

# Computer Science

By Dr. Mark Light, Leader, Ohio 4-H STEM and Digital Engagement Innovations, 4-H Youth Development.

Reviewed by 4-H Youth Development professionals: Molly Avers, Educator; Margo Long, Workforce Prep Program Manager, Christy Millhouse, STEM Educator; and Jess Soffee, Educator.

Computer science is the study of computers, computing, and information processing. It involves everything from understanding how computers work to designing software systems that solve complex problems.

Understanding computer science is helpful even if you are not a computer programmer. Most careers involve technology and devices that are needed to conduct business. Basic computer knowledge is a necessity, and if you know how to communicate with a computer programmer, even better. Now is a great time to get a basic understanding of computer science and different levels of computer programming.

## Computational Thinking

At its core, computer science is about using computational thinking to solve problems. This means breaking down problems into smaller, more manageable parts and designing algorithms and programs to solve them. Whether or not you realize it, you use computational thinking every day. Think about how you usually get ready for school. You probably do not think about it, but your morning routine can be broken up into a smaller set of instructions such as, get out of bed, walk to the bathroom, shower, get dressed, and so forth. Most complex tasks can be broken down into smaller, manageable bits.

Computational thinking is a valuable skill in today's data-driven world, as it helps individuals analyze complex problems, develop efficient solutions, and communicate ideas effectively.



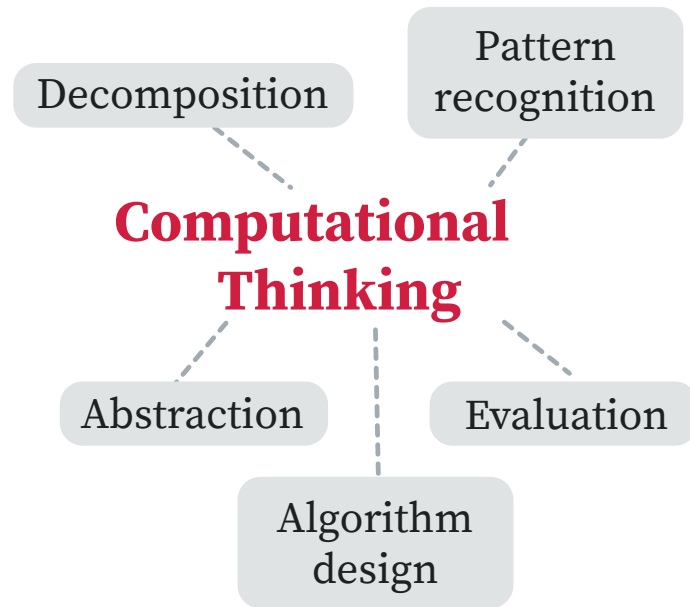
## PLAN YOUR PROJECT

Use this idea starter AND publication 4-H 365 *Self-Determined Project Guide* as the starting place for your 4-H self-determined project. The *Self-Determined Project Guide* is available from your county OSU Extension office or on the Web at [ohio4h.org/selfdetermined](http://ohio4h.org/selfdetermined). You may choose to do a little or a lot depending on your level of interest. Be sure to register your project with your county OSU Extension office.



Computational thinking includes these core elements:

- **Decomposition.** Breaking down a problem into smaller, more manageable parts.
- **Pattern recognition.** Identifying patterns and trends in data or information.
- **Abstraction.** Focusing on the important information and ignoring irrelevant details.
- **Algorithm design.** Developing step-by-step instructions for solving a problem.
- **Evaluation.** Testing and refining solutions to ensure they are effective.



**Functions.** Functions are blocks of code that can be called multiple times within a program. They are used to group related code and make it easier to reuse.

**Control structures.** These are used to control the flow of a program. Common control structures include loops (repeating instructions multiple times) and conditional statements (executing different instructions based on conditions).

**Input/output.** A program often needs to take input from the user and display output. Most programming languages provide functions or libraries for handling input/output.

**Error handling.** Programs often encounter errors or unexpected situations. Error handling is a way to gracefully manage these situations and provide meaningful feedback to the user.

Computational thinking is not limited to computer science and programming. It is routinely applied to a wide range of fields, from science and engineering to business and social sciences. It is a valuable skill for anyone who needs to solve complex problems, analyze data, or develop innovative solutions.

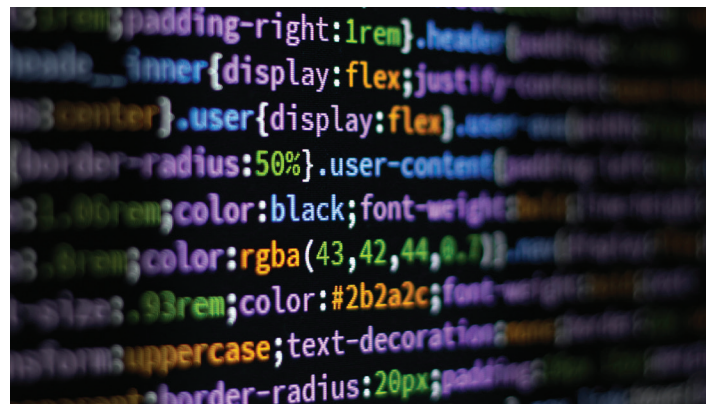
Even if computer programming and coding are not part of your career path, there will be times when you need to solve complex problems. Understanding coding and computer science can also help you communicate needs to your organization to resolve problems.

## Coding Syntax

In computer programming, syntax refers to the set of rules for writing code in a particular programming language. Syntax specifies the correct order and structure of keywords, symbols, and other elements used to write code. Every coding language has its own set of syntax. There are, however, basic principles that apply to most coding languages:

**Variables.** Variables are used to store data in a program. They can be assigned values and their values can be changed throughout the program.

**Data types.** Every coding language has different data types to represent different kinds of data. For example, integers represent whole numbers, while strings represent text.



## Programming Languages

Digital devices are made of many parts, called components. When assembled, components work together so that you can use the device reliably and

efficiently. Technology devices, called hardware, cannot operate without instructions, called software or code. And code cannot be written without a programming language. Many coding languages are used in computer science, each with its own strengths and weaknesses.

Some of the most popular languages are described here:

**Java.** A language used in enterprise-level applications, web applications, and Android apps.

**Python.** A versatile language used for web development, scientific computing, artificial intelligence, and machine learning.

**Swift.** A language developed by Apple and used for building iOS and macOS applications.

**C#.** A language developed by Microsoft used for building Windows applications, games, and enterprise software.

**C++.** A high-performance language used for system-level programming, game development, and building large-scale software.

Other languages in computer science include Ruby, HTML, Go, SQL, PHP, and JavaScript. Each has a unique application that makes it best for a given job.

---

## AREAS OF INTEREST AND THINGS TO DO

Every self-determined 4-H project can be broken down into areas of interest. These are the specific things members want to address during their project adventures. Using 4-H 365 *Self-Determined Project Guide*, identify at least three areas of interest with at least three activities per area to explore. ***If you are a 4-H member, one area of interest should include coding a project to take to county 4-H judging. Examples include a Sphero robot, Apple Swift Coding, app development, or something using another coding language you learn.*** Take your ideas from the following list or make up your own.

---

### Computer Science in Our Lives

- Make a list of how computer science is used in your home, school, or community. Share your list to see if someone can add more.
- Explore the different ways your phone uses algorithms to solve everyday problems. What about a different phone? Does it solve problems the same way?
- Investigate the future uses of computer technology in artificial intelligence. Share what you learn with your project helper.
- Find your phone settings and take more control over your notifications and app features. Offer to do the same for someone you know.

### Computer Science in Careers

- Interview someone who codes or who works with technology daily. Do they have a job that interests you? Why or why not?
- Investigate computer science career possibilities. Make a video, flyer, or poster that promotes your findings.
- Tour a tech company in Ohio. Put what you learn into a brochure to attract new employees.
- Explore the websites of at two tech companies. What careers are open right now and what skills are needed?
- Job shadow someone who is in the technology field.

### Computer Science in Action

- Identify examples of open-source code on the web.
- Go to **Code.org** and try out coding activities. Share your favorites with someone younger than you or find a way to compete with a friend.
- Download a coding app and practice computer science exercises. Examples of coding apps include ScratchJr, SpheroEdu, CodeSpark, Tynker, and Swift Playgrounds.
- Visit **scratch.mit.edu** and make your own story, game, or animation. Share what you create on the scratch website for others to see.
- Complete Learn to Code exercises in Apple's Swift Playgrounds app for iPad and Mac. These exercises move a character named Byte, through a virtual world using code you write.
- Explore CS Lessons on **experience-cs.org/?sjid=14171179579344759132-NC**

## Coding with Sphero Robotics

- ❑ Use the SpheroEdu app at [edu.sphero.com](https://edu.sphero.com).
- ❑ Explore activities that were created by others to learn to code.
- ❑ Create your own programming through block or text coding.
- ❑ Code your Sphero to move forward and make turns.
- ❑ Code your Sphero to create shapes like a square, rectangle, triangle or circle.
- ❑ Create a loop to repeat code.
- ❑ Design a maze using multiple turns and obstacles. Code your Sphero to navigate the maze.
- ❑ Create your own program to showcase at 4-H project judging.
- ❑ Teach what you have learned through a demonstration at your 4-H club or other youth group.

## Coding through Programming

- ❑ Choose a programming language to explore. Find resources on the internet (web pages or videos) to help you learn that language.
- ❑ Complete the computer science modules on the national 4-H CLOVER site by signing up at [4-h.org/clover](https://4-h.org/clover).
- ❑ Storyboard an app you are interested in creating. Using Keynote or PowerPoint, each slide would be a different screen of your app.



---

## RELATED RESOURCES

---

Clover by 4-H, [4-h.org/clover](https://4-h.org/clover)

Computer Science Coding, [code.org](https://code.org)

Sphero Education Lessons, [edu.sphero.com](https://edu.sphero.com)

Computer Science Game Creation, [scratch.mit.edu](https://scratch.mit.edu)

Raspberry Pi Foundation, Experience CS,  
[experience-cs.org/?sjid=14171179579344759132-NC](https://experience-cs.org/?sjid=14171179579344759132-NC)

---

Ohio State University Extension is part of The Ohio State University College of Food, Agricultural, and Environmental Sciences.

CFAES provides research and related educational programs to clientele on a nondiscriminatory basis. For more information, visit [cfaes.osu.edu/nondiscrimination](https://cfaes.osu.edu/nondiscrimination). For an accessible format of this publication, visit [cfaes.osu.edu/accessibility](https://cfaes.osu.edu/accessibility).