guide includes references to online resources (including external websites and downloadable documents) that enhance classroom instruction. Look for the icon on the left.

Use this link to download the CKSci Online Resources for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

## Teaching Strategies

---

### Start with the familiar.

Lead with an experience. Begin each lesson with a demonstration, activity, or question about a phenomenon to engage students and focus their attention on the topic. Start with the familiar. Every science topic introduced to students relates in some way to their known world and everyday experiences. The purpose of every lesson is to build a bridge between what is familiar to students and broader knowledge about the way the world works.

### Ask driving questions.

Each multipart lesson is built around a lesson Driving Question. And at the beginning of each Teacher Guide lesson segment, you will find a reminder of the Overarching Unit Objective. Each lesson activity provides an incremental concept to help students move toward the objective and ultimately be able to offer a confident answer to the Driving Question. Pose discussion questions in each class session with the Driving Question and Overarching Unit Objective in mind.

### Encourage scientific thinking.

Approach the lessons with students not as learning about computer science, but as learning about the world with a scientific mind.

Throughout the lessons, encourage students to ask questions about what they observe, do, and read. Record relevant questions in a prominent place in the classroom. Guide students back to these questions as opportunities to answer them emerge from readings, demonstrations, and activities.

### Use continuous Core Vocabulary instruction.

During instruction, emphasize Core Vocabulary terms and their meanings in context rather than relying on isolated drill for memorization of definitions. Through scaffolded questioning, encourage students to come up with definitions in their own words and to use the words in their own sentences.

Core Vocabulary words for each lesson, as well as Language of Instruction, other key terms teachers are encouraged to use in discussing topics with students, are provided at the start of each lesson. You can find Core Vocabulary and selected Language of Instruction definitions in the Glossary at the back of this Teacher Guide.

---

**Emphasize observation and experience.**

Lessons employ various ways for students to learn, including watching, listening, reading, doing, discussing, and writing.

---

**Make frequent connections.**

Use a combination of demonstrations and reading materials, rich with examples, to help students recognize how the concepts they are learning apply in their everyday lives. Prompt students to relate lesson content to their own experiences, to relate the new and unfamiliar to the familiar, and to connect ideas and examples across disciplines.

---

**Monitor student progress.**

Use verbal questioning, student work, and the Check for Understanding prompts at the end of each lesson part to monitor progress during each lesson and to measure understanding at the conclusion of the unit. Many lessons provide tips to help you support students who need further explanations or clarifications.

---

## Effective and Safe Classroom and Online Activities

---

Conducting safe classroom demonstrations and activities and safe online interactions is essential to successful elementary education. The following resources, included at the back of the Teacher Guide, provide Core Knowledge's recommendations for developing effective science and computer science classroom activities:

- Internet Safety
- Student Online Safety Contract
- Strategies for Acquiring Materials

Online Resources



These resources may also be accessed within the CKSci Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

## MATERIALS AND EQUIPMENT

The unit requires a variety of materials to support various ways of learning (including doing, discussing, listening, watching, reading, and writing). Prepare in advance by collecting the materials and equipment needed for all of the demonstrations and hands-on investigations.

- A large whiteboard, blank sheets of printer paper, student notebooks, pencils, and scissors are routinely required but not listed below.
- A classroom computer, tablet, or other computing device, as needed for discussion, is referred to in the materials for lesson segments in which it is used but is not repeated in the materials listed here. Internet access and the means to project images/videos for whole-class viewing are also required in many lesson segments but not repeated below.

### Lesson 1, Part 3

- variety of classroom craft materials such as chenille sticks, craft sticks, rubber bands, pencils, etc.
- variety of classroom building/game sets such as dominoes, blocks, magnet tiles, marble tracks, connecting blocks, etc.
- marbles or table tennis balls for student groups of four or five

### Lesson 2, Part 2

- 3 paper clips or rubber bands

### Lesson 3, Part 2

- variety of measurement tools as needed for student data collection

### Lesson 3, Part 3

- markers, colored pencils, crayons, etc.
- data sheet prepared in Lesson 3, Part 2

### Lesson 5, Part 2

- colored whiteboard markers
- pattern blocks (one set of the six shapes per pair of students)

### Lesson 5, Part 3

- computer access with a word-processing application
- printer access, preferably color
- blue, yellow, pink, and green highlighters

### Lesson 6, Part 2

- colored markers

### Lesson 6, Part 3

- mind map from Lesson 6, Part 2
- highlighter

### Lesson 7, Part 3

- internet access or resources about Universal Design concepts
- variety of materials to design a Universal Design prototype (optional extension)

## Lesson 9, Part 2

- computers and internet access to the recommended programming app (ScratchJr)
- printable motion blocks (from ScratchJr Fun) (4 to 5 sets, 1 set per group)  
(See the Online Resources Guide for links to these items. [www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources))
- small manipulative or play figurine (1 per group)
- pencils with erasers

## Lesson 10, Part 2

- index cards (minimum of 2 per student)
- library access for resources
- internet access (optional)
- document projector (optional)

# DESIGNING COMPUTER PROGRAMS PACING

\_\_\_\_\_’s Class

**Note to Teacher:** *Designing Computer Programs* can be taught on consecutive days as a supplemental unit to your science or math curriculum. It can also be offered as an enrichment program. If the program is implemented in consecutive thirty- to forty-five-minute class sessions, it will take twenty-eight days to complete.

## Week 1

Day 1

Day 2

Day 3

Day 4

Day 5

--	--	--	--	--

## Week 2

Day 6

Day 7

Day 8

Day 9

Day 10

--	--	--	--	--

## Week 3

Day 11

Day 12

Day 13

Day 14

Day 15

--	--	--	--	--

## Week 4

Day 16

Day 17

Day 18

Day 19

Day 20

--	--	--	--	--

**Week 5**

**Day 21**

**Day 22**

**Day 23**

**Day 24**

**Day 25**

--	--	--	--	--

**Week 6**

**Day 26**

**Day 27**

**Day 28**

**Day 29**

**Day 30**

--	--	--	--	--

# Computer Systems

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Describe contributions of component parts to the function of a system.

### Instructional Activities

- student reading
- whole-class discussion

### CSTA Standards

**Concept:** Computing Systems

#### Subconcepts and Standards:

- **Devices: 1B-CS-01** Describe how internal and external parts of computing devices function to form a system. (P7.2)
- **Hardware and Software: 1B-CS-02** Model how computer hardware and software work together as a system to accomplish tasks. (P4.4)
- **Troubleshooting: 1B-CS-03** Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. (P6.2)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**component**

**system**



**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

### troubleshoot

## Instructional Resources

Student Reader



Ch. 1

**Student Reader, Chapter 1**  
Computer Systems

## Materials and Equipment

**Collect or prepare the following:**

- classroom computer, tablet, or other computing device as needed for discussion

## THE CORE LESSON

### 1. Focus attention on today's topic.

How does a computer system work? Ask students to describe the individual **components** within a **system** and how they work together to achieve a task. Prompt them with a simple issue that would break the system, such as a connection being unplugged. Discuss how one component not functioning properly can disrupt the whole system but that by considering the role of the individual parts and how they interact step-by-step, the problem can typically be found and resolved, which is called troubleshooting.

### 2. Read and discuss "Computer Systems."

Student Reader



Ch. 1

Read together or have students read independently "Computer Systems," in the Student Reader. The selection introduces the idea that a computer is comprised of many different components that are necessary for the computer to function properly.

### Guided Reading Supports

When reading aloud together as a class, always prompt students to follow along. Pause for discussion. Include suggested questions and prompts:

#### Page 2

Prompt students to share their own experiences with being an older or younger "buddy" to someone—at school, at home, or in an activity or sport. Ask them what was special about being the younger buddy. What did they enjoy about being the older buddy?

#### Page 3

Ask students to describe one component of a computer device they have in the classroom or school and why it is important to the overall function of the computer. Is it hardware or software? Is it input or output?

**Page 4**

Ask students to think about their own troubleshooting with any type of device. When they encountered a problem, did they give up and stop using it? Did they try to resolve the issue or ask for help?

**Page 5**

Prompt students to discuss their experiences with using a computing device to help them learn to read. What were the hardware and software they used? How did it help them with their reading?

### 3. Check for understanding.

---

Unplugged  
Activity



On any classroom computing device, point or refer to various components, and ask students to name them and describe their function within the system. Ask students what would happen to the system if that component were not working and what steps they would take to troubleshoot and resolve the issue.

## Know the Science

Computing devices depend on other devices or components to complete tasks. A system is a set of things that work together as parts of a mechanism or network. Hardware and software are both needed for a functional computer. Input and output devices are also essential components to most computing systems. For example, the keyboard (hardware) detects a key press (input), which the operating system and word-processing application (software) display as a new character that has been inserted into the document and is visible through the screen of the monitor (hardware). The printer (hardware) can transfer the image from the screen to a paper printout (output). Although computing systems vary, understanding the basic components and terminology of a computing system can help students use troubleshooting strategies to identify and develop solutions to common computer problems.

# What Is a System?

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Articulate the components of a familiar system, and demonstrate how they interact to carry out functions that the individual parts cannot.

### Instructional Activities

- independent activity
- whole-class activity
- whole-class discussion

### CSTA Standards

**Concept:** Computing Systems

#### Subconcepts and Standards:

- **Devices: 1B-CS-01** Describe how internal and external parts of computing devices function to form a system. (P7.2)
- **Hardware and Software: 1B-CS-02** Model how computer hardware and software work together as a system to accomplish tasks. (P4.4)
- **Troubleshooting: 1B-CS-03** Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. (P6.2)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

## Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**input/output**

**system**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**cause and effect**

**interaction**

**solution**

## Materials and Equipment

---

### Collect or prepare the following:

- blank sheets of paper or student notebooks
- classroom projector

## Advance Preparation

---

- Create a simple table with two columns: “Student” and “System.”
- List all students’ names in the “Student” column.
- Prepare the classroom projector for class discussion.

## THE CORE LESSON

### 1. Focus attention on today’s topic.

---

Ask students to remember what makes a computer system work—individual components working together to achieve a task. Ask students if computer systems are the only type of system. Prompt them to name other types of systems they are familiar with, such as a body system (cardiovascular, digestive, nervous) or a home system (plumbing, electrical, heating).

### 2. Facilitate the activity.

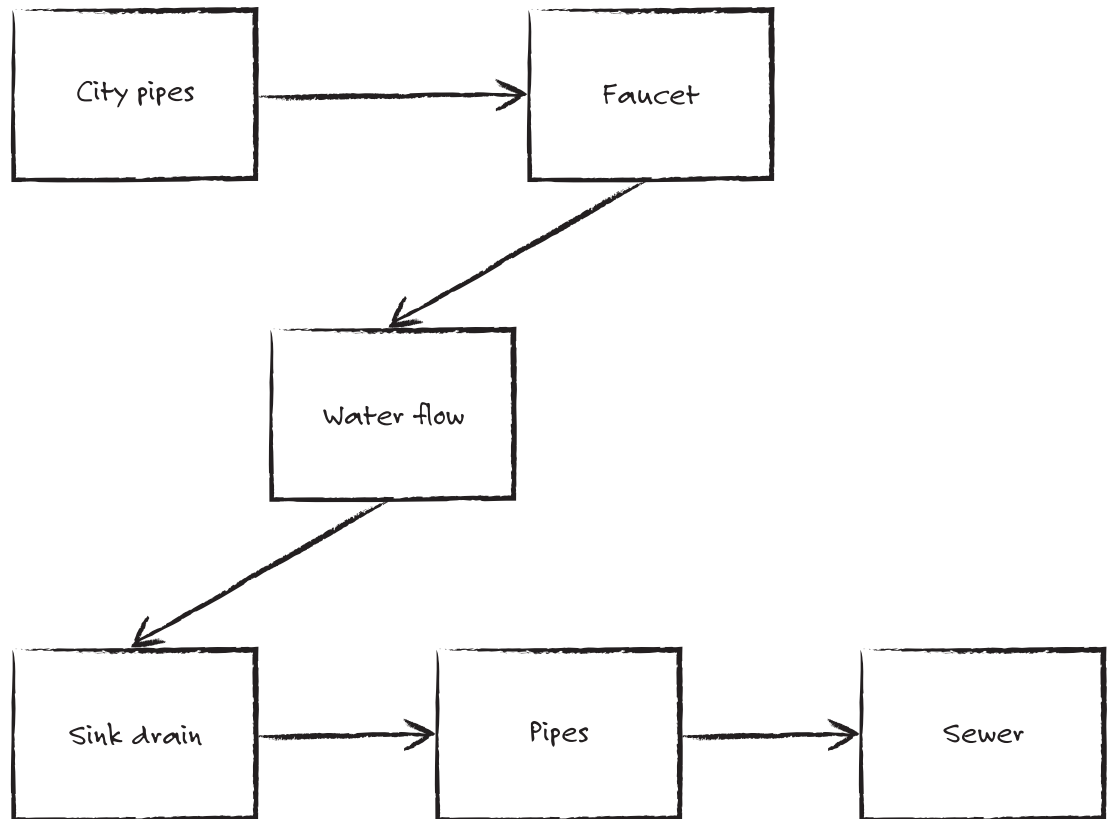
---

Device-Based  
Activity



Ask students to quietly think of a system they are familiar with other than a computer system. They should be able to name the basic components or functions within the system they have selected.

On a blank sheet of paper, tell students to draw five empty boxes, each about two inches by two inches. Students then use the boxes to write at least five components or steps in the system they are describing. They do not name their system anywhere on the paper. Students then use arrows to show the interactions within that system. More components and interactions can be used as needed but should not exceed eight boxes or steps. For example:



Students leave their system charts faceup on their desks. Students use a notebook or blank sheet of paper to create a simple table for “Student” and “System.” Students travel around the room, writing in each classmate’s name and the system they believe they described. For example:

Student	System
Henry	Electricity
Mabel	Digestion
Carina	Cardiovascular
Juan	Solar power

After all desks are visited and a complete class chart is made, students return to their seats. Using a projector, show your chart of student names with the empty “System” column. Call on each student to stand up and read their system components on their diagram. Prompt the class to raise their hands and share their guess of what system is depicted. Ask others if they had any different guesses. Each student confirms the system they have described as you enter it into the projected classroom chart.

### 3. Check for understanding.

---

Unplugged  
Activity



After all students have shared their system chart and all guesses are made, ask students what all of these systems have in common. Prompt them to explain that just like a computer system, each has individual components that interact in order to complete a task.

Using the plumbing example, ask students to share their thoughts on what would happen if the faucet were removed from the system. Use other student diagrams to ask what would happen if a component were removed.

**EXTEND**—Extend student understanding of systems to computing devices by creating a similar flowchart on the whiteboard for a basic computer system, including hardware elements such as a processor, keyboard, mouse, monitor, printer, etc. and software such as an operating system, word processing app, internet browser, etc.

## Know the Standards

*1B-CS-01 Describe how internal and external parts of computing devices function to form a system.*

Computing devices often depend on other devices or components. For example, a robot depends on a physically attached light sensor to detect changes in brightness, whereas the light sensor depends on the robot for power. Keyboard input or a mouse click could cause an action to happen or information to be displayed on a screen; this could only happen because the computer has a processor to evaluate what is happening externally and produce corresponding responses.

*1B-CS-02 Model how computer hardware and software work together as a system to accomplish tasks. In order for a person to accomplish tasks with a computer, both hardware and software are needed.*

# How Can You Troubleshoot a Computer Program?

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Demonstrate how components in a system are critical to the overall function of the system.

### Instructional Activities

- small-group activity
- whole-class demonstration
- whole-class discussion

### CSTA Standards

**Concept:** Computing Systems

#### Subconcepts and Standards:

- **Hardware and Software: 1B-CS-02** Model how computer hardware and software work together as a system to accomplish tasks. (P4.4)
- **Troubleshooting: 1B-CS-03** Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. (P6.2)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

## Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**system**

**troubleshoot**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**cause and effect**

**interaction**

**solution**

## Materials and Equipment

### Collect or prepare the following:

- variety of classroom craft materials such as chenille sticks, craft sticks, rubber bands, pencils, etc.
- variety of classroom building/game sets such as dominoes, blocks, magnet tiles, marble tracks, connecting blocks, etc.
- marbles or table tennis balls for student groups of four or five.

## Advance Preparation

- Ask for donations of recycled materials such as toilet paper, paper towel, and gift wrap tubes; egg cartons; small containers; craft sticks; and twist ties.

## THE CORE LESSON

### 1. Focus attention on today's topic.

Remind students of the plumbing diagram from Lesson 1.2 and their responses to the faucet being removed from the system. Discuss how one component in a system not being there or functioning properly can disrupt the whole system.

Now, prompt students with a plumbing problem—water is filling up in the sink! Ask them to share their thoughts on which component is likely not working correctly. Encourage them to identify that the sink drain is clogged, covered, or missing and what could be done to make the system work properly again.

### 2. Facilitate the activity.

Unplugged  
Activity



Place students in teams of four or five. Tell them they need to work together to create a chain reaction system that has at least three interactions that work together to get a marble or table tennis ball to land on a target. They will only have twenty minutes to create their system.

Using classroom building materials or recycled materials, teams work together to design a simple but functional chain reaction path. Some suggestions are as follows.

1. Dominoes or rectangular blocks are lined up vertically in a chain.
2. When the chain is activated, the last domino taps a cup, which pushes out a marble or table tennis ball.
3. The marble rolls down a paper towel tube set up on an incline.



- The marble drops through a hole made in the bottom of an egg carton suspended by chenille sticks (in order to slow it down.)
- The marble lands on a target made out of building blocks.

After twenty minutes has passed, ask students to stop working. Groups take turns demonstrating their system to the rest of the class, even if it is not completely functional. Each group should describe each component in the chain and the step-by-step interaction it has with the other components. If the system is not yet functional, ask students to troubleshoot their system.

### 3. Check for understanding.

---

Unplugged  
Activity



Student groups should share their troubleshooting strategies with the class. They should identify the individual components in their chain that are not yet working and describe what effect that has on the overall chain. They should also try to explain how the issue could be resolved.

**CHALLENGE**—Time permitting, allow student groups to complete their chain reaction designs and demonstrate their fully functional system. Challenge students to add three more interactions to their chain.

## Know the Standards

*1B-CS-03 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.* Although computing systems may vary, common troubleshooting strategies can be used on all of them. Students should be able to identify solutions to problems such as the device not responding, no power, no network, app crashing, no sound, or password entry not working. Should errors occur at school, the goal would be that students would use various strategies, such as rebooting the device, checking for power, checking network availability, closing and reopening an app, making sure speakers are turned on or headphones are plugged in, and making sure that the caps lock key is not on, to solve these problems, when possible.

# Communication Between Computers

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Conceptualize the transmission of digital messages.

### Instructional Activities

- Read Aloud
- whole-class discussion

### CSTA Standards

**Concept:** Networks and the Internet

**Subconcepts and Standards:**

- **Network Communication and Organization: 1B-NI-04** Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination. (P4.4)
- **Cybersecurity: 1B-NI-05** Discuss real-world cybersecurity problems and how personal information can be protected. (P3.1)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**internet**

**network**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**hacking**

**protection**

**transport**

## Instructional Resource

Student Reader



Ch. 2

**Student Reader, Chapter 2**  
Communication Between  
Computers

## Materials and Equipment

**Collect or prepare the following:**

- classroom computer, tablet, or other computing device as needed for discussion

## THE CORE LESSON

### 1. Focus attention on today's topic.

How do computers communicate? Help students understand that as packets of information travel across the internet, it is critical that security measures are taken to protect personal information. Discuss examples of what can happen when personal information is stolen, and prompt them to name ways to protect it, such as using strong passwords, not uploading private information and photos, etc.

### 2. Read and discuss “Communication Between Computers.”

Student Reader



Ch. 2

Read together or have students read independently “Communication Between Computers,” the Student Reader. The selection introduces the idea that it is important to protect information as it travels across the internet in small packets.

#### Guided Reading Supports

When reading aloud together as a class, always prompt students to follow along. Pause for discussion. Include suggested questions and prompts:

**Page 6**

There are many different components to learning to read. Prompt students to share why they think each of the activities the buddies are doing is helpful in learning to read.

**Page 7**

Ask students if they have ever wondered how a computer gets information from one place to another. Have them discuss how they can use an account at school but also log into that same account at home.

**Page 8**

Ask students what hardware or software has a password. Ask them what they think makes a strong password.

Briefly discuss the need for cybersecurity with students, and point out that cybersecurity is an important computer science profession. Ask volunteers to describe what they think the cybercrime of identity theft is.

- (It means using someone’s private information to pretend to be them, often to take their money.)

### 3. Check for understanding.

Unplugged  
Activity



Ask students to demonstrate their understanding of the flow of information over the internet by sharing an example of

- something they enter into a computing device,
- how it travels through the internet, and
- what happens to it when it reaches its destination.

For example, they experience this sequence if they have an account for any game software or if they have used apps on a phone. Also ask students to articulate why it is important to create strong passwords.

## Know the Science

Online Resources



Information sent and received over the internet is broken down into smaller pieces called packets, which are sent independently and reassembled at the destination. Personal information should be protected physically and digitally. Cybersecurity is the protection from unauthorized use of electronic data or the measures taken to achieve this. There are physical and digital means of protecting personal information, such as locked doors, backup drives, encryption, strong passwords, biometric scanners, etc. See the Online Resources Guide for links to resources about the critical elements of cybersecurity and how to instruct students in online safety.

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

# Information Through a Computer System

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Classify and organize information based on consistent criteria.

### Instructional Activities

- whole-class activity
- whole-class discussion

### CSTA Standards

**Concept:** Networks and the Internet

**Subconcept and Standard: Network Communication and Organization: 1B-NI-04** Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination. (P4.4)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

## Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**criteria**

**information**

**internet**

**network**

**packets**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**classification                      protocol                      transport**

## Materials and Equipment

### Collect or prepare the following:

- large whiteboard
- scissors
- 3 paper clips or rubber bands

Activity Page



AP 2.2

### Activity Page 2.2 Mystery Messages (AP 2.2)

## Advance Preparation

- Make a copy of Activity Page 2.2.
- Cut each message on the lines, as shown and shuffle the messages within each set. Do not mix the sets together.
- Attach each set of pieces together with a paper clip.
- Prepare a recording display on the whiteboard. Write *Start Time:* and *End Time:* for each trial—Message A, Message B, and Message C.
- Clear a table or desk surface long enough for the message pieces to be lined up across the surface.

## THE CORE LESSON

### 1. Focus attention on today’s topic.

How does information travel through a computer system? Ask students to describe computer code. By this point in their learning, they should be able to articulate that computer code is long strings of symbols that form instructions and represent data. Prompt them to think about how information travels through the internet. The long strings of code transmit as smaller packets that are broken up and then reassembled at their destination. Invite students to share how they think the reassembly occurs. Point out that the segmented packets need to have a means to be organized and reassembled efficiently.

### 2. Facilitate the activity.

Activity Page



AP 2.2

Have students work together in groups to decode mystery words by using progressively more categorized data. Begin with the cut pieces for Trial A. Distribute the pieces to students. (Some students may get more than one.)

#### Trial A

CA	N_	YO	U_	SO
LV	E_	A_	MY	ST
ER	Y_	CO	DE	_M

Unplugged Activity



<b>ES</b>	<b>SA</b>	<b>GE</b>	<b>_W</b>	<b>IT</b>
<b>H_</b>	<b>NO</b>	<b>_C</b>	<b>LU</b>	<b>ES</b>
<b>_B</b>	<b>UT</b>	<b>_L</b>	<b>ET</b>	<b>TE</b>
<b>RS</b>	<b>?</b>			

With no additional context, ask students to join together to attempt to reassemble the message. Instruct students to form a line and file past a table surface at the front of the room to each take a turn placing their letters to assemble the first message. If they cannot decide where to place their letter, they should return to the back of the line. Mark the start time on the whiteboard, and say “GO!” Allow the line to cycle repeatedly past the table for a few minutes, then pause and ask students how the reassembly is going. Prompt them to discuss why decoding this message is challenging, such as lack of context for the message and no specific classification linking the pieces. Abandon the task if necessary, and mark the time on the whiteboard. Collect all of the pieces, and move on to the next message.

Next, distribute the cut-apart message pieces for Trial B.

### **Trial B**

1. T	2. H	3. E	4.	5. M
6. Y	7. S	8. T	9. E	10. R
11. Y	12.	13. M	14. E	15. S
16. S	17. A	18. G	19. E	20.
21. =	22.	23. P	24. A	25. C
26. K	27. E	28. T	29. S	30. !

Without any other context, cue students to reassemble the message. Mark the start time on the whiteboard. Students should quickly realize their pieces have a “protocol” on them—numbers. By putting the pieces in numerical order, they will quickly decode the message. When complete, ask them to read the message out loud, and mark the completion time on the board. Collect all of the pieces, and move on to the next message.

Finally, distribute the cut-apart message pieces for Trial C.

### **Trial C**

*1. I	*2. N	*3. T	*4. E	*5. R
*6. N	*7. E	*8. T	x1. N	x2. E
x3. T	x4. W	x5. O	x6. R	x7. K
+1. P	+2. A	+3. C	+4. K	+5. E
+6. T	+7. S	>1. C	>2. O	>3. M
>4. P	>5. U	>6. T	>7. E	>8. R

Again, without any other context, ask them to reassemble the message. Mark the start time on the whiteboard. Student pieces have a symbol category as well as a number on them. They should quickly realize that by grouping themselves by symbol and then using a numerical order, they will decode four different terms. Note the time of completion on the whiteboard.

### 3. Check for understanding.

---

Unplugged  
Activity



Review the start and completion times for the three messages. Ask students to share their explanation of why Trial A either took longer to complete or was not possible to complete. Prompt them to discuss the differences between Trials B and C, such as the effect of shorter messages and the combination of both ordering and the grouping protocol, which should have allowed them to reassemble the message most efficiently. Reiterate that when data transfers, it travels in packets that include both data and clues about how the data reassembles. It is similar to having a puzzle with pieces that not only show part of a picture but also contain clues about where in the puzzle they go.

**EXTEND**—Encourage students to create their own mystery messages for each other to reassemble using their own embedded classification system.

## Know the Standards

*1B-NI-04 Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination.* Information is sent and received over physical or wireless paths. It is broken down into smaller pieces called packets, which are sent independently and reassembled at the destination.



# Protecting Information on a Network

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Detail effective password usage.

### Instructional Activities

- whole-class discussion
- partner activity
- partner discussion

### CSTA Standards

**Concept:** Networks and the Internet

**Subconcept and Standard: Cybersecurity: 1B-NI-05** Discuss real-world cybersecurity problems and how personal information can be protected. (P3.1)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**cybersecurity**

**information**

**internet**

**network**

**packets**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**hacking**

**protection**

**transport**

## Materials and Equipment

### Collect or prepare the following:

- large whiteboard
- blank sheets of paper

## Advance Preparation

- Determine student pairs.
- Create the following table on the whiteboard:

Code 1	Code 2	Code 3
A B C D	A B C D	A B C D
	1 2 3 4	1 2 3 4

## THE CORE LESSON

### 1. Focus attention on today's topic.

Remind students that it is important that packets of information are protected as they travel across the internet, especially when they contain private personal information. Passwords are a good first step of protecting information but need to be secure if they are to be useful. They also need to be kept private and not shared with others. Ask students what makes a password strong or secure, such as type of characters used, number of characters used, and personal meaning.

### 2. Facilitate the activity.

Unplugged  
Activity



**Round 1:** Student pairs sit across from each other. On a blank sheet of paper, have students make a table with three columns and five rows. Student 1 creates and writes a three-character secret password using only A, B, C, and D with no repeats, and then hides the password in a pocket, the desk, etc. To play the game, Student 2 makes a guess in the first row, such as B, D, C. Student 1 crosses out the letters that are not correct. Student 2 then guesses again using the next row, bringing any correct guesses down to that row. For example:

**Sample table for A B C D** (where the answer = C-B-A)

B	D	C
D	C	B
A	B	D
C	B	A

Students record the time on the paper as the game begins.

Student 2 should accurately guess this code in only a few rows. They should mark the time it took to solve the secret code.

**Round 2:** On a new sheet of paper, students create another three-column chart but this time with ten rows. Point to the table on the whiteboard, and ask the class to predict how long it will take to decipher the code when using more characters. Student 2 creates and hides a three-character secret code but now can use the characters A, B, C, D, 1, 2, 3, and 4. The time is recorded, and Student 1 makes a guess in the first row. Student 2 crosses out the letters that are not correct. The activity continues as above. For example:

Sample table for **A B C D 1 2 3 4** (where the answer = 4-2-A)

A	B	C
D	A	B
C	D	A
B	C	A
D	1	A
2	3	A
1	4	A
3	2	A
4	2	A

Student 1 should accurately guess this code but require more rows to do so. Pairs should mark the completion time and compare the time to solve with the first puzzle, as well how many rows each code took to decipher.

**Round 3:** On a new sheet of paper, students now create a five-column chart with ten rows. Point to Code 3 in the table on the whiteboard, and ask the class to predict how long it will take to decipher a longer code when using more characters.

Student 1 creates and hides a five-character code using the same set of characters. The time is recorded, and Student 2 attempts to solve it in the same manner. If time permits, students should use as many rows as necessary to complete the code. Pairs should mark the finish time and compare it with the first two codes, as well how many rows each code took to decipher.

Sample table for **A B C D 1 2 3 4** (It could take as many as 56 attempts to decipher a five-character code with eight possible non-repeating characters.)

A	B	€	Ð	†
¿	3	4	A	B
€	Ð	4	†	¿

### 3. Check for understanding.

Unplugged  
Activity



Ask student pairs to share how much time and how many rows each code took. Prompt students to identify the key elements that make a password harder to guess, such as the number of different characters used and length of the code.

**EXTEND**—Extend student understanding by asking what other characters could be used to create an even more challenging code. Start a list on the board, such as # \* + > or other keyboard characters. Prompt students to think about lowercase letters, using characters more than once, etc.

**CHALLENGE**—Challenge students to create another code with even more variety and length in the characters used. Students should first agree on the set of characters to be used as well as how much to increase the code length. Students can pair up to decipher the more challenging codes.

## Know the Standards

*1B-NI-05 Discuss real-world cybersecurity problems and how personal information can be protected. (P3.1) Just as we protect our personal property offline, we also need to protect our devices and the information stored on them. Information can be protected using various security measures. These measures can be physical and/or digital.*

# Making Data Useful

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Describe management of data to support revealing patterns, answering questions, and making predictions.

### Instructional Activities

- student reading
- whole-class discussion

### CSTA Standards

**Concept:** Data Analysis

#### Subconcepts and Standards:

- **Storage: 1A-DA-05** Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data. (P4.2)
- **Collection, Visualization, and Transformation: 1B-DA-06** Organize and present collected data visually to highlight relationships and support a claim. (P7.1)
- **Inference and Models: 1B-DA-07** Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea. (P7.1)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**data**

**storage**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**abstraction**

**capacity**

**retrieve**

**visualization**

## Instructional Resource

Student Reader



Ch. 3

**Student Reader, Chapter 3**  
Making Data Useful

## Materials and Equipment

**Collect or prepare the following:**

- classroom computer, tablet, or other computing device as needed for discussion

## THE CORE LESSON

### 1. Focus attention on today's topic.

How does raw data become useful? Explain that raw data are not very useful, and typically needs to be organized and interpreted to provide meaning, find patterns, and make predictions. It is important to consider the collection tools used, the means of recording the data, and how the findings will be visualized.

### 2. Read and discuss "Making Data Useful."

Student Reader



Ch. 3

Read together or have students read independently "Making Data Useful," in the Student Reader. The selection introduces the idea that visualization techniques can help us understand data more easily.

#### Guided Reading Supports

When reading aloud together as a class, always prompt students to follow along. Pause for discussion. Include suggested questions and prompts:

**Page 10**

Prompt students to name a learning or gaming program in which they progressed into higher and higher levels. Ask them how the computer showed them their progress. Did the computer store their progress after they stopped playing?

**Page 11**

Ask students to define *data*. Remind them that data can include images, text documents, audio files, or video files. Data are not just numbers!

**Pages 12–13**

Invite students to share an example of a report they have viewed. This could be from a computer game they have played, a report card, or a bill someone received. Ask them to describe what information was depicted in the report and how it was helpful.

### 3. Check for understanding.

---

Unplugged  
Activity



Ask students to think of some visual data that they have seen before. Have them draw or explain in words the visual data. Tell them to state what was helpful with the visual data and what questions they still needed answered by only seeing visual data without any context.

## Know the Science

Online Resources



Data can include images, text documents, audio files, and video files. Raw data have little meaning on their own until they organized or grouped to provide additional clarity. Data visualization is how we examine information through graphic representations, such as graphs, charts, or tables. Data points can be clustered by commonalities to make interpreting and communicating them to others easier or to emphasize particular aspects or parts of the data set. Trends, insights, and patterns can all be understood more easily when data are reviewed visually.

See the Online Resource Guide for links to further reading about data visualization.

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

# Collecting and Storing Data

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Identify best methods and use tools for collecting and recording data.

### Instructional Activities

- whole-class discussion
- small-group activity

### CSTA Standards

**Concept:** Data Analysis

#### Subconcepts and Standards:

- **Storage: 1A-DA-05** Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data. (P4.2)
- **Collection, Visualization, and Transformation: 1B-DA-06** Organize and present collected data visually to highlight relationships and support a claim. (P7.1)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**collection**

**data**

**pattern**

**prediction**

**storage**



**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**abstraction**

**capacity**

**visualization**

## Materials and Equipment

### Collect or prepare the following:

- variety of measurement tools as needed for student data collection
- blank sheets of paper or student notebooks

## Advance Preparation

- To spark student ideas, set out a variety of measurement devices, such as rulers, tape measures, measuring cups, scales, etc.
- Determine assignments of students to groups of four or five.

## THE CORE LESSON

### 1. Focus attention on today's topic.

Ask students to define *data*, and give examples of types of data. Prompt them to see sources of data in the classroom, such as a reading log or a class job chart. Inquire if those data sources are easy to comprehend and, if so, what makes them clear and useful. Ask students how the data were collected and recorded.

### 2. Facilitate the activity.

Unplugged  
Activity



Students work in small groups to collect at least twenty-five unique data points of their choice. Student groups work together to collect and record the data using non-computing devices. Student groups should also think about how they will present their data to the class in Lesson 3.3.

1. Work together to discuss the type of data student groups want to collect, such as classmates' favorite hot lunch item, favorite P.E. activity, daily temperature, length of pencils used in the classroom, etc. The groups need to consider something they can collect twenty-five times. Students should discuss if their data are based on fact or opinion.
2. Determine how the groups will collect their chosen data, such as a thermometer, a ruler, a paper survey, a verbal question, a class list, etc., and from whom or what (classmates, other grades, family members, class pets, teachers, classroom artifacts, etc.).
3. Create a way to record the data students collect, such as a numbered list or a labeled table.
4. Consider what groups want to depict with their data, such as a bar graph showing the most and least popular items or events over time or a chart organizing all of the data. Students should briefly ideate how their group will visualize their findings.

5. Use available tools and their data record sheet to collect and record at least twenty-five pieces of meaningful data. For example:

Name	Favorite P.E. Activity
Selena	Four Square
Oscar	Tetherball
Tara	Capture the Flag

### 3. Check for understanding.

---

Unplugged  
Activity



After students have completed their data collection and recording, ask them to engage in a small-group discussion about their findings. Prompt students to look for trends and patterns. Time permitting, ask each group to make a prediction about what another group's data will provide just by looking at the data given.

## Know the Standards

*1A-DA-05 Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data. All information stored and processed by a computing device is referred to as data. Data can be images, text documents, audio files, software programs or apps, video files, etc. As students use software to complete tasks on a computing device, they will be manipulating data.*

# Looking at Data

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objectives

- ✓ Organize data sets to reveal patterns and highlight relationships.
- ✓ Use data patterns to support claims.

### Instructional Activities

- whole-class discussion
- small-group activity
- small-group presentation

### CSTA Standards

**Concept:** Data Analysis

#### Subconcepts and Standards:

- **Storage: 1A-DA-05** Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data. (P4.2)
- **Collection, Transformation, and Visualization: 1B-DA-06** Organize and present collected data visually to highlight relationships and support a claim. (P7.1)
- **Inference and Models: 1B-DA-07** Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea. (P7.1)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

## Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

collection

data

pattern

prediction

storage

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**abstraction**

**capacity**

**retrieve**

**visualization**

## Materials and Equipment

### Collect or prepare the following:

- blank sheets of paper or student notebooks
- markers, colored pencils, crayons, etc.
- data sheet prepared in Lesson 3, Part 2
- classroom computer, tablet, or other computing device with internet access and projection capability
- software/apps to create different types of graphs

## Advance Preparation

- Prepare a few images depicting data visualization to present via projector. Print out examples if computer/internet access is not available. See the Online Resources Guide for links to good examples.  
[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

## THE CORE LESSON

### 1. Focus attention on the lesson question.

How can data be gathered, organized, and presented to communicate accurately and make helpful predictions? As a whole class, remind students that generally raw data have very little use on their own but that once they are organized and represented visually, patterns and meaning can be revealed and predictions can be made. Ask student groups to share a pattern prediction of what they will find with their data.

Device-Based  
Activity



If computer and internet access is available, show the class some online examples of tables, charts, and graphs. Use printed copies if computer and internet access is not available. Discuss the similarities and differences between each type of data visualization.

### 2. Facilitate the activity.

Unplugged  
Activity



Device-Based  
Activity



Using the factual data collected in the previous lesson, small groups find and use the best way to visualize the data collected in Lesson 3.2.

1. Discuss their goals of what they are trying to determine with their data and what they predict will happen. For example, "Four Square will be the most popular P.E. activity."
2. Carefully review their data to analyze if they believe it will show what they predicted.
3. Consider a few ways they could visually represent their data, such as a colorful bar graph or a table with different icons.

They will select the best visualization tool to highlight the relationships they found or support a claim they wish to make.

4. Collect materials they will need, such as markers or colored pencils, and create their data visualization on paper (if computers or tablets are not available).
5. Complete at least two representations of their data in a visual format.
6. Discuss if their predictions were correct and what other patterns or meaning they revealed.

### 3. Check for understanding.

---

#### Device-Based Activity



When all are finished, student groups share their data visualizations with the class, using the projector if possible. Groups should be sure to first explain how they collected their data and then show their data record. They should describe their data in detail, including any trends or patterns. They should indicate if their findings match the predictions they made at the beginning of the lesson. Ask student groups which visualization showed the most useful information.

**EXTEND**—Extend student understanding by prompting them to use a different means of visualizing their findings, such as another chart or by using a construction toy. Does a different visualization uncover any different findings?

## Know the Standards

*1B-DA-06 Organize and present collected data visually to highlight relationships and support a claim.* Raw data has little meaning on their own. Data is often sorted or grouped to provide additional clarity. Organizing data can make interpreting and communicating easier. Data points can be clustered by a number of commonalities. The same data could be manipulated in different ways to emphasize particular aspects or parts of the data set.

*1B-DA-07 Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.* The accuracy of data analysis is related to how realistically data are represented. Inferences or predictions based on data are less likely to be accurate if the data are not sufficient or if the data are incorrect in some way. Students should be able to refer to data when communicating an idea.

# Collaboration

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

✓ Describe the elements of a successful team.

### Instructional Activities

- student reading
- whole-class discussion

### CSTA Standards

**Concept:** Algorithms and Programming

**Subconcepts and Standards:**

- **Program Development: 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)
- **Program Development: 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)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**collaboration**

**plan**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**iterative**

**perspective**

**program development**

## Instructional Resource

Student Reader



Ch. 4

**Student Reader, Chapter 4**  
Collaboration

## Materials and Equipment

**Collect or prepare the following:**

- classroom computer, tablet, or other computing device as needed for discussion

## THE CORE LESSON

### 1. Focus attention on today's topic.

Why are collaboration and respect important when developing a computer program? Discuss with students how team members often take on individual roles and work together and consider each other's perspectives to successfully complete a program.

### 2. Read and discuss "Collaboration."

Student Reader



Ch. 4

Read together or have students read independently "Collaboration," in the Student Reader. The selection introduces the idea that planning and collaboration are key components to developing a computer program.

#### Guided Reading Supports

When reading aloud together as a class, always prompt students to follow along. Pause for discussion. Include suggested questions and prompts:

**Page 14**

Ask students if they have ever had a great idea. What prompted that idea? Was it improving something? Was it coming up with a different way to use something? Was it combining something with something else?

**Page 15**

Ask students why creating a plan is important for any project. Ask them how a plan can help a project go more smoothly and quickly. Can they give any examples of something "not going to plan"?

**Page 16**

Ask students to define *collaboration*. What do they feel are important aspects of being a good collaborator? Prompt them with items such as listening to one another, sharing the work, sharing ideas, etc.

Pair students up, and ask them to read the items in the team’s plan and discuss the questions on the page. Encourage them to talk about both the team’s plan itself and how the team will work together, or collaborate.

### 3. Check for understanding.

---

Unplugged  
Activity



After the pairs read and discuss the team’s plan, ask the pairs to share what they thought of the plan with the class. Did they think of any other tasks the team did not list? Who do they think would enjoy each of the tasks most? Also ask them to share their thoughts on how the team can collaborate effectively. Should each member only have one role? Should everyone contribute equally on a group project?

## Know the Science

Online Resources



Planning and collaboration are important parts of the iterative process of program development. Collaborative computing asks for the contributions and feedback of others and can often lead to better (and faster) outcomes than working independently. Help students understand that communication, conflict resolution, and task management are all necessary parts of creating a plan before beginning to build a program. This might include outlining key features and listing time and resource constraints and user expectations. See the Online Resource Guide for links to information about helping students learn to collaborate effectively.

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)



# Including People’s Ideas

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Use teamwork, respectful communication, and varying roles to achieve a goal.

### Instructional Activities

- whole-class discussion
- small-group activity

### CSTA Standards

**Concept:** Algorithms and Programming

#### Subconcepts and Standards:

- **Program Development: 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)
- **Program Development: 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)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

## Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**collaboration      planning      respect      role      teamwork**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**iterative      perspective      program development**

## Materials and Equipment

### Collect or prepare the following:

- None

## Advance Preparation

- Determine a start and end point, and prepare a clear path for students to navigate around the classroom.

## THE CORE LESSON

### 1. Focus attention on today's topic.

How can diverse perspectives be integrated to design the best possible solution? Ask students to share one element of being a good team member, such as listening to others, perspective, respect, trusting each other, working hard, or taking on individual roles. Can they name some of the roles involved in various teams, such as a soccer team, a computer program development team, a restaurant staff, or a book club? Prompt students to consider how teams must work together and be respectful to one another's perspective in order to be successful.

### 2. Facilitate the activity.

Unplugged  
Activity



Using the same groups from Lessons 3.2 and 3.3, teams will work together to guide a team member with eyes closed through a maze using only verbal instructions.

1. Demonstrate the starting point, course, and end point of the classroom maze. (If space permits, an outdoor course can also be created in a play area or using hula hoops, cones, etc.)
2. Teams decide who will take on one of the following roles (students may need to take on more than one role, depending on group size):
  - a. Mazerunner
  - b. Forward mover
  - c. Stopper
  - d. Left turner
  - e. Right turner
3. The mazerunner goes to the starting line and keeps eyes closed.
4. The other team members carefully navigate the mazerunner through the course by calling out only four commands: "forward," "stop," "left turn," or "right turn." Team members try to be clear, not speak over one another, and prompt each other when a call is missed.
5. The team guides the mazerunner through the course until they reach the finish line.

6. Other teams get their turn on the course using the same instructions.
7. Time permitting, teams can switch roles and navigate a different team member through the course.

### 3. Check for understanding.

---

Unplugged  
Activity



Prompt student groups to share their experience with teamwork—respect, trust, listening, and clear communication. Ask:

- Mazerunners: Did you feel trust in your teammates to guide you safely?
- Left or right turners: Was there any time you disagreed with the call being given? How did you handle it?
- Forward movers: Did you ever realize you gave the wrong call? What did you do?
- Stoppers: Did you ever have to speak over anyone to avoid a collision?

**CHALLENGE**—Challenge students by grouping the class into larger teams and creating a larger maze/obstacle course outside. Come up with a variety of other roles to give each team member, such as Jump, Climb, Bend, Crawl, etc.

## Know the Standards

*1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.* Planning is an important part of the iterative process of program development. Planning a computer program includes outlining key features, time and resource constraints and user expectations, and documenting the plan.

*1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.* Collaborative computing is the process of performing a computational task by working in pairs or on teams. Because it involves asking for the contributions and feedback of others, effective collaboration can lead to better outcomes than working independently.

*1A-IC-17 Work respectfully and responsibly with others online.* Online communication facilitates positive interactions, such as sharing ideas with many people, but the public and anonymous nature of online communication also allows intimidating and inappropriate behavior in the form of cyberbullying. Students should take care to avoid sharing information that is inappropriate or that could personally identify them selves to others. They should provide feedback to others on their work in a kind and respectful manner and tell an adult if others are sharing inappropriate things or are treating others in an unkind or disrespectful manner in online collaborative spaces.

# Designing Programs

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Explain why breaking down a task efficiently and accurately is important for completing a task.

### Instructional Activities

- student reading
- whole-class discussion

### CSTA Standards

**Concept:** Algorithms and Programming

#### Subconcepts and Standards:

- **Algorithms: 1B-AP-08** Compare and refine multiple algorithms for the same task and determine which is the most appropriate. (P6.3, P3.3)
- **Variables: 1B-AP-09** Create programs that use variables to store and modify data. (P5.2)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**algorithm**

**compare**

**variable**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**design**

**refine**

## Instructional Resource

Student Reader



Ch. 5

### Student Reader, Chapter 5 Designing Programs

## Materials and Equipment

### Collect or prepare the following:

- classroom computer, tablet, or other computing device as needed for discussion

## THE CORE LESSON

### 1. Focus attention on today's topic.

How are computer programs designed? Explain why breaking down a task most efficiently and accurately is important for a computer to know how to complete a task. There can be many different algorithms to achieve the same result, but some are much better than others, depending on the situation. Algorithmic thinking allows students to break down problems and think of solutions in terms of discrete steps. Explain that variables can represent any type of data—numbers, characters, images, etc.

### 2. Read and discuss “Designing Programs.”

Student Reader



Ch. 5

Read together or have students read independently “Designing Programs,” in the Student Reader. The selection introduces the importance of comparing multiple designs to find the most efficient and accurate way to complete a computer task.

### Guided Reading Supports

When reading aloud together as a class, always prompt students to follow along. Pause for discussion. Include suggested questions and prompts:

#### Page 18

Ask students to define what design means to them. Is designing something only done in art? Prompt them to discuss the unique design roles each of the team members will take on and which one they can relate to most.

#### Page 19

Ask students why it is important to determine the roles needed to build a computer program. Invite students to share which roles are most appealing to them so far.

What were the most important variables the students wanted to collect? Ask students if they can think of other variables that would be valuable to collect.

#### Page 20

Invite students to give examples of a helpful way to compare two things, such as the flavor of two or more ice pops to see which they like more or which new pants fit better.

Prompt students to carefully review and reflect on each of the two program designs. Encourage them to look back to the feedback the team received from the teachers.

### 3. Check for understanding.

---

Unplugged  
Activity



After students look over the two program designs, ask them to share their opinion of which will be easiest for the first graders to use and which will best give the teachers the information they need. Prompt students to explain how the comparison was important to the program development process.

## Know the Science

Online Resources



Different algorithms can achieve the same result, though sometimes one algorithm might be more appropriate for a specific solution. Examining which variables should be used to store and modify data and how those variables should be sequenced and interact is all part of the design process. The design process can vary depending on the type of project but generally consists of Imagining/Brainstorming, Planning, Designing, Building, Testing, and Sharing. A design process can help developers determine which is the best solution for a specific scenario. See the Online Resource Guide for a link to learn more about the design process as well as a helpful visual.

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

# Decision Trees

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objectives

- ✓ Create a decision tree.
- ✓ Compare the efficacy of two different decision trees.

### Instructional Activities

- whole-class discussion
- small-group activity

### CSTA Standards

**Concept:** Algorithms and Programming

**Subconcepts and Standards:**

- **Algorithms: 1B-AP-08** Compare and refine multiple algorithms for the same task and determine which is the most appropriate. (P6.3, P3.3)
- **Variables: 1B-AP-09** Create programs that use variables to store and modify data. (P5.2)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**algorithm**

**compare**

**decision tree**

**design**

**yes/no questions**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**branching**

**comparison**

**group**

**pattern**

## Materials and Equipment

### Collect or prepare the following:

- large whiteboard or anchor chart
- colored whiteboard markers
- pattern blocks (one set of each of the six shapes per pair of students)
- pencils and drawing paper

## Advance Preparation

- Set up the same classroom course created for Lesson 4.2.
- Set up the classroom projector.
- Sort the pattern blocks, with one set per pair of students.

## THE CORE LESSON

### 1. Focus attention on today's topic.

How do you create a decision tree? Remind students that an algorithm is a sequence of discrete steps for solving a problem or completing a task.

Explain that different algorithms can achieve the same result and that it is helpful to generate multiple algorithms to determine which is more successful.

Explain that the steps in an algorithm can include yes/no questions that help us decide what the next steps are going to be. They can also be used to put things into different categories or groups. Provide some examples to the class, such as “Is today Monday?” or “Is my shirt purple?”

On the whiteboard or anchor chart, draw an orange square, a green triangle, and a yellow hexagon. Explain that you are going to pick one of the three shapes in secret. **Ask:** What are some yes/no questions you could ask to figure out which shape I've selected? Record responses on the board. Point out that there are two different ways to distinguish between the shapes you drew: by color and by shape. Highlight any of the student responses that refer to the objects' color or shape.

Next, explain that we can make a decision tree to decide which shape I've picked. A decision tree is used to make decisions based on the answers to yes/no questions.

Tell the students that you will start with the question “Is the shape orange?” Write the question on the board, and enclose it with a rectangle. (See **Know the Science**.) Draw two arrows below it. Label one arrow “yes” and one arrow “no.” Since only one

## Know the Science

**Decision Trees** A decision tree is a map of possible outcomes of a series of related choices. It consists of a root node that is a starting point question; branches that indicate a possible outcome or action based on a decision; decision nodes that split off from the starting point question; and leaf, or endpoint, nodes, which indicate a final outcome. In standard flowchart programs, the root node is represented by a rectangle, the branches are represented by arrows, the decision nodes are represented by either a rectangle or a circle, and the leaf nodes are represented by an inverted triangle.



of the choices is orange, if the answer is “yes,” it must be a square. Draw a triangle at the end of the “yes” arrow, and write “It is a square” inside. If the answer is “no,” you have to ask another yes/no question. Draw a rectangle at the end of the “no” question, and write “Is the shape a triangle?” For “yes,” write “It is a triangle” inside a triangle. For “no,” write “Is the shape yellow?” inside a rectangle. Since there are no other options, draw only a “yes” line, and write “It is a hexagon” inside a triangle.

Point out that we call this a decision tree because it looks like a branching tree. We ask each of the equations in order as we move from the top to the bottom of the tree. Have students work through the decision tree questions aloud until they have figured out the shape you have selected.

## 2. Facilitate the activity.

Unplugged  
Activity



Tell students they will make their own decision trees to figure out a shape. Divide the class into pairs, and distribute sets of pattern blocks, pencils, and drawing paper to each pair.

Tell student pairs they should choose a colored shape from the six pattern blocks in secret. They will then each make their own decision tree for that shape so that another team can figure out which piece they picked. Remind them that the tree starts with a yes/no question and that for this activity they can either ask questions about the shape (e.g., “Is it a square?”) or about the color (e.g., “Is it a blue?”). Encourage students to use rectangles to show questions and triangles to show “yes” answers. The final trees should show six possible outcomes.

## 3. Check for understanding.

Have students compare their decision trees with their partner. **Ask:** Do all of your trees have the same number of branches? Do yours and your partner’s trees have the same yes/no questions, or were there multiple ways to get to the same result?

After the partner comparison, have pairs choose one of the decision trees and then team up with another pair to try to guess each other’s shape. Were students able to use the decision tree to determine the other team’s shape, or were any modifications needed?

## Know the Standards

*1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.* Different algorithms can achieve the same result, though sometimes one algorithm might be most appropriate for a specific situation. Students should be able to look at different ways to solve the same task and decide which would be the best solution.

*1B-AP-09 Create programs that use variables to store and modify data.* Variables are used to store and modify data, such as using mathematical operations to add to the score of a game or using a variable as a countdown timer.

# Variables

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Recognize the prevalence of variable factors and the sources of variability.

### Instructional Activities

- whole-class discussion
- independent activity
- whole-class activity

### CSTA Standards

**Concept:** Algorithms and Programming

**Subconcept and Standard: Variables: 1B-AP-09** Create programs that use variables to store and modify data. (P5.2)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**modify**      **predict**      **value**      **variable**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**manipulate**      **prediction**

## Materials and Equipment

### Collect or prepare the following:

- large whiteboard
- computer access with a word-processing application
- printer access, preferably color
- blank sheets of paper
- blue, yellow, pink, orange, and green highlighters
- scissors

## Advance Preparation

- Set up computer and printer access for individual students to take turns. On the whiteboard, write and color-code the following legend:

Name = Blue

Food = Yellow

Activity = Pink

Animal = Green

Goal = Orange

## THE CORE LESSON

### 1. Focus attention on today's topic.

Ask students to share their understanding of variables. Remind the class that a variable is like a container that holds information. Each variable has a name that remains constant and contains a value that can be modified in order to manipulate data. Variables can represent any kind of data—numbers, characters, words, images, etc.

### 2. Facilitate the activity.

#### Device-Based Activity



1. If access to computing devices is available, students should open up a new document in a word-processing application. If computer devices are not available, students can use paper and pencil to write their stories.
2. Tell students they will be typing a short story (three or four sentences) about themselves. Their story must contain their:
  - \*Name
  - \*Favorite food
  - \*Favorite thing to do (sport, activity, game, etc.)
  - \*Favorite animal
  - \*Future goal

3. Stories should be typed in size 50 font and black text.

*For example:*

*My name is **Tina**. My favorite food is **spaghetti**. I enjoy **doing yoga**. My favorite animal is a **lion**. When I grow up, I want to be a **computer programmer**.*

4. Students should be sure to save their document so they can open it later.
5. Students should then remove five sections from their story and place those words (or variables) on a separate page. Blanks should be added in place of the removed words.

*For example:*

*My name is \_\_\_\_\_. My favorite food is \_\_\_\_\_. I enjoy \_\_\_\_\_. My favorite animal is a \_\_\_\_\_. When I grow up, I want to be a \_\_\_\_\_.*

6. On the variable page, they should use the font color or highlighter tool for each of the removed words/phrases:

Name = Blue

Food = Yellow

Activity = Pink

Animal = Green

Goal = Orange

7. Students print their stories and their variable pages. If a color printer is not available, print in black and white and then use a highlighter or marker to color-code.
8. Students cut out their variable cards and hand them to the teacher. Sort the cards into variable piles (Name, Food, etc.), and shuffle.
9. Pass out a set of five cards—one from each variable pile—to each student. Students place them in the correct variable blank.
10. Students take turns reading the humorous mixed-up student stories. For fun, they can mix up the variable values and read the silly stories.

*For example:*

*My name is **Zoe**. My favorite food is **basketball**. I enjoy playing **hamburger**. My favorite animal is a **soccer player**. When I grow up, I want to be a **toucan**.*

11. As time permits, variable cards can be reshuffled and redistributed to create more funny student stories.

**EXTEND**—Keep student card piles together, and read all but the name sentence. See if students can predict what the name will be based on the other values.

### 3. Check for understanding.

---

Unplugged  
Activity



Ask students why it is funny to hear mismatched values associated with a classmate's name. Confirm students understand that the values in a variable can always change, even though the name of the variable remains the same.

**EXTEND**—Pair up students, and on a blank sheet of paper, have them create their own phrases with two variables. For example, “The \_\_\_\_\_ eats \_\_\_\_\_.” Using index cards, they write one value on one side, such as “Panda.” On the other side, they write the second value, such as “Bamboo.” They create at least three other cards and then share their variable phrases and cards with another student pair. Pairs try to guess the second value on the back of each card.

## Know the Standards

*1B-AP-09 Create programs that use variables to store and modify data. Variables are used to store and modify data, such as using mathematical operations to add to the score of a game or using a variable as a countdown timer.*

# Computer Code

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Identify computer commands to instruct a computer to do simple tasks.

### Instructional Activities

- student reading
- whole-class discussion

### CSTA Standards

**Concept:** Algorithms and Programming

#### Subconcepts and Standards:

- **Modularity: 1B-AP-11** Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. (P3.2)
- **Control: 1B-AP-10** Create programs that include sequences, events, loops, and conditionals. (P5.2)
- **Modularity: 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)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**code**

**condition**

**event**

**loop**

**sequence**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**control**

**decomposition**

**execution**

**remix**

## Instructional Resource

Student Reader



Ch. 6

**Student Reader, Chapter 6**  
Computer Code

## Materials and Equipment

**Collect or prepare the following:**

- classroom computer, tablet, or other computing device as needed for discussion

## THE CORE LESSON

### 1. Focus attention on today's topic.

How is computer code written? Help students comprehend how the individual components of computer code come together to provide a computer with the discrete steps needed to complete tasks. Ask students to describe other areas in their lives where they are given step-by-step instructions and need to complete them in a particular order (e.g., building a piece of furniture, completing a school project).

### 2. Read and discuss "Computer Code."

Student Reader



Ch. 6

Read together or have students read independently "Computer Code," in the Student Reader. The selection introduces the idea that a sequence of discrete commands instructs a computer to do simple tasks.

#### Guided Reading Supports

When reading aloud together as a class, always prompt students to follow along. Pause for discussion. Include suggested questions and prompts:

**Page 22**

Invite students to share their definition of what coding a computer means. Ask them to give one simple task that a computer might do in a reading log program, such as enter your name, select their grade, or enter a book title.

**Page 23**

Ask students to name tasks they do over and over, such as making their beds, doing the dishes, getting a snack, etc. Why are loops helpful for repetitive tasks? Ask students to define *events* and *conditions*.

Confirm an event is a user-defined action that results from a particular occurrence, such as a program opening or a keystroke. A condition is an instruction that tells the computer to execute certain actions if certain criteria are met.

## Page 24

Ask students if they are familiar with the term *remix* from music and if they can name a song that was remixed. If possible, play that song after playing the original. Ask students to raise their hand each time they hear a remixed part of the song. Prompt them to define *remix* as creating a different version of something by adding, removing, or changing pieces of it.

## Page 25

If time permits, pair students up, and instruct them to look closely at the diagram and discuss how the team used events, loops, and conditions. Ask them to closely analyze the sequence of steps the team used. Students should spot two loops and two conditions.

### 3. Check for understanding.

---

Unplugged  
Activity



If students collaborate with a partner to analyze the team's algorithm, ask the pairs to write their responses to the following questions on paper. Ask student pairs to share their responses with the whole class.

- How many loops can you identify?
- How many conditions?
- How many events?
- Is there anything you would change to make the algorithm more efficient?

## Know the Science

Online Resources



Computer code is made up of a series of small steps that when put together in a particular order, instruct a computer on what tasks to perform and when to perform them. Programmers must first break down each task into smaller parts, called decomposition. These parts can be incorporated into new or existing programs. The order of the tasks is specified and can be combined to create more complex tasks. Events allow portions of a program to run based on a specific action. Conditions allow for the execution of a portion of code in a program when a certain condition is true. Loops allow for the repetition of a sequence of code multiple times. See the Online Resources Guide for a link to resources to learn more about basic programming concepts.

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)



# Instructions for Actions

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Execute coding fundamentals to complete a task using events, sequences, conditions, and loops.

### Instructional Activities

- whole-class discussion
- partner activity
- whole-class presentations

### CSTA Standards

**Concept:** Algorithms and Programming

#### Subconcepts and Standards:

- **Modularity: 1B-AP-11** Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. (P3.2)
- **Control: 1B-AP-10** Create programs that include sequences, events, loops, and conditionals. (P5.2)
- **Modularity: 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)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

## Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**condition**

**event**

**loop**

**sequence**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**control**

**execute**

## Materials and Equipment

### Collect or prepare the following:

- classroom projector
- scissors
- blank sheets of paper
- colored markers

Activity Page



AP 6.2

### Activity Page 6.2

Instructions for Actions (AP 6.2)

## Advance Preparation

- Make sufficient copies of AP 6.2 for student pairs.
- A completed sample algorithm for AP 6.2

## THE CORE LESSON

### 1. Focus attention on today's topic.

How does a computer know to perform a specific action? Algorithms contain the precise sequence of discrete steps that a computer is told to perform. Ask students if they can define other fundamental coding terms, such as *loops*, *events*, and *conditions*. Remind as needed that events tell a part of a program to run based on a specific action. Conditions tell a part of the program to run when a certain condition is true. Loops tell a part of the program to repeat multiple times.

### 2. Facilitate the activity.

Unplugged Activity



Place students into pairs. Provide a copy of AP 6.2 Instructions for Actions to each group. Tell them that an alien has never seen a bike before but wants to learn to ride it around the block. Prompt students to read and reflect on each element and think about the level of detail of each step. Refer students to Student Reader page 25 for a reminder example of the kind of diagram they will produce.

Activity Page



AP 6.2

Instruct student pairs to cut out and color code individual pieces of paper for each step in an algorithm diagram, or mind map. They should write the steps from the list provided on AP 6.2. On a blank sheet of paper, students work together to lay out an algorithm for how they would teach the alien to ride a bike. Encourage them to get up and demonstrate the movements to one another in order to think through the proper sequence.

Give students a hint that they should create at least one loop, one event, and one condition. They should draw arrows on their paper to connect the steps and indicate loops, events, or conditions. They are welcome to draw additional elements in their algorithm as desired. Remind students that there is usually more than one way to create an algorithm.

When students are confident that their algorithm will successfully help the alien return to the initial starting point, they can use glue to secure it to their paper.

### 3. Check for understanding.

---

Unplugged  
Activity



Project the completed sample algorithm to the class. Review each step in the sequence, and ask students if they had other ways to order the tasks. Point out the loops, events, and conditions, and ask students to explain why the alien needs those to reach its goal. Time permitting, invite student pairs to share their own solutions on the projector, especially those that have a different sequence or original steps.

**CHALLENGE**—Challenge students to break down the steps and add a right turn into their algorithm.

## Know the Standards

*1B-AP-10 Create programs that include sequences, events, loops, and conditionals.* Control structures specify the order (sequence) in which instructions are executed within a program and can be combined to support the creation of more complex programs. Events allow portions of a program to run based on a specific action. Conditions allow for the execution of a portion of code in a program when a certain condition is true. Loops allow for the repetition of a sequence of code multiple times.

# Modification

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objectives

- ✓ Incorporate smaller portions of existing programs to develop something new.

### Instructional Activities

- whole-class discussion
- partner activity
- whole-class presentations

### CSTA Standards

**Concept:** Algorithms and Programming

**Subconcepts and Standards:**

- **Modularity: 1B-AP-11** Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. (P3.2)
- **Modularity: 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)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**remix**

**reuse**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**control**

**execution**

**modification**

## Materials and Equipment

### Collect or prepare the following:

- mind map from Lesson 6.2
- highlighter
- blank sheets of paper

## Advance Preparation

- Complete Lesson 6.2 with students.

## THE CORE LESSON

### 1. Focus attention on the lesson question.

How can a program be modified to do something different? Ask students if the algorithm they wrote for the bike-riding alien could be used to teach it to bake a cake. Why not? Next, ask students if there would be any similarities if they were to teach the alien to ride a stationary bike. Which steps would be the same (pedaling, holding the handlebars, etc.)? Which would be different (braking, using the kickstand, etc.)?

### 2. Facilitate the activity.

Unplugged  
Activity



The alien's friend also wants to get to the park, but it wants to learn to ride a scooter instead of a bike. Tell students they will write a new algorithm for the alien's friend, but their goal is to reuse as much of their bike-riding code as possible.

The same student pairs discuss how riding a bike and riding a scooter are similar and different. Encourage them to get up and demonstrate the movements to one another in order to think through the proper sequence. Partners use blank sheets of paper to draw a new algorithm for the alien's friend. Partners should first copy as much as they can from their original bike-riding algorithm (such as holding the handlebars). Partners then color the reused portions of the algorithm with their highlighter.

Partners then create new distinct steps the alien's friend needs for learning to ride a scooter (such as replacing pedaling with leg pushing). Encourage students to try to reuse loops, events, and conditions whenever possible. Remind students that there is usually more than one way to create an algorithm.

### 3. Check for understanding.

---

Unplugged  
Activity



Ask students to count how many steps were new and how many were reused in their new algorithms. Ask students if it would have been easier to start a new sequence from scratch or if it was more efficient to incorporate portions of the previous algorithm to develop the new program. Do they think the same would be true if they were writing computer code?

**CHALLENGE**—Challenge students to create a new algorithm for teaching an alien other similar tasks, such as roller-skating or riding a skateboard to the park. Can they reuse more pieces of their code than they did for the scooter algorithm?

## Know the Standards

*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.* Programs can be broken down into smaller parts, which can be incorporated into new or existing programs. For example, students could modify prewritten code from a single-player game to create a two-player game with slightly different rules, remix and add another scene to an animated story, use code to make a ball bounce from another program in a new basketball game, or modify an image created by another student.

# Testing and Usability

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Describe how testing and usability research help programs meet the needs of the most users.

### Instructional Activities

- student reading
- whole-class discussion

### CSTA Standards

#### Concepts:

- Algorithms and Programming
- Impacts of Computing

#### Subconcepts and Standards:

- **Program Development: 1B-AP-15** Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. (P6.1, P6.2)
- **Culture: 1B-IC-19** Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users. (P1.2)
- **Social Interactions: 1B-IC-20** Seek diverse perspectives for the purpose of improving computational artifacts. (P1.1)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

## Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**accessibility**

**debug**

**testing**

**usability**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**equitable**

**iterative**

## Instructional Resource

Student Reader



Ch. 7

**Student Reader, Chapter 7**  
Testing and Usability

## Materials and Equipment

**Collect or prepare the following:**

- classroom computer, tablet, or other computing device as needed for discussion

## THE CORE LESSON

### 1. Focus attention on today's topic.

Why are testing and usability important for a computer program? Discuss how a thorough testing program can ensure that a program is running properly and help to find and fix bugs as well as make modifications to increase usability for diverse users, including those with disabilities. Ask students to share examples of products they have used that do not function properly or have broken while they are using them. Then, ask students to share examples of products, apps, and physical devices that accommodate users that may have unique needs.

### 2. Read and discuss "Testing and Usability."

Student Reader



Ch. 7

Read together or have students read independently "Testing and Usability," in the Student Reader. The selection introduces the idea that the students understand that testing is a crucial step in developing a program to be sure it works as it is supposed to for a variety of users.

### Guided Reading Supports

When reading aloud together as a class, always prompt students to follow along. Pause for discussion. Include suggested questions and prompts:

**Page 26**

Ask students why testing is important when developing a computer program. Prompt them to discuss ways that user testing is most valuable, such as with a large group of users and with as diverse a group of users as possible.



**Page 27**

Ask students why they think testers mark down every time something doesn't function properly. They should give some examples of things that may not function properly in a computer program. Next, ask them why it is important to know how often a problem happens. Which items are the most important to debug?

**Page 28**

Ask students to define *accessibility* and *usability*. How does Mrs. Hale's team improve their programs for individuals with visual or auditory impairments?

### 3. Check for understanding.

---

Unplugged  
Activity



Invite students to share ways that they think computer devices or the programs they run might be hard for some people to use because of individual differences. Then ask students to brainstorm how the same technology can be modified to make it possible for people with different needs and abilities to do certain tasks that they otherwise may not be able to perform.

## Know the Science

Online Resources



Programs do not always run properly. Students need to understand how to test and successfully identify and fix errors in (debug) their programs to ensure they run properly and with optimal ease of use, or usability. In order to enable global collaboration and sharing of ideas, computing devices need to be accessible to as many users as possible—those with as many different backgrounds, ability levels, points of view, and physical limitations as possible. Developing and modifying computing technology should be driven by people's needs and wants. Soliciting feedback from a diverse group of users helps improve the usability and accessibility of computer programs. User surveys, video conferencing tools, shared documents, and other online collaborative spaces, such as blogs, forums, or website comments, can help gather and synthesize feedback from individuals and groups about programming projects. See the Online Resources Guide for links to useful resources for learning more about usability and accessibility.

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

# Testing and Debugging

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objectives

- ✓ Test and gather data about a solution.
- ✓ Determine if solutions meet intended criteria and constraints.

### Instructional Activities

- small-group activity
- small-group discussion
- whole-class discussion

### CSTA Standards

#### Concepts:

- Algorithms and Programming
- Impacts of Computing

#### Subconcepts and Standards:

- **Program Development: 1B-AP-15** Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. (P6.1, P6.2)
- **Culture: 1B-IC-19** Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users. (P1.2)
- **Social Interactions: 1B-IC-20** Seek diverse perspectives for the purpose of improving computational artifacts. (P1.1)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**debug**

**testing**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**iterative**

## Materials and Equipment

### Collect or prepare the following:

- blank sheets of paper
- open and safe areas to navigate within school campus, such as indoor or outdoor corridors

## Advance Preparation

- Predetermine three or four safe destinations on the school campus for students to navigate to from the classroom, such as the library, office, gym, basketball court, etc. If possible, try to include a level change, such as stairs, as well as opening a door or gate.

## THE CORE LESSON

### 1. Focus attention on today's topic.

How can a program be tested and modified to ensure it is working as expected? Ask students if their previous algorithms worked the first time they tried them out. Lead a group discussion on what it means to test a product or program. Ask students to define the term *debug*.

Remind them that any program should be thoroughly tested in a systematic way to be sure it works properly and to fix any problems or bugs.

### 2. Facilitate the activity.

Unplugged  
Activity



1. Place students into teams of three or four. (It is recommended that the same teams from Lessons 3.2, 4.2, and 5.2 remain together.)
2. Instruct teams that they will design a different algorithm using the same coding system they developed in Lesson 5.2.
3. Provide students with your list of approved locations at the school. Explain that there is no running, only walking through the route.
4. Student teams work together to walk through and code each discrete step required to navigate from the classroom to their chosen destination. They draw the same symbols for left, right, forward, and backward that they created previously, but they will add new coding for other aspects, such as stairs, opening a door, etc.
5. Teams write out their algorithm in as specific detail as possible and test it exactly with one member of their group being the walker and others calling out or taking notes. Another teammate should always accompany the walker to ensure they do not bump into anything, fall, etc.

6. Once they are satisfied, they create a copy of their algorithm on a new sheet of paper, but they change, remove, or add steps so that this version has two or three incorrect steps.
7. Teams trade algorithms with another team and test out each other's route, making sure a team member is accompanying the walker.
8. Teams try to uncover the bugs in each other's code.
9. Teams then debug the incorrect code by providing the fix to make the route correct.

### 3. Check for understanding.

---

Unplugged  
Activity



Each team provides the other with the original, correct version of their algorithm. Teams compare their debugged version with the original to see how well they found and fixed all the bugs.

**EXTEND**—Provide students with an even more challenging route to design an algorithm. Include more distance, obstacles to navigate, doors, stairs, etc.

## Know the Standards

*1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. As students develop programs, they should continuously test those programs to see that they do what is expected and fix (debug) any errors. Students should also be able to successfully debug simple errors in programs created by others.*

# Universal Design

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Articulate diverse perspectives related to improving computer technology tools.

### Instructional Activities

- small-group activity
- whole-class discussion

### CSTA Standards

**Concept:** Impacts of Computing

#### Subconcepts and Standards:

- **Culture: 1B-IC-19** Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users. (P1.2)
- **Social Interactions: 1B-IC-20** Seek diverse perspectives for the purpose of improving computational artifacts. (P1.1)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**accessibility**

**Universal Design**

**usability**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**equitable**

**iterative**

## Materials and Equipment

---

### Collect or prepare the following:

- internet access or resources about Universal Design concepts
- variety of materials to design a Universal Design prototype (optional extension)

## Advance Preparation

---

See the Online Resources Guide for links to the recommended resources.

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

## THE CORE LESSON

### 1. Focus attention on today's topic.

---

How can something created for a particular group be useful for other groups? Ask students if they have ever gone through an automatic door or seen a ramp next to steps. Share with students that many things used in buildings are made for a particular group in mind to be more accessible for that group. However, other groups also find the design useful. Automatic doors and ramps are examples of Universal Design. Universal Design makes a design concept more accessible for the intended group and other individuals.

### 2. Facilitate the activity.

---

Unplugged  
Activity



1. Have students read about a variety of Universal Design ideas and the main purpose or benefit for each design.
2. Place students in teams of four or five. Explain that each team will come up with a plan for a more accessible design. The design should have one group in mind for accessibility and must also be usable for other groups. Team members will choose which group they are designing for. Optionally, allow students to create for a non-human group (for example, dogs).
3. Once teams have come up with the accessible design and the group the design is being created for, tell them to illustrate and label their design idea.
4. After a team discussion, each team member should write a few sentences explaining their team's Universal Design idea, what group it most benefits in terms of accessibility, and how other groups can also use the design.
5. Have students showcase their illustrations and write-ups around the room.

### 3. Check for understanding.

---

Unplugged  
Activity



Have students do a gallery walk and review all the teams' labeled illustrations. Before looking at the write-ups, students should try to decide what group will most benefit from the design by looking only at the illustration. Then, they should think of other groups that can use the design plan. Next, they will read the write-ups to compare their answers with those from the original team.

**EXTEND**—Extend student learning by asking students to use one of the Universal Design ideas and make a prototype for it. Provide a variety of materials, or have students make a list of materials they would like to use for their prototype.

## Know the Standards

*1B-IC-19 Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.* The development and modification of computing technology are driven by people's needs and wants and can affect groups differently. Anticipating the needs and wants of diverse end users requires students to purposefully consider potential perspectives of users with different backgrounds, ability levels, points of view, and disabilities. For example, students may consider using both speech and text when they wish to convey information in a game. They may also wish to vary the types of programs they create, knowing that not everyone shares their own tastes.

*1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts.* Computing provides the possibility for collaboration and sharing of ideas and allows the benefit of diverse perspectives. For example, students could seek feedback from other groups in their class or students at another grade level. Or, with guidance from their teacher, they could use video conferencing tools or other online collaborative spaces, such as blogs, wikis, forums, or website comments, to gather feedback from individuals and groups about programming projects.

# Respecting Copyrights

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objectives

- ✓ Explain the concept of copyright.
- ✓ Summarize the responsibilities of media use.

### Instructional Activities

- student reading
- whole-class discussion

### CSTA Standards

#### Concepts:

- Algorithms and Programming
- Impacts of Computing

#### Subconcepts and Standards:

- **Program Development: 1B-AP-14** Observe intellectual property rights and give appropriate attribution when creating or remixing programs. (P7.3)
- **Safety, Law, and Ethics: 1B-IC-21** Use public domain or creative commons media, and refrain from copying or using material created by others without permission. (P7.3)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**copyright**

**credit**

**permission**



**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**attribution**

**intellectual property**

**public domain**

## Instructional Resource

Student Reader



Ch. 8

**Student Reader, Chapter 8**  
Respecting Copyrights

## Materials and Equipment

**Collect or prepare the following:**

- classroom computer, tablet, or other computing device as needed for discussion

## THE CORE LESSON

### 1. Focus attention on today's topic.

Why is it important to use legal materials when developing a computer program? Creators of any type of media own the rights to their work, and others may not use their work unless they have permission to do so. Copyright laws protect creative works by default, until they become part of public domain. Creative Commons allows works to be used for free, provided attribution is given. There are a number of types of Creative Commons licenses.

### 2. Read and discuss "Respecting Copyrights."

Student Reader



Ch. 8

Read together or have students read independently "Respecting Copyrights," in the Student Reader. The selection introduces the importance of using only legal materials when creating a computer program.

#### Guided Reading Supports

When reading aloud together as a class, always prompt students to follow along. Pause for discussion. Include suggested questions and prompts:

**Page 29**

Ask students to define *copyright*. Confirm that it is the right given to a creator to print, publish, or perform their work or authorize others to do the same. Ask students why they think copyrights are part of legal matters.

**Page 30**

Ask students to give an example of getting permission to do something, such as borrow a bike or go to a friend's house. What would happen if they didn't get permission prior to doing so?

Ask students to explain why not all images or videos on the internet are open to being used in their program. Can they define *public domain*? Confirm that it is the state of something not being subject to copyright and therefore being available to the public. Ask them if permission should still be given when using a public domain item.

### 3. Check for understanding.

Unplugged  
Activity



Ask each group to name something that falls into each of the following categories.

- Copyright
- Public domain
- Creative Commons

## Know the Science

Online Resources



Intellectual property rights can vary by country, but copyright laws give the creator of a work a set of rights. It prevents others from copying the work and using it in ways that the creator may not like. Everything that's published online is protected by copyright by default. That includes images, writing, music, podcasts, videos, and so on. The creator can allow people to use it if they agree to certain terms, called licenses.

Students should consider common licenses that place limitations or restrictions on the use of others' work, such as images and music downloaded from the internet. Creative Commons is an international organization that created a system in which every creator decides how to license their own work, but attribution must always be given. It saves both creator and developer from asking every time they wish to use the work.

After a certain amount of time, the work is considered part of the public domain. This means that the work is owned by the public and permission is not needed to use it. See the Online Resources Guide for links to useful additional classroom resources related to copyright and permissions.

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

# Using Material and Giving Credit

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Model the responsible and ethical use of media from outside sources.

### Instructional Activities

- whole-class discussion
- small-group activity

### CSTA Standards

#### Concepts:

- Algorithms and Programming
- Impacts of Computing

#### Subconcepts and Standards:

- **Program Development: 1B-AP-14** Observe intellectual property rights and give appropriate attribution when creating or remixing programs. (P7.3)
- **Safety, Law, and Ethics: 1B-IC-21** Use public domain or creative commons media, and refrain from copying or using material created by others without permission. (P7.3)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

## Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

copyright

Creative Commons

credit

permission

public domain

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**attribution**

**ethical**

**intellectual property**

## Materials and Equipment

**Collect or prepare the following:**

- scissors
- paper clips

Activity Page



AP 8.2

### Activity Page 8.2

Copyright Dilemma (AP 8.2)

## Advance Preparation

- Make sufficient copies of AP 8.2 for groups of five.
- Cut the Activity Page as marked, and attach the five pieces together with a paper clip.

## THE CORE LESSON

### 1. Focus attention on today's topic.

Do you only have to give credit to information from books or articles? Ask students if they have ever created anything—an art piece, a musical beat, a sports move—only to see others copying it a few days later. Prompt them to discuss the consequences of copying—legally, ethically, and financially. Remind them that creators of any type of media own the rights to their work and that unless they give permission, others may not use their work.

### 2. Facilitate the activity.

Activity Page



AP 8.2

1. Read the following Copyright Dilemma scenario to the class:

*You are part of a student green team that is creating a new poster for your school's big eco-event. The poster needs some graphics that are fun and colorful to draw attention to the upcoming event.*

*One member of your team has found some fantastic and fitting graphics online from a visual designer in another part of the country. They show the rest of your team, and everyone loves the visual look and how well it represents the event.*

*Your team realizes that they need to consider if they can legally use the graphics. They remember that creators of any type of media own the rights to their work and need to give permission before others can use it, unless it is no longer protected by copyright. All team members want to be the most ethical and responsible digital citizens possible, so they discuss the ways they can best handle using the graphics.*

Unplugged  
Activity



2. Divide the class into teams of five to discuss the scenario.
3. Pass out the five responses to each group, anonymously giving one to each group member. When teams are more or less than five, ask two students to pair up to read the response or ask a student to read more than one.
4. In numbered order, group members read their response verbatim to the rest of the group. After all are read, they should discuss how they would proceed if they were the poster design team.
5. Remind students that there is no correct answer to this scenario. The point is to get them to discuss their options and learn more about copyright, permissions, public domain, and Creative Commons.

### 3. Check for understanding.

---

Unplugged  
Activity



Ask each group to share a summary of their discussion. Instruct them to communicate if they come to any conclusions about how they would proceed. Were the group members in agreement about what actions to take? Is there only one correct way to handle the situation?

## Know the Standards

*1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.* Intellectual property rights can vary by country, but copyright laws give the creator of a work a set of rights that prevents others from copying the work and using it in ways that they may not like. Students should identify instances of remixing, when ideas are borrowed and iterated upon, and credit the original creator. Students should also consider common licenses that place limitations or restrictions on the use of computational artifacts, such as images and music downloaded from the internet. At this stage, attribution should be written in the format required by the teacher and should always be included on any programs shared online.

*1B-IC-21 Use public domain or Creative Commons media, and refrain from copying or using material created by others without permission.* Ethical complications arise from the opportunities provided by computing. The ease of sending and receiving copies of media on the internet, such as video, photos, and music, creates the opportunity for unauthorized use, such as online piracy, and disregard of copyrights. Students should consider the licenses on computational artifacts that they wish to use. For example, the license on a downloaded image or audio file may have restrictions that prohibit modification, require attribution, or prohibit use entirely.

# Using Feedback

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objectives

- ✓ Recognize communicating with peers about proposed solutions as an important part of the design process.
- ✓ Value that shared ideas can lead to improved designs.

### Instructional Activities

- student reading
- whole-class discussion

### CSTA Standards

**Concept:** Algorithms and Programming

**Subconcept and Standard: Program Development: 1B-AP-17** Describe choices made during program development using code comments, presentations, and demonstrations. (P7.2)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**feedback**

**reflect**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**critique**

**demonstrate**

**modification**

## Instructional Resource

Student Reader



Ch. 9

### Student Reader, Chapter 9 Using Feedback

## Materials and Equipment

### Collect or prepare the following:

- classroom computer, tablet, or other computing device as needed for discussion

## THE CORE LESSON

### 1. Focus attention on today's topic.

Why is it important to get feedback on proposed solutions? Getting feedback from others is an important part of the design process and can help improve a computer program. The larger and more diverse the feedback is, the more valuable it will be to the development process.

### 2. Read and discuss “Using Feedback.”

Student Reader



Ch. 9

Read together or have students read independently “Using Feedback,” in the Student Reader. The selection introduces the importance of getting feedback on proposed solutions.

### Guided Reading Supports

When reading aloud together as a class, always prompt students to follow along. Pause for discussion. Include suggested questions and prompts:

#### Page 32

With whom does Mrs. Hale’s team share their program? Ask students if they think showing the program to their potential customers to get their feedback is valuable and why.

#### Page 33

Ask students to define *critique*. Confirm that it means a detailed analysis or assessment of something. Invite students to share any critiques they have ever encountered. How were the critiques helpful?

#### Page 34

Ask students to define *reflection*. Confirm that it means to give serious thought or consideration. Ask them if they have ever received feedback that wasn’t what they wanted to hear. After reflecting on the feedback, were they able to see how it could help them improve?

#### Page 35

If time permits, pair students and prompt them to choose one image on the page and discuss how what is happening in the image is important to creating a successful computer program.

### 3. Check for understanding.

---

Unplugged  
Activity



Why is it important to get feedback on proposed solutions? Ask students to share their thoughts on why getting feedback is important to any program development project. Prompt them to share a situation in which they received feedback on an art project, a math test, a dance performance, a sports skill, or something else. Ask if they were able to integrate that feedback to make improvements.

## Know the Science

People share their computer programs to help others understand and use their programs. They explain design choices so others can get a better understanding of the work—the purpose, the function, and the process. Explaining the design process can be done using presentation software, storyboards, annotated images, videos, or journal entries. Getting feedback from others is an important part of the design process and can help improve a computer program.



# Program Designs and Functions

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objectives

- ✓ Design, test, and revise a computer project.
- ✓ Present or demonstrate a design, and reflect on choices made to achieve desired goals and what was learned from the experience.
- ✓ Critique the projects of other students.

### Instructional Activities

- whole-class discussion
- small-group activity
- whole-class presentations

### CSTA Standards

**Concept:** Algorithms and Programming

**Subconcept and Standard: Program Development: 1B-AP-17** Describe choices made during program development using code comments, presentations, and demonstrations. (P7.2)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**critique      feedback      present      reflection      sharing**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**constructive      demonstrate      modification      presentation**

## Materials and Equipment

---

### Collect or prepare the following:

- large whiteboard or anchor chart
- tablets and internet access to the recommended programming app (ScratchJr)
- printable motion blocks (from ScratchJr Fun) (4 to 5 sets, 1 set per group)  
(See the Online Resources Guide for links to these items. [www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources))
- scissors
- small manipulative or play figurine (1 per group)
- drawing or writing paper
- pencils with erasers

## Advance Preparation

---

- ScratchJr is an introductory programming language that enables young children to create their own interactive stories and games. ScratchJr is available as a free app. For full information, please visit <https://www.scratchjr.org/about/info>.
- Download the free printable ScratchJr motion blocks (you may have to sign up to download). Print one copy each of the motion blocks on pages 3–8 for classroom display.
- Print four copies of page 43 scaled at 60 percent for each small group.
- Set up tablets with internet access and the ScratchJr app opened.
- Set up “design” stations with motion block pages, scissors, manipulative, drawing or writing paper, and pencils.
- On the whiteboard or anchor chart, write the heading “Steps of the Design Process” followed by these bulleted steps: Ask, Imagine, Plan, Create, Test, Improve, Share.

## THE CORE LESSON

### 1. Focus attention on the lesson question.

---

How can a program design and function be clearly explained to others? Ask students about the process of communicating their algorithm to the teammate in Lesson 4. Did they feel they communicated the instructions well and got their teammate to the right place, or did they require more explanation? What feedback did those who were being led have? Encourage students to share their thoughts on how a design can be best shared with others. Ask them why getting feedback is an important part of the design process and can help improve a design.

Tell students that today they’re going to act as programmers and design a program! Then, they will share that program with classmates to get their feedback. Walk students through the steps of the design process written on the board (Ask, Imagine, Plan, Create, Test, Improve, Share), and remind them what happens at each step in the process.

## 2. Facilitate the activity.

---

### Online Resources



### Device-Based Activity



1. Demonstrate the motion block area of the ScratchJr app, beginning with the video tutorial. When computers are not available, use the printed pages to explain how to code using the ScratchJr motion blocks.
2. Tell students that these are blocks of code, and explain that the symbol on each block represents a particular instruction for a computer. **Ask:** What instructions do you think each symbol represents? (*move right, move left, move forward, move back, turn right, turn left*) Demonstrate how students can represent the number of times each motion should occur by writing a number in the space at the bottom of a block. For example, a “4” under the move right symbol means “move right” four times.
3. **Ask:** How do you think I can use the blocks together to give instructions? Allow several students to share their ideas. Then demonstrate how the blocks can be put next to each other in a sequence to create instructions. For example, you might show how placing a “move left” block with a “3” next to the “turn left” with a “1” means “move left three times and then turn left once.” Give additional examples until students are comfortable with using the motion blocks.
4. Place students in groups of four or five, and assign each group to a tablet or table. Ask students to work together to practice using the ScratchJr app or cut out the motion blocks from the pages at the station.  
  
**SUPPORT**—Depending on time and your students, you may wish to precut the blocks before class.
5. Next, tell students that they are ready to begin designing their program. Explain that the purpose of the program is to tell the computer how to move the character in a large square and that they will use the motion blocks to write the instructions.
6. Direct students’ attention to Step 1 (**Ask**) of the design process on the board. This step involves the question “What is the purpose of the program?” Since students already know the purpose of the program, tell them they can move on to the next step. Have students work together to **imagine** ideas about how they could move the object. Next, have them **plan** out the steps they want to take. As long as their character completes the large square movement, they can add any other moves, loops, and designs the group wishes. They should record any ideas they have on the drawing or writing paper.
7. Next, ask students to use the motion blocks to **create** their program. They should indicate the number of times a motion should repeat by writing the number in pencil on the block and then place the blocks in order.
8. They should **test** their program by having one student read out the steps in the sequence while another student moves the object along the squares of the maze, following the directions. If students encounter any problems, they should revise the program to **improve** it.

**EXTEND**—If you want to include loops in the activity, you can also print out page 39 and print out four copies of page 49 scaled at 60 percent for each small group, cutting out the loop tile (middle tile of top row). Explain to students that any steps in the loop should be placed in order below the loop block.

### 3. Check for understanding.

---

Unplugged  
Activity



Ask for volunteers to **share** their group’s code with the class and explain the steps they took to create it and test it. **Ask:** What part of the design process do you like best? What part did you like least? Discuss the importance of each stage of the design process. Encouraging at least two students to provide constructive feedback to the presenting group. Ask them to give constructive feedback, such as design elements they thought were unique or useful and elements that need more explanation.

## Know the Standards

*1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.* People communicate about their code to help others understand and use their programs. Another purpose of communicating one’s design choices is to show an understanding of one’s work. These explanations could manifest themselves as in-line code comments for collaborators and assessors or as part of a summative presentation, such as a code walk-through or coding journal.

# Computer Technology and Culture

**Driving Question:** What kinds of computer programs can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Describe how technology has evolved to improve lives.

### Instructional Activities

- student reading
- whole-class discussion

### CSTA Standards

**Concept:** Impacts of Computing

**Subconcept and Standard: Culture: 1B-IC-18** Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices. (P7.1)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion and in written responses. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**culture**                      **impact**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**influence**                      **society**

## Instructional Resource

Student Reader



Ch. 10

**Student Reader, Chapter 10**  
Computer Technology and Culture

## Materials and Equipment

**Collect or prepare the following:**

- classroom computer, tablet, or other computing device as needed for discussion

## THE CORE LESSON

### 1. Focus attention on today's topic.

Prompt students to discuss how people did certain tasks prior to computing technology and what benefits computing technology provides society (e.g., robotics, GPS, internet, social media, smartphones, self-driving cars, etc.).

### 2. Read and discuss "Computer Technology and Culture."

Student Reader



Ch. 10

Read together or have students read independently "Computer Technology and Culture," in the Student Reader. The selection introduces the idea that computing devices both affect and are affected by culture.

#### Guided Reading Supports

When reading aloud together as a class, always prompt students to follow along. Pause for discussion. Include suggested questions and prompts:

**Page 36**

Invite students to define *impact*. Confirm that it means to have a strong effect on something or someone. Ask students to describe the impact that the reading log program could have on the first-grade students, the fifth-grade students, and the teachers.

**Page 37**

Prompt students to name ways that technology has changed the school culture since the time their parents went to school. What impact has computer technology had on the teachers in the story? Encourage them to discuss things like online school, saving paper, saving time, faster communication, etc.

**Page 38**

Ask students to define *influence*. Confirm that it is the capacity to have an effect on someone or something. Ask students to share a time they were influenced by someone or when they influenced someone.

**Page 39**

Invite interested students to share if they think they would be interested in creating computer programs as a job when they grow up and finish school. Ask volunteers to elaborate on what kinds of computer programs interest them the most and what they might like to create.

### 3. Check for understanding.

---

Unplugged  
Activity



In a round-robin discussion, prompt students to choose one device and share how the lives of teachers and students were different before and after that technology's introduction into society.

- Tablet
- Video conference screen
- Computer
- Projector
- Smartphone camera
- Smartphone app

For example, a student response could include that in the past, if students wanted to read about a topic, they needed access to a library to find a book about it. Today, students can view and read information on the internet about a topic, or they can download e-books about it directly to a device. Such information may be available in more than one language and could be read to a student, allowing for great accessibility.

## Know the Science

Computing technologies have changed the way people live and work. New computing technologies are created and existing technologies are modified for many reasons, including to increase their benefits, decrease their risks, and meet societal needs. There are positive and negative impacts to the world due to these technologies. Current technology news content might include topics such as robotics, wireless internet, mobile computing devices, GPS devices, wearable computing, self-driving cars, and how social media has influenced social and political changes.

# Becoming a Computer Programmer

**Driving Question:** What kinds of problems can I develop?

**Overarching Unit Objective:** Write, evaluate, and debug code to execute grade-appropriate tasks.

## AT A GLANCE

### Learning Objective

- ✓ Explore online coding and programming options for students.

### Instructional Activities

- supervised individual computer operation
- group reflection

### CSTA Standards

**Concept:** Algorithms and Programming

**Subconcept and Standard: Program Development: 1B-AP-17** Describe choices made during program development using code comments, presentations, and demonstrations. (P7.2)

For detailed information about the CSTA Standards, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

### Core Vocabulary and Language of Instruction

A Glossary at the end of this Teacher Guide lists definitions for Core Vocabulary and selected Language of Instruction.

**Core Vocabulary** terms are those that students should learn to use accurately in discussion. During instruction, expose students repeatedly to these terms but not through isolated drill or memorization.

**coding**      **programming**

**Language of Instruction** consists of additional terms that you should use when talking about concepts in the lesson. Students benefit from your modeling the use of these words without the expectation that students will use or explain the words themselves.

**application**      **open source**      **programming language**      **visual interface**      **website**



## Materials and Equipment

### Collect or prepare the following:

- computer and internet access to the recommended programming app (Scratch) for each student

(See the online Resources Guide for links to these items. [www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources))

- means to project the teacher’s screen for demonstration and student guidance
- Coding Cards for ‘Make a Chase Game’ and ‘Pong Game’

## Advance Preparation

- Watch the ‘Getting Started’ tutorial. Practice using the coding application in advance of introducing students to it. Scratch is a coding language with a visual interface that enables youth to program stories, animations, and games. Scratch is available as a free resource. For full information, including tutorials, please visit <https://www.scratch.mit.edu/about>.
- Use your school’s protocol for logging in and making sites available for students.
- Set aside multiple class sessions for students to practice with the interface. The more interactive sessions you can provide students with, the greater their skills will be reinforced.

Set up a Scratch teacher account. Use this to create student accounts.

Download and review the Educator Guide for ‘Make a Chase Game’ and ‘Pong Game’. Please note that this lesson follows guidance within the Educator Guide.

Make sufficient copies of Coding Cards for ‘Make a Chase Game’ and ‘Pong Game’ for each student or pair of students. Found on [scratch.mit.edu/ideas](http://scratch.mit.edu/ideas)

Watch and practice tutorials for ‘Make a Chase Game’ and ‘Pong Game’.

## THE CORE LESSON

### 1. Focus attention on the lesson topic.

Device-Based  
Activity



Ask students about video games they have played before. Have students describe the games: What was the purpose of the game? How did you control the game? Tell students that today, they will be working on creating their own video game! They will be using a computer program called Scratch, act as real video game programmers and program their very own video game. Show students the tutorials for ‘Make a Chase Game’ and ‘Pong Game’.

### 2. Facilitate the activity.

After viewing the tutorials, have students decide which type of game they would like to program – a pong game or a chase game. Optionally, choose one program for the students to focus on. Spark interest by asking students to brainstorm a theme for their game. Some examples may be a cat and mouse chase game, or a frog and lily pad pong game.

Using the coding cards as a guide, demonstrate the first steps of the tutorials for students. Have students follow along with these first steps. Ask students to share any ideas, problems, and solutions they encounter during these steps.

Provide students with at least 45 minutes to explore and practice creating their program. Recommend that they use the coding cards to guide them through their programming. Tell students that they should try out many different combinations of blocks to see the result. They should learn from this practice to make decisions on their final product.

Suggest that they practice and/or make decisions on the following:

- Creation of main characters
- Moving characters
- Different backgrounds
- Keep score
- Show messages
- Add sounds

As students are working through their programming, remind them to revisit the tutorials for 'Make a Chase Game' and 'Pong Game' for additional ideas on how they can enhance their game.

### 3. Check for understanding.

---

#### Online Resources



#### Unplugged Activity



Convene the class for discussion after students have had ample time for programming experience. You may choose to display some student outcomes for discussion. Prompt discussion with students using questions like these:

- What were you able to do using the programming application?
- What did you enjoy most about it?
- Was any of it hard to do?
- What would make that easier?
- What would you like to do next?

After games are completed, pairs students or groups together and have students share their programs with each other. Ask students to explain their goals for their game, the decisions they had to make, any problems and solutions they encountered, and any changes they would make to their game now. Have students reflect on what they knew about programming before creating their game and how their understandings have changed.

**EXTEND**—Have students return to their game and implement any changes now that they have seen other games and thought about changes they would like to make.

**EXTEND**—Have students explore different games listed on the Scratch app at <https://scratch.mit.edu/explore/projects/all> or additional skills from <https://scratch.mit.edu/ideas>. Ask students to think about any changes they would like to make, what they are curious about, or what they want to explore further. Encourage them to make changes to their games.

# Teacher Resources

## **Activity Pages**

- Mystery Messages (AP 2.2) **102**
- Instructions for Actions (AP 6.2) **104**
- Copyright Dilemma (AP 8.2) **106**

Name \_\_\_\_\_

Date \_\_\_\_\_

## Activity Page 2.2

Use with Lesson 2, Part 2

**Mystery Messages**

Copy the contents of page 101 and 102. Cut apart message pieces for reassembly during the student activity. Make sure to remove the answers for each trial from the message sets. Keep the pieces for the different trials separated.

Trial A (*CAN YOU SOLVE A MYSTERY CODE MESSAGE WITH NO CLUES BUT LETTERS?!*)

CA	N_	YO	U_	SO
LV	E_	A_	MY	ST
ER	Y_	CO	DE	_M
ES	SA	GE	_W	IT
H_	NO	_C	LU	ES
_B	UT	_L	ET	TE
RS	?			

Trial B (*THE MYSTERY MESSAGE = PACKETS!*)

1. T	2. H	3. E	4.	5. M
6. Y	7. S	8. T	9. E	10. R
11. Y	12.	13. M	14. E	15. S
16. S	17. A	18. G	19. E	20.
21. =	22.	23. P	24. A	25. C
26. K	27. E	28. T	29. S	30. !

Name \_\_\_\_\_

Date \_\_\_\_\_

Activity Page 2.2 (continued)

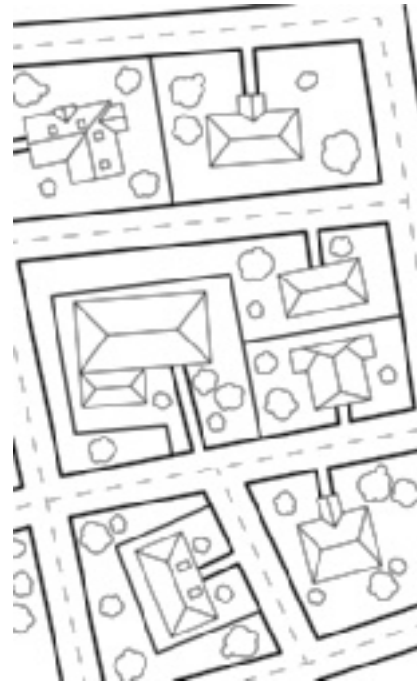
Use with Lesson 2, Part 2

Trial C (INTERNET; NETWORK; PACKETS; COMPUTER)

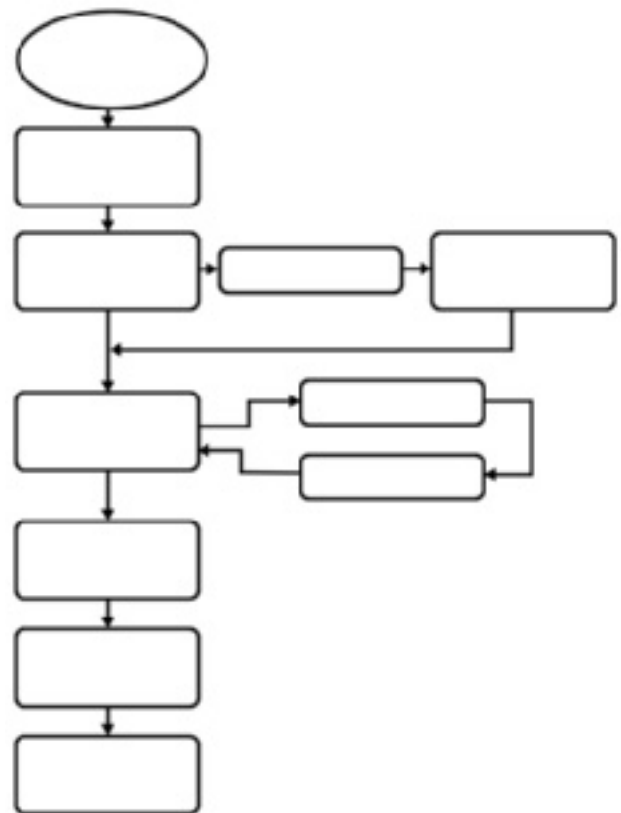
*1. I	*2. N	*3. T	*4. E	*5. R
*6. N	*7. E	*8. T	x1. N	x2. E
x3. T	x4. W	x5. O	x6. R	x7. K
+1. P	+2. A	+3. C	+4. K	+5. E
+6. T	+7. S	>1. C	>2. O	>3. M
>4. P	>5. U	>6. T	>7. E	>8. R

### Instructions for Actions

#### Program an Alien to Ride a Bike Around the Block



Suppose you needed to give detailed instructions to your tentacled alien friend to help it ride a bike around a neighborhood block. Use only the commands below to make a sequence diagram or mind map. Refer to the algorithm plan from your Student Reader on page 25 as an example. Use one shape/color for events, a different shape/color for instructional steps, another color/shape for conditionals, and arrows to indicate sequences and loops. You can reuse steps, and there are some that you may not need to use at all. The alien is new to bike riding, so better to keep it on the sidewalk.



Name \_\_\_\_\_

Date \_\_\_\_\_

Activity Page 6.2 (*continued*)

Use with Lesson 6, Part 2

Stop and park the bike.

If corner is not reached

If driveway is not reached

Rest body weight on the seat.

Turn left and release the break.

Start at the end of the driveway.

Turn right and release the break.

Gently squeeze the break to slow down.

Curl up tentacles to keep them out of the way.

Push the left pedal down with the left tentacle.

Push the right pedal down with the right tentacle.

Straddle the bike seat with eyes toward the handlebars.

Place one tentacle on the ground on each side of the bike.

Grab each handlebar grip and hold each with one tentacle.

Push back on the ground with the left tentacle to move the bike forward.

Push back on the ground with the right tentacle to move the bike forward.

Name \_\_\_\_\_

Date \_\_\_\_\_

Activity Page 8.2

Use with Lesson 8, Part 2

## Copyright Dilemma

Cut the responses apart and fold them so the text is not visible. Each group member should draw one response to use as a prompt during discussion.

-----

This artwork is just what we need. No one will know where we got it, and whoever originally made it won't know we are using it.

-----

Let's just use these designs as a guide and redraw them. We won't download the actual images but make some just like them.

-----

Let's do another search for graphics that are open source or public domain and build our design around what is available to us without permission.

-----

Let's contact the maker of the material and see if we can get permission to use it.

-----

Let's make our own graphics in a style like this, but change our designs enough that they won't be confused with the work that inspired us.



## Glossary

**Teal words and phrases** are Core Vocabulary terms for the unit. **Bold-faced words and phrases** are additional vocabulary terms related to the unit that you should model for students during instruction and that are often used within the Student Reader, and these latter terms do not have specific page numbers listed. Vocabulary words are not intended for use in isolated drill or memorization.

---

### A

**abstraction, n.** in computer science, keeping the core parts of a program while removing extra details

**accessibility, n.** the degree to which a program can be used by the greatest number of people with the widest range of differences

**algorithm, n.** steps or rules to follow to complete a specific task or to solve a problem

**attribution, n.** credit that is given to someone; explanation of the cause of something

---

### B

**branching, adj.** splitting or dividing from the main part of something

---

### C

**capacity, n.** the ability to contain or perform

**cause-effect, n.** a relationship in which one factor, the cause, produces the other, the effect

**classification, n.** the process of arranging things into groups based on shared characteristics

**code, n.** a system of symbols that tell meaning or give instruction

**coding, n.** the practice of writing instructions for computer programs

**collaborate, v.** to work together

**collaboration, n.** working together as a team

**collection, n.** a group of things that are selected because of some way that they are related

**compare, v.** to note the similarities of two things

**comparison, n.** an observation of shared characteristics

**component, n.** a part of a greater whole

**condition, n.** an instruction for a computer to execute certain actions if certain criteria are met

**constructive, adj.** helpful toward an improvement

**control, v.** to command or manage

**copyright, n.** the protection of a person's or company's creative ideas from use by others without permission

**Creative Commons, n.** an organization that documents licenses that make creative work available for others to use and share

**credit, n.** recognition of the person who originally came up with an idea or made a creative work

**criteria, n.** the conditions that a solution must meet to be judged as successful

**critique, n.** a detailed analysis or assessment of something

**culture, n.** the traditions, beliefs, and behaviors of a group of people

**cybercrime, n.** theft or misuse of someone else's digital data

**cybersecurity, n.** the protection of computer systems, networks, and data from theft or damage from cybercrimes.

**cyberspace, n.** the concept of a place where all the world's interconnected digital data are stored and moving about

---

### D

**data, n.** details of information collected by observation or measurement

**debug, v.** to identify a problem in a system, determine its source, and correct it

**decision tree, n.** a map of possible outcomes based on decisions

**decomposition, n.** the process of breaking something into parts

**demonstrate, v.** to show how something works

**design, n.** a plan for solving a problem that shows what the solution is intended to be like before it is produced

---

### E

**equitable, adj.** fair for all

**ethical, adj.** correct and in accordance with rules and shared values

**event, n.** an action or occurrence that can be identified by a computer program, such as a keystroke or a program opening up

**execute, v.** to carry out or perform

**execution, n.** the process of carrying out a planned action

---

## F

**feedback, n.** reactions to a product or a performance used as a basis for improvement

---

## H

**hacking, v.** breaking into a secured computer system without permission

---

## I

**impact, n.** effect

**influence, v.** to affect an outcome

**information, n.** collected knowledge

**input/output, n.** signals or information sent to and from a computer

**intellectual property, n.** original ideas that belong to the person who came up with them

**interaction, n.** the effects of factors on one another within a system

**internet, n.** the network of connected computers all over the world

**iterative, adj.** describing a process that involves repeated review, testing, and adjustment to improve a product or program

---

## L

**loop, n.** a sequence of instruction which continually repeats until a certain condition is reached

---

## M

**manipulate, v.** to manage or influence so a certain result is achieved

**modification, n.** a change or update to a design

**modify, v.** to change or update

---

## N

**network, n.** a group of computer devices that communicate with each other

---

## P

**packets, n.** small, contained collections

**pattern, n.** visual details or events that occur in a regular or predictable way

**permission, n.** an official statement allowing someone to do something

**perspective, n.** a persons' particular point of view or way of thinking, based on their own experiences

**plan, v.** a list of steps to achieve a desired outcome

**planning, v.** determining the steps and requirements of a task, design, or solution

**predict, v.** to say what will happen before it happens

**prediction, n.** a statement about what a person thinks will happen in the future

**present, v.** to display information to others and tell them about it

**presentation, n.** a delivery of information to other people by speaking and showing them visual aides

**program development, n.** the process of creating new programs for computers and electronic devices

**programming, n.** the practice of creating commands for computers

**protection, n.** the act of keeping something safe and secure

**protocol, n.** a system of rules for transmitting computer data

**public domain, n.** the status of materials that are free from copyright protection and be republished by anyone

---

## R

**refine, v.** to improve

**reflect, v.** to think back on something that has happened

**reflection, n.** thought or consideration about a past event or effort

**remix, v.** to create a different version of something by adding, removing, or changing pieces of it

**respect, n.** regard for the rights of others and the way they prefer to be treated

**retrieve, v.** to find and bring back

**role, n.** a part a person plays in a given situation; a job

---

## S

**sequence, n.** the order in which steps occur

**sharing, v.** using or experiencing something with others

**society, n.** members of a community and their actions as a group

**solution, n.** a design that meets a need or solves a problem

**storage, n.** the practice of keeping information temporarily or permanently

**system, n.** a set of parts that work together to complete a task

---

## T

**teamwork, n.** working together as a group

**testing, v.** a process of trying out a system to determine if it works as intended

**transport, v.** to move from one location to another

**troubleshoot, v.** to identify and correct the problem in a system

---

## U

**Universal Design, n.** an approach to designing solutions that prioritizes making the solution as accessible as possible for people regardless of differences

**usability, n.** the degree to which a program is easy to use

---

## V

**value, n.** a numerical amount; the worth or importance of something

**variable, n.** an element that can change based on conditions

**visualization, n.** the formation of a mental picture that helps with the understanding of abstract information

---

## Y

**yes/no questions, n.** questions that can only be answered either *yes* or a *no*

## Internet Safety

---

Though online resources present many rich opportunities for student learning, unsupervised online activity for children is not advised. The U.S. Department of Justice provides the following guidelines, Keeping Children Safe Online:

- **Discuss internet safety and develop an online safety plan** with children before they engage in online activity. Establish clear guidelines, teach children to spot red flags, and encourage children to have open communication with you.
- **Supervise young children’s use of the internet**, including periodically checking their profiles and posts. Keep electronic devices in open, common areas of the home, and consider setting time limits for their use.
- **Review games, apps, and social media sites** before they are downloaded or used by children. Pay particular attention to apps and sites that feature end-to-end encryption, direct messaging, video chats, file uploads, and user anonymity, which are frequently relied upon by online child predators.
- **Adjust privacy settings and use parental controls** for online games, apps, social media sites, and electronic devices.
- **Tell children to avoid sharing personal information, photos, and videos online** in public forums or with people they do not know in real life. Explain to your children that images posted online will be permanently on the internet.
- **Teach children about body safety and boundaries**, including the importance of saying “no” to inappropriate requests both in the physical world and the virtual world.
- **Be alert to potential signs of abuse**, including changes in children’s use of electronic devices, attempts to conceal online activity, withdrawn behavior, angry outbursts, anxiety, and depression.
- **Encourage children to tell a parent, guardian, or other trusted adult** if anyone asks them to engage in sexual activity or other inappropriate behavior.

Copy and distribute the Student Safety Contract, found on the next page. Prior to the start of the first lesson, do a read-along, and have students agree to the expectations for when they engage in computer and online activities.

### Online Resources



For additional support for safety in the computer science and online instruction, follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

# Student Online Safety Contract

---

Dear Parent or Guardian,

During computer science class, we want to create and maintain a safe classroom. With this in mind, we are making sure students are aware of the expectations for their behavior while engaged in computer science activities. We are asking you to review the safety rules below with your student and sign this contract. If you have any questions, please feel free to contact me.

For important safety information about children, computers, and the internet, consider resources at these sites:

<https://protectyoungeyes.com/>

<https://sharedhope.org/>

<https://www.justice.gov/coronavirus/keeping-children-safe-online>

\_\_\_\_\_ /\_\_\_\_\_/\_\_\_\_\_/

Teacher signature and date

\_\_\_\_\_ /\_\_\_\_\_/\_\_\_\_\_/

Parent or guardian signature and date

-----  
When doing online activities, I will do the following:

- Only do online activities with the supervision of an adult.
- Only visit websites and use apps that I am guided to by my teacher, parent, or trusted adult guardian.
- Never use my real name or reveal personal information if I communicate with others online.
- Tell a trusted adult right away if anyone online asks questions about my name, where I live, or where I go to school.
- Be careful around electronic devices and only plug them in or unplug them when an adult is supervising.

I understand and agree to the safety rules in this contract.

\_\_\_\_\_ /\_\_\_\_\_/\_\_\_\_\_/

Student signature and date

\_\_\_\_\_

Print name

## Strategies for Acquiring Materials

The materials used in the Core Knowledge Computer Science program are readily available and can be acquired through both retail and online stores. Some of the materials will be reusable and are meant to be used repeatedly. This includes items such as plastic cups that can be safely used again. Often these materials are durable and will last for more than one activity or even one school year. Other materials are classified as consumable and are not able to be used more than once.

### Online Resources



The Material Supply List for this unit's activities can be found online. Follow the links in the Online Resources Guide for this unit:

[www.coreknowledge.org/cksci-online-resources](http://www.coreknowledge.org/cksci-online-resources)

## Ways to Engage with Your Community

The total cost of materials and technology can add up for an entire unit, even when the materials required for activities and demonstrations have been selected to be individually affordable. And the time needed to acquire the materials adds up, too. Reaching out to your community to help support STEM education is a great way to engage parents, guardians, and others with the teaching of science and computer science, as well as to reduce the cost and time of collecting the materials. With that in mind, the materials list can be distributed or used as a reference for the materials teachers will need to acquire to teach the unit.

### Consider some of the following as methods for acquiring the science materials:

- **School Supply Drive**—If your school has a supply drive at any point in the year, consider distributing materials lists as wish lists for the science department.
- **Open Houses**—Have materials lists available during open houses. Consider having teams of volunteers perform an activity to show attendees how the materials will be used throughout the year.
- **Parent-Teacher Organizations**—Reach out to the local PTO for assistance with acquiring materials.
- **Science Fair Drive**—Consider adding a table to your science fair as part of a science materials drive for future units.
- **College or University Service Project**—Ask service organizations affiliated with your local higher education institutions to sponsor your program by providing materials.
- **Local Businesses**—Some businesses have discounts for teachers to purchase school supplies. Others may want to advertise as sponsors for your school/programs. Usually you will be asked for verifiable proof that you are a teacher and/or for examples of how their sponsorship will benefit students.

Remember: If your school is public, it will be tax exempt, so make sure to have a Tax Identification Number (TIN) when purchasing materials. If your school is private, you may need proof of 501(c)(3) status to gain tax exemption. Check with your school for any required documentation.



Core Knowledge®

CKSci™

Core Knowledge SCIENCE™

**Editorial Director**

Daniel H. Franck

## Subject Matter Expert

Sarah J. Huibregtse, PhD

## Illustrations and Photo Credits

Aleksei Gorodenkov / Alamy Stock Photo: Cover D, i, iii

Roman Lacheev / Alamy Stock Photo: Cover A

Within this publication, the Core Knowledge Foundation has provided hyperlinks to independently owned and operated sites whose content we have determined to be of possible interest to you. At the time of publication, all links were valid and operational, and the content accessed by the links provided additional information that supported the Core Knowledge curricular content and/or lessons. Please note that we do not monitor the links or the content of such sites on an ongoing basis and both may be constantly changing. We have no control over the links, the content, or the policies, information-gathering or otherwise, of such linked sites.

By accessing these third-party sites and the content provided therein, you acknowledge and agree that the Core Knowledge Foundation makes no claims, promises, or guarantees about the accuracy, completeness, or adequacy of the content of such third-party websites and expressly disclaims liability for errors and omissions in either the links themselves or the contents of such sites. If you experience any difficulties when attempting to access one of the linked resources found within these materials, please contact the Core Knowledge Foundation:

**[www.coreknowledge.org/contact-us/](http://www.coreknowledge.org/contact-us/)**

Core Knowledge Foundation

801 E. High St.

Charlottesville, VA 22902



# Core Knowledge Curriculum Series™

## CKSci™ Core Knowledge SCIENCE™

### Designing Computer Programs Core Knowledge Science 5



#### What is the Core Knowledge Sequence?

The *Core Knowledge Sequence* is a detailed guide to specific content and skills to be taught in Grades K–8 in language arts, history, geography, mathematics, science, computer science, and the fine arts. In the domain of computer science, the *Core Knowledge Sequence* outlines topics that build systematically grade by grade to support student learning progressions coherently over time.



#### For which grade levels is this book intended?

In general, the content and presentation are appropriate for students in the upper elementary grades. For teachers and schools following the *Core Knowledge Sequence*, this book is intended for Grade 5 and is part of a series of **Core Knowledge SCIENCE** units of study.

For a complete listing of resources in the **Core Knowledge SCIENCE** series, visit [www.coreknowledge.org](http://www.coreknowledge.org).

# CKSci™ Core Knowledge SCIENCE™

A comprehensive program in science, integrating topics from Earth and Space, Life, and Physical Sciences with concepts specified in the **Core Knowledge Sequence** (content and skill guidelines for Grades K–8).

## Core Knowledge SCIENCE™

units at this level include:

**Investigating Matter**  
**Energy and Matter in Ecosystems**  
**Modeling Earth's Systems**  
**Protecting Earth's Resources**  
**Astronomy: Space Systems**  
**Human Hormones and Reproduction**  
**Designing Computer Programs**

[www.coreknowledge.org](http://www.coreknowledge.org)

Core Knowledge Curriculum Series™