

# COMPUTER SCIENCE FOR KIDS

A STORYTELLING APPROACH



JEN LOOPER

WILEY



# Computer Science for Kids

---



# Computer Science for Kids

## A Storytelling Approach

---

Jen Looper

WILEY

Copyright © 2023 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.  
Published simultaneously in Canada and the United Kingdom.

ISBN: 978-1-119-91251-4  
ISBN: 978-1-119-91253-8 (ebk)  
ISBN: 978-1-119-91252-1 (ebk)

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at [www.copyright.com](http://www.copyright.com). Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at [www.wiley.com/go/permission](http://www.wiley.com/go/permission).

**Trademarks:** WILEY and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

**Limit of Liability/Disclaimer of Warranty:** While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Further, readers should be aware that websites listed in this work may have changed or disappeared between when this work was written and when it is read. Neither the publisher nor authors shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our website at [www.wiley.com](http://www.wiley.com).

**Library of Congress Control Number: 2023933144**

Cover image: © Mirla Montaña  
Cover design: Wiley

*To all the kids who don't see themselves as having a place in  
Computer Science . . . Remember that your story is yet to be told!*





# About the Author

---



**Jen Looper** is the Head of Academic Advocacy at AWS, previously at Microsoft, and a Google Developer Expert with over 22 years' experience as a web and mobile developer, specializing in creating cross-platform mobile and web apps. With a PhD in medieval French literature, Jen is a multilingual multiculturalist with a passion for web technologies and applied machine learning, and for discovering new things every day. Her area of focus is curriculum

development and the application of sound pedagogy to technical topics. A lecturer in computer science at Boston University's Metropolitan College, she is also the founder and CEO of Front-End Foxes, Inc., an international non-profit charity that promotes diversity in front-end developer communities. Visit Jen at [www.jenlooper.com](http://www.jenlooper.com) or connect via Twitter @jenlooper.



# About the Technical Editors

---

**Andrea Russo** is the Field Enablement and Go to Market Lead for the Microsoft Philanthropies Digital Inclusion Program. She has worked for the program for the past six years. When she started as a regional manager, her role was co-funded by Governor Gina Raimondo’s CS4RI initiative through 2018. While Andrea was in that role, Rhode Island became the first state to offer computer science opportunities to every public school in the state and received Harvard’s Innovation in Government Award. Before joining Technology Education and Literacy in Schools, Andrea worked for the National Council of Technological Literacy and taught grades 6–12 in New York City as a STEM teacher. Prior to her teaching career, she worked at the New York Botanical Garden in the Continuing Education department. She holds an M.S. and an M.Ed.

**Chris Noring** is a Senior Cloud Advocate at Microsoft. He’s been in the IT industry for more than 15 years and has worked on multiple tech stacks as well as front- and backend. Chris is a community organizer and co-creator of the Angular conference ngVikings, and is a published author on the JavaScript framework Angular as well as the Go language.

**Justin Leonard** is a middle school educator with Prince George County Public Schools, Maryland. In his current role, he is a Project Lead the Way (PLTW) Gateway instructor. He also teaches graduate-level courses for the Master of Arts in Education in the STEM Education program at the University of Maryland, Baltimore County.



# Acknowledgments

---

I'd like to acknowledge the many readers of drafts of this book, including Mrs. Gentes's eighth-grade computer science students at Wellesley Middle School. Thanks also go to Wellesley High School's Dr. Cohen, always a helpful teacher, who has listened to many of my wild ideas with interest and has given me useful feedback. Thank you to Allegra Hu, who created the prototype of the Guide as a cloaked fox figure. Illustrations are by Mirla Montaña. I'd like to thank my patient editors and hardworking technical reviewers. I'd above all like to thank my family for their encouragement in writing this book.



# Contents

---

Introduction	xv
How to Contact Wiley or the Author	xxi
Preface	1
<b>I The Missing Fireflies</b>	<b>3</b>
1. The Trapper	5
2. The Firefly Refuge	23
3. The Fireflies' Message	35
4. A Home for All	49
<b>II The Glowing Moss</b>	<b>61</b>
5. Goblin's Gold	63
6. Securing the Moss	83
7. The Whisper Network	97
8. A Well-Tested Solution	109
<b>III Geode Quest</b>	<b>121</b>
9. Searching for Geodes	123
10. Cleaning and Categorizing the Collection	137
11. The Stone Library	155
12. Hide or Seek	171
	<b>xiii</b>

<b>IV The Petrified Forest</b>	<b>183</b>
13. The Petrified Forest	185
14. The Butterfly Brigade	201
15. Power of the Weak	215
16. The Written Artifact	227
<b>V Legends of the Field Mice</b>	<b>241</b>
17. The Field Mice	243
18. Lights, Sounds, Action	273
19. Unlocking the Vault	291
20. The Real World	307
Index	317



# Introduction

---

When you think about computer science, what pops into your head? Someone crouched over a computer, typing code all night long? Someone in a room full of computers, messing around with wires? Someone solving long mathematical equations by writing a lot of hard-to-understand computer programs? Does that person look like you?

What if you thought about computer science in a different way—as a way to solve mysteries? Isn't it more fun to think about yourself as an intrepid explorer, out to solve challenging puzzles by writing clever bits of software? Can you see yourself as the hero or heroine of a story, ready to slay the dragons of slow, buggy, hard-to-read software and make the world a little more pleasant by means of great user design and well-functioning programs?

In this book, you'll be given the opportunity to become a helper—someone who is able to solve challenges by writing computer programs. Summoned by a mysterious figure who emerges from a forest and asks for your help, you are invited to participate in a series of tasks to make this forest a little safer and more hospitable for its native inhabitants.

Are you up to the task?

## Computer Science Teachers Association Standards

This book is structured using the Computer Science Teachers Association standards for Grades 6–8, covering ages 11–14. These standards are available at <https://csteachers.org>. In general, they are arranged around five concepts:

- Algorithms & Programming
- Computing Systems
- Data & Analysis
- Impacts of Computing
- Networks & the Internet

The standards help shape the projects that are used to drive home general concepts of computer science. This is not a book about programming languages, however. Rather than diving deep into JavaScript and Python, we focus on the standards that teach general concepts of computer science. The projects in each chapter are designed to be part of an evolving story line, rather than teaching a language in depth. A good computer scientist will eventually become skilled in many programming languages, however, so it's a good idea to pair this book with study of Python or JavaScript.

Although the book is structured to these standards, the content has not been reviewed, validated, or endorsed by CSTA.

Each chapter includes a story line, suggestions for angles to research, a “Think Like a Computer Scientist” section to introduce computer science concepts, a sketch of a project, a Project Recipe with detailed instructions on how to create a project to illustrate the computer science concept, a vocabulary review, a final quiz to help emphasize concepts, and an assignment. All the projects can be built in a web browser using free and open source tools, so you won't need any special installations or downloads to get started. This book can be used for self-study, as a textbook for homeschoolers, or in a classroom. Use it chapter by chapter, piece by piece, or section by section for a more modular learning experience.

## What Does This Book Cover?

In the five sections of this book, you'll discover many aspects of computer science, presented as project-based stories:

**Part I: The Missing Fireflies** – Using Microsoft micro:bit in a browser emulator, create a home for runaway fireflies.

**Chapter 1: The Trapper**

**Chapter 2: The Firefly Refuge**

**Chapter 3: The Fireflies' Message**

**Chapter 4: A Home for All**

**Part II: The Glowing Moss** – Save the endangered moss by building a gamified enclosure for them using Microsoft MakeCode.

**Chapter 5: Goblins' Gold**

**Chapter 6: Securing the Moss**

**Chapter 7: The Whisper Network**

**Chapter 8: A Well-Tested Solution**

**Part III: Geode Quest** – Conserve the rare geodes hidden in the riverbed by using data science techniques to categorize the forest's inventory of stones.

**Chapter 9: Searching for Geodes**

**Chapter 10: Cleaning and Categorizing the Collection**

**Chapter 11: The Stone Library**

**Chapter 12: Hide or Seek**

**Part IV: The Petrified Forest** – Protect the fragile petrified wood in the forest grove by visualizing a 3D environment to simulate their preservation.

**Chapter 13: The Petrified Forest**

**Chapter 14: The Butterfly Brigade**

**Chapter 15: Power of the Weak**

**Chapter 16: The Written Artifact**

**Part V: Legends of the Field Mice** – Using Twine, a free tool to tell winding stories, help the field mice unlock the secret of their underground realm.

**Chapter 17: The Field Mice**

**Chapter 18: Lights, Sounds, Action**

**Chapter 19: Unlocking the Vault**

**Chapter 20: The Real World**

## **Additional Resources**

This textbook is paired with a full website found at <https://cs4kids.club>. This website includes a kids' section called Quests with links to completed sample projects on MakeCode, GitHub, and Twine. Worksheets can also be found that can be used to complete assignments. There's also a teacher's section with full lesson plans, answers to quizzes, differentiated learning suggestions, and more resources for each chapter's topic.

## **Standards Map**

Table I.1 includes a map of standards covered within this book along with the chapters where each appear.

**TABLE I.1 Standards mapping**

Identifier	Standard	Chapter
2-CS-01	Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.	2
2-CS-02	Design projects that combine hardware and software components to collect and exchange data.	1
2-CS-03	Systematically identify and fix problems with computing devices and their components.	3
2-NI-04	Model the role of protocols in transmitting data across networks and the Internet.	5
2-NI-05	Explain how physical and digital security measures protect electronic information.	6
2-NI-06	Apply multiple methods of encryption to model the secure transmission of information.	7
2-DA-07	Represent data using multiple encoding schemes.	9
2-DA-08	Collect data using computational tools and transform the data to make it more useful and reliable.	10
2-DA-09	Refine computational models based on the data they have generated.	11
2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms.	13
2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.	14
2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	15
2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	17
2-AP-14	Create procedures with parameters to organize code and make it easier to reuse.	18
2-AP-15	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	19
2-AP-16	Incorporate existing code, media, and libraries into original programs, and give attribution.	13
2-AP-17	Systematically test and refine programs using a range of test cases.	8

---

Identifier	Standard	Chapter
2-AP-18	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	17,18
2-AP-19	Document programs in order to make them easier to follow, test, and debug.	16
2-IC-20	Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.	20
2-IC-21	Discuss issues of bias and accessibility in the design of existing technologies.	4
2-IC-22	Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.	17
2-IC-23	Describe tradeoffs between allowing information to be public and keeping information private and secure.	12

---

# How to Contact Wiley or the Author

---

Contact the author using the Contact Us form at <https://cs4kids.club>.

If you believe you have found a mistake in this book, please bring it to our attention. At John Wiley & Sons, we understand how important it is to provide our customers with accurate content, but even with our best efforts an error may occur.

To submit your possible errata, please email them to our Customer Service Team at [wileysupport@wiley.com](mailto:wileysupport@wiley.com) with the subject line “Possible Book Errata Submission.”





# Preface

---

It's a steamy summer evening, and you struggle to concentrate on the pages of your math homework. The curtains hanging over your open window are completely motionless, as not a breath of air stirs the murky twilight. You gaze out into the gathering darkness, not thinking about anything other than whether you have the energy to walk downstairs to get another cool drink of water. Outside, the steady buzz-buzz-buzz of cicadas starts to thrum through the dusk, and as the darkness intensifies, you notice a pinpoint of light twinkling fitfully in the woods outside your house. Normally, you only visit the forest in the daytime, where you enjoy watching insects and animals go about their daily business. They always seem busy and happy, and you like observing their habits with your handy binoculars and magnifying glass that help you watch both birds and tiny bugs.

Tonight, however, the forest is strangely dark, except for that feebly flickering light. Even though your room is too warm for comfort, you feel a cold shiver down your spine. What is that light? It doesn't look like the fireflies that usually swirl merrily in the dark, flashing their gaudy yellow, white, and sometimes green luminescence. Tonight, the forest has faded to black. Have all the fireflies left? Are their glows concentrated in that one pinpoint of light? Giving up on trying to concentrate on your homework, you decide to investigate.

Rummaging through your drawer of handy tools, you grab your explorer kit containing a flashlight, some bug spray, a pad of paper and a pencil, and other random things such as a spare shoelace, an extra battery, and a stick of gum. Leaning cautiously out of your window, you shine your flashlight into

the blackness, but, as you noticed before, there is no friendly flicker of a firefly to twinkle and light up the night. Where have they gone? And why does that flickering light seem to be pulsing in a pattern now? Is it a code that you need to decipher?

...--...

You recognize the old Morse code signal: "S.O.S." It stands for Save Our Ship, and it's a distress signal! Three short flashes, three long flashes, three short flashes. Something is definitely wrong. It's time to investigate.



# The Missing Fireflies

---





# 1. The Trapper

Standard: 2-CS-02: Understand how to design a project that combines hardware and software components that collect and work with data

**Equipped with** your explorer paraphernalia, you boldly walk toward the pulsing light. Crunching through underbrush, you watch as the light becomes brighter as you approach it. Soon, you see that it's a lantern, held by . . . a cloaked figure.

You stop short as it turns to face you. Removing its hood, it is revealed to be a figure resembling a fox. Her eyes glow with a green luminescence and she raises her lantern. "Welcome, explorer," she says quietly. "Are you here to help restore this sacred space, or are you yet another one who pillages, steals, and destroys?"

Assuring the figure of your good intentions, you ask how you can help. She responds, "I'm sure you noticed how dark the forest has become. I have come to seek help in restoring the fireflies to their rightful home. Are you up to the task?"

You're quite sure that, given the right technology, you can create some kind of hardware system that will attract fireflies back to their home. You decide to build a firefly trapper gadget to make them return to the forest. Maybe, by creating a machine that mimics the flash patterns of different types of fireflies, you can attract the insects to the Trapper.

## Note

Depending on where you're from, you might call fireflies "lightning bugs." They have many other names; in Jamaica they are called "blinkies" or "peenie wallies." Their larvae are sometimes called "glow worms."

## Do Some Research

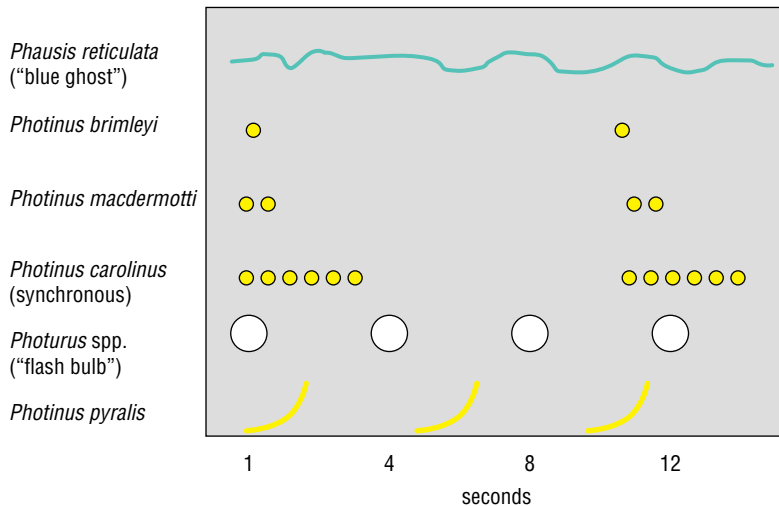
Can you build a firefly trap to capture fireflies and discover their secrets? Learn a bit about fireflies and their habits, as well as what attracts them, and you can design a firefly trapper using MakeCode's micro:bit in the emulator or, if you have it, on a real device. Your task is to use your device emulator to simulate the flashing patterns of fireflies as they try to attract mates.

### Note

MakeCode and micro:bit are free tools that we'll be using in this section to build device designs and games. MakeCode (<https://arcade.makecode.com>) has an "arcade" where you can develop simple 2D games, and micro:bit (<https://makecode.microbit.org>) helps you build hardware systems that sense, blink, and react to their surroundings. Micro:bit works both in the emulator, an online environment that simulates a real hardware device, and on its own hardware, which you can purchase online if you want. For this book, you aren't required to buy micro:bits, but they are fun gadgets to own!

All the chapters in this book use tools that work in the browser. You can develop and save your work right in your web browser as long as you have an Internet connection. To preserve your work, you can download a copy from your online browser session, or, better yet, upload a copy to GitHub if you have an account there. GitHub is a place that allows you to store your work.

According to the National Park Service, different fireflies flash various colors and patterns, depending on their type. Can you copy the pattern shown in Figure 1.1 using a micro:bit to attract a firefly?



**FIGURE 1.1: The National Park Service’s comparison of flash patterns of fireflies in the Great Smoky Mountains Park**

Your research leads you to focus on the interplay between male and female fireflies. A particular species of firefly, *Photinus brimleyi*, has an interesting pattern of communication that you can try to copy:

- The males of this species flash yellow light at intervals of 10 seconds or longer.
- Females respond to the male’s light flash with a single flash from the forest floor.

You surmise that if two fireflies flash their lights at the same time, you can consider it a “match” and the attraction will succeed—at least, that is your hypothesis.

## Sketch It Out

Draw a quick paper sketch of the elements you need to build your prototype. It’s a hardware project, but fortunately you don’t actually need hardware to

accomplish it. This challenge will allow you to learn how to prototype, or make a model of, interesting software projects using *blocks*, visual representations of your software that snap together like puzzle pieces. You can do this in a web browser using MakeCode’s micro:bit emulator.

The micro:bit emulator also includes Python and JavaScript tabs if you want to jump right into writing code rather than using blocks. For this prototype, however, you can use blocks. Don’t worry if you don’t know Python or JavaScript for the projects in this book. These programming languages, which are used to build websites and work with data, are very useful to know but they are not covered in depth here. To dive deeper into these languages, you can use the many free online tools, such as [FreeCodeCamp.org](https://www.freecodecamp.org) and [Code.org](https://code.org).

Since you need to create a device that simulates the colored flashing lights of male and female fireflies, and the basic micro:bit device only has a few LED lights available as built-in lights and they can only flash red light, you need to add two “strips” of LEDs that will act like the fireflies.

### Definition

*Light-emitting diodes (LEDs)* are a semiconductor light source that uses less power to emit light than a conventional incandescent bulb, and so are often used in low-power applications.

Draw a sketch of your device with two LED strips connected to the micro:bit and the score displayed on the built-in LED display. Your sketch might look something like Figure 1.2.

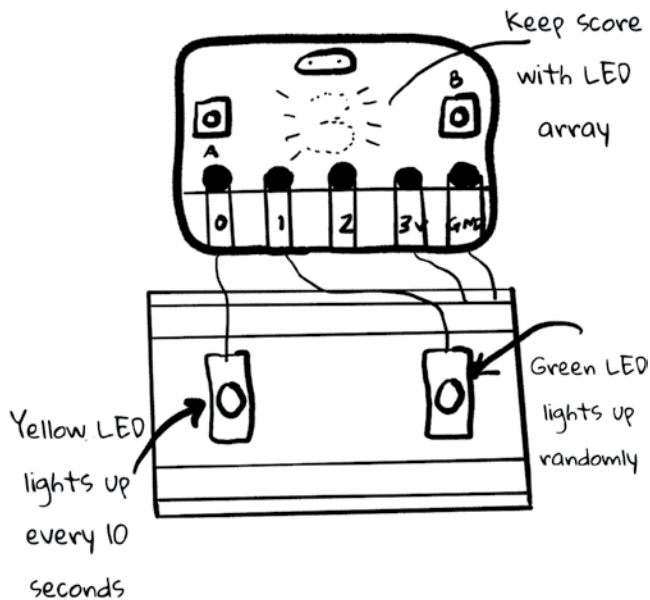
## Your Challenge

Your job is to design a program that will:

- Reproduce a sequence of yellow male light flashes that appear every 10 seconds



- Generate random flashes of white light to reproduce the female flash
- Create a system whereby if a female flash happens at the same time as a male flash, a “match” can be made (in your case, a point tallied)



**FIGURE 1.2:** A sketch of how your device might look

With this plan, or software specification, in hand, you are ready to build a prototype of your firefly trapper using blocks, right in the MakeCode micro:bit emulator.

## Think Like a Computer Scientist: Design Your System

Your task is to design a project that combines hardware and software components to collect and exchange information. What does that look like? Imagine you are designing a device that can collect the temperature from a sensor.