(//code.org)

Hi Jonathan ▼

My Dashboard (//studio.code.org/home) Course Catalog (//studio.code.org/courses)

Projects (//studio.code.org/projects)

Professional Learning (//studio.code.org/my-professional-learning)

Computer Science Principles

What is CS Principles?

Computer Science Principles introduces students to the foundational concepts of computer science and challenges them to explore how computing and technology can impact the world. More than a traditional introduction to programming, it is a rigorous, engaging, and approachable course that explores many of the foundational ideas of computing so all students understand how these concepts are transforming the world we live in.

This year-long course can be taught as an AP or non-AP course - no prerequisites required for students or for teachers new to computer science! In addition, our curriculum is available at no cost for anyone, anywhere to teach. For more information about our goals and approach to our courses, please see our **curriculum values (/educate/curriculum/values)** and our **professional learning values (/educate/professional-learning/values)**.

Computer Science Principles

(//studio.code.org/courses/csp)

Recommended for Grades 9-12

Computer Science Principles is a course designed to prepare students (and teachers) who are new to computer science for the AP CS Principles exam. The course covers many topics including the Internet, Big Data and Privacy, and Programming and Algorithms.

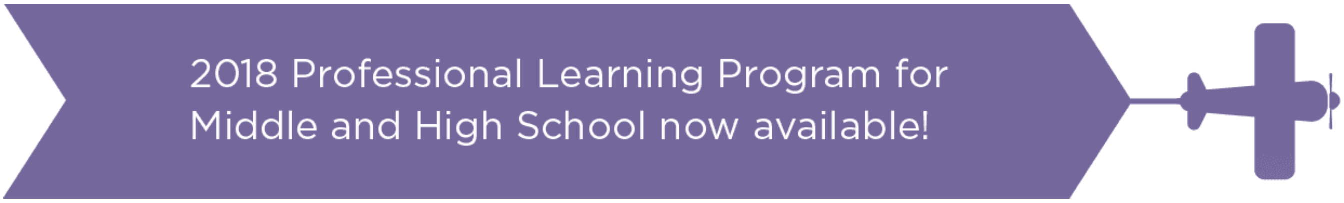
View course **(//studio.code.org/courses/csp)**

Lesson plans **(https://curriculum.code.org/csp)**

Professional Learning - Applications now available!

Our middle and high school programs offer year-round support. The program kicks off with a 5-day summer workshop where you'll have an opportunity to work hands-on with the curriculum and meet other teachers from your area. Throughout the year, we offer online support for upcoming units, forum support, and 1-day quarterly workshops. You don't need any prior computer science experience to get started. And teachers love it! 90% rank it the best professional development ever.

Applications for the CS Principles Professional Learning Program (/educate/professional-learning/cs-principles) are now available!


2018 Professional Learning Program for
Middle and High School now available!

Learn more and apply now! **(/educate/professional-learning/cs-principles)**

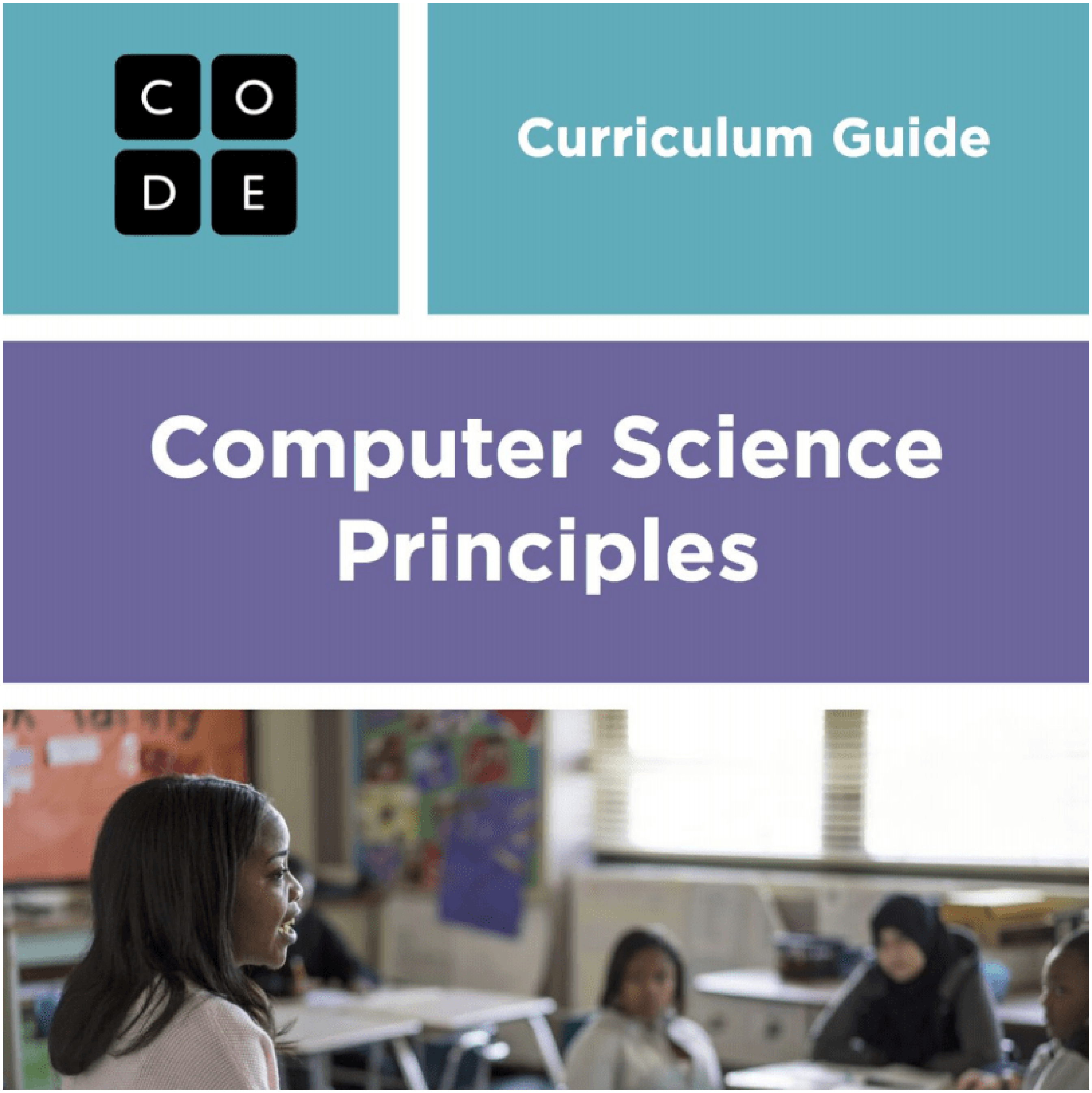
Code.org's AP CS Principles Curriculum

Code.org is recognized by the **College Board (http://collegeboard.org/APCSP)** as an endorsed provider of curriculum and professional development for AP® Computer Science Principles. This endorsement affirms that all components of Code.org CS Principles's offerings are aligned to the AP Curriculum Framework standards and the AP CS Principles assessment. Using an endorsed provider affords schools access to resources including an AP CS Principles syllabus pre-approved by the College Board's AP Course Audit, and officially recognized professional development that prepares teachers to teach this course.

- **Code.org 2017-18 AP CS Principles Endorsed Syllabus (/files/CSPSyllabusMay2017.pdf)**
- **Instructions: How to add our authorized syllabus to your AP Course Audit (https://drive.google.com/open?id=0B21d4g64Z62YSVptX3JKcWdiVWlKWVc5MWlXQldkUmVxLXFZ)**

Curriculum Resources and Features

(/files/CSP_CurriculumGuide_2017_forWeb.pdf)



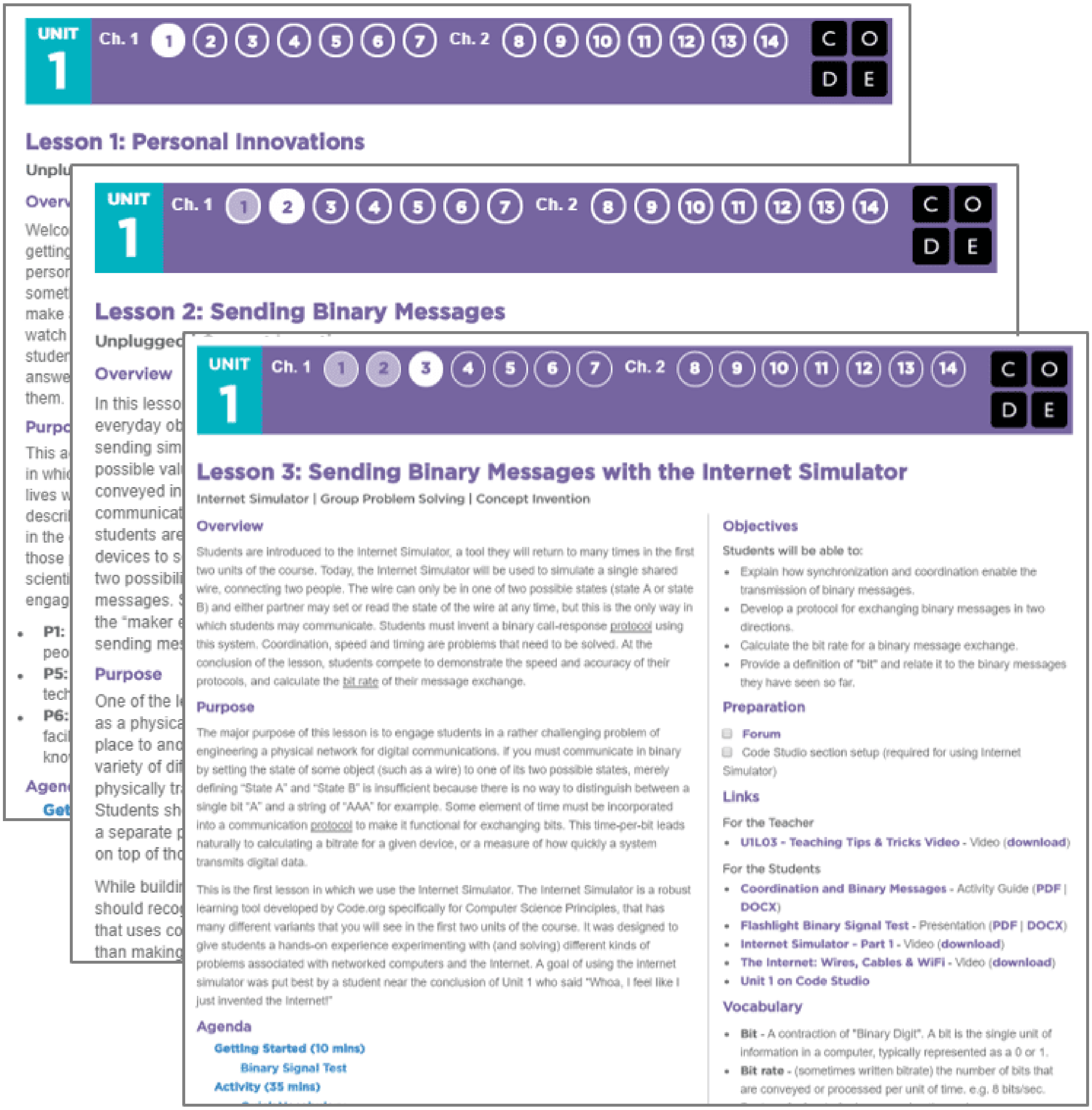
Curriculum Guide

Computer Science Principles

(/files/CSP_CurriculumGuide_2017_forWeb.pdf)

The **Curriculum Guide (/files/CSP_CurriculumGuide_2017_forWeb.pdf)** contains details on the classroom and student practices that flow throughout the course, overviews of each unit, and implementation considerations.

(https://curriculum.code.org/csp)



Lesson 1: Personal Innovations

Lesson 2: Sending Binary Messages

Unplugged

Lesson 3: Sending Binary Messages with the Internet Simulator

Internet Simulator | Group Problem Solving | Concept Invention

Overview

Students are introduced to the Internet Simulator, a tool they will return to many times in the first two units of the course. Today, the Internet Simulator will be used to simulate a single shared wire, connecting two people. The wire can only be in one of two possible states (state A or state B) and either partner may set or read the state of the wire at any time, but this is the only way in which students may communicate. Students must invent a binary call-response protocol using this system. Coordination, speed and timing are problems that need to be solved. At the conclusion of the lesson, students compete to demonstrate the speed and accuracy of their protocols, and calculate the bit rate of their message exchange.

Purpose

The major purpose of this lesson is to engage students in a rather challenging problem of engineering a physical network for digital communications. If you must communicate in binary by setting the state of some object (such as a wire) to one of its two possible states, merely defining "State A" and "State B" is insufficient because there is no way to distinguish between a single bit "A" and a string of "AAA" for example. Some element of time must be incorporated into a communication protocol to make it functional for exchanging bits. This time-per-bit leads naturally to calculating a bitrate for a given device, or a measure of how quickly a system transmits digital data.

This is the first lesson in which we use the Internet Simulator. The Internet Simulator is a robust learning tool developed by Code.org specifically for Computer Science Principles, that has many different variants that you will see in the first two units of the course. It was designed to give students a hands-on experience experimenting with (and solving) different kinds of problems associated with networked computers and the Internet. A goal of using the internet simulator was put best by a student near the conclusion of Unit 1 who said "Whoa, I feel like I just invented the Internet!"

Agenda

- **Getting Started (10 mins)**
  - **Binary Signal Test**
- **Activity (35 mins)**

Objectives

Students will be able to:

- Explain how synchronization and coordination enable the transmission of binary messages.
- Develop a protocol for exchanging binary messages in two directions.
- Calculate the bit rate for a binary message exchange.
- Provide a definition of "bit" and relate it to the binary messages they have seen so far.

Preparation

□ **Forum**
□ Code Studio section setup (required for using Internet Simulator)

Links

For the Teacher

- **U1L03 - Teaching Tips & Tricks Video** - Video (**download**)

For the Students

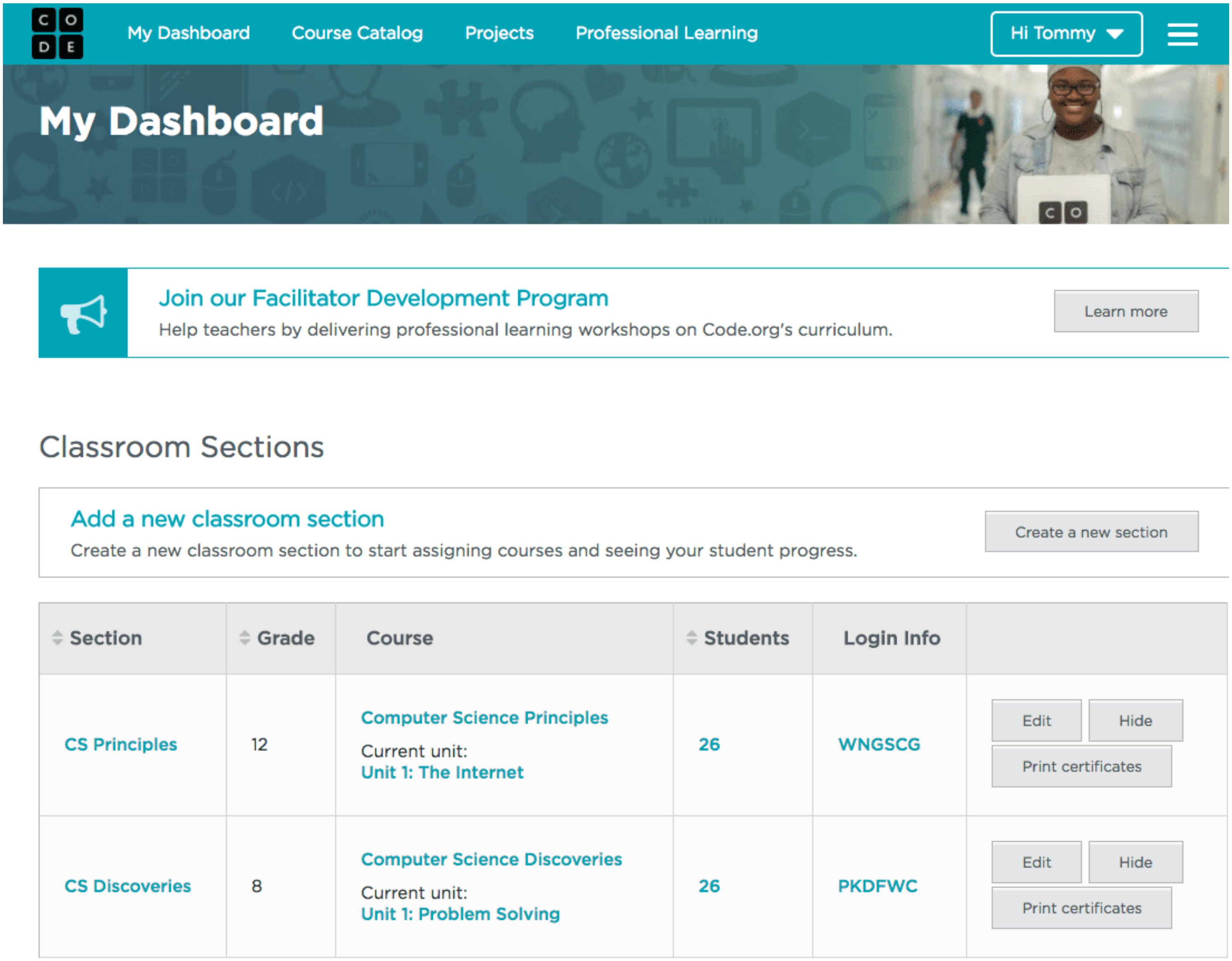
- **Coordination and Binary Messages** - Activity Guide (**PDF** | **DOCX**)
- **Flashlight Binary Signal Test** - Presentation (**PDF** | **DOCX**)
- **Internet Simulator - Part 1** - Video (**download**)
- **The Internet: Wires, Cables & WiFi** - Video (**download**)
- **Unit 1 on Code Studio**

Vocabulary

- **Bit** - A contraction of "Binary Digit". A bit is the single unit of information in a computer, typically represented as a 0 or 1.
- **Bit rate** - (sometimes written bitrate) the number of bits that are conveyed or processed per unit of time. e.g. 8 bits/sec.

(https://curriculum.code.org/csp)

Daily lesson plans (https://curriculum.code.org/csp) come with detailed instructions, activity guides, assessments, standards mappings, and more.
(https://studio.code.org/home)



Join our Facilitator Development Program

Help teachers by delivering professional learning workshops on Code.org's curriculum.

Classroom Sections

Add a new classroom section

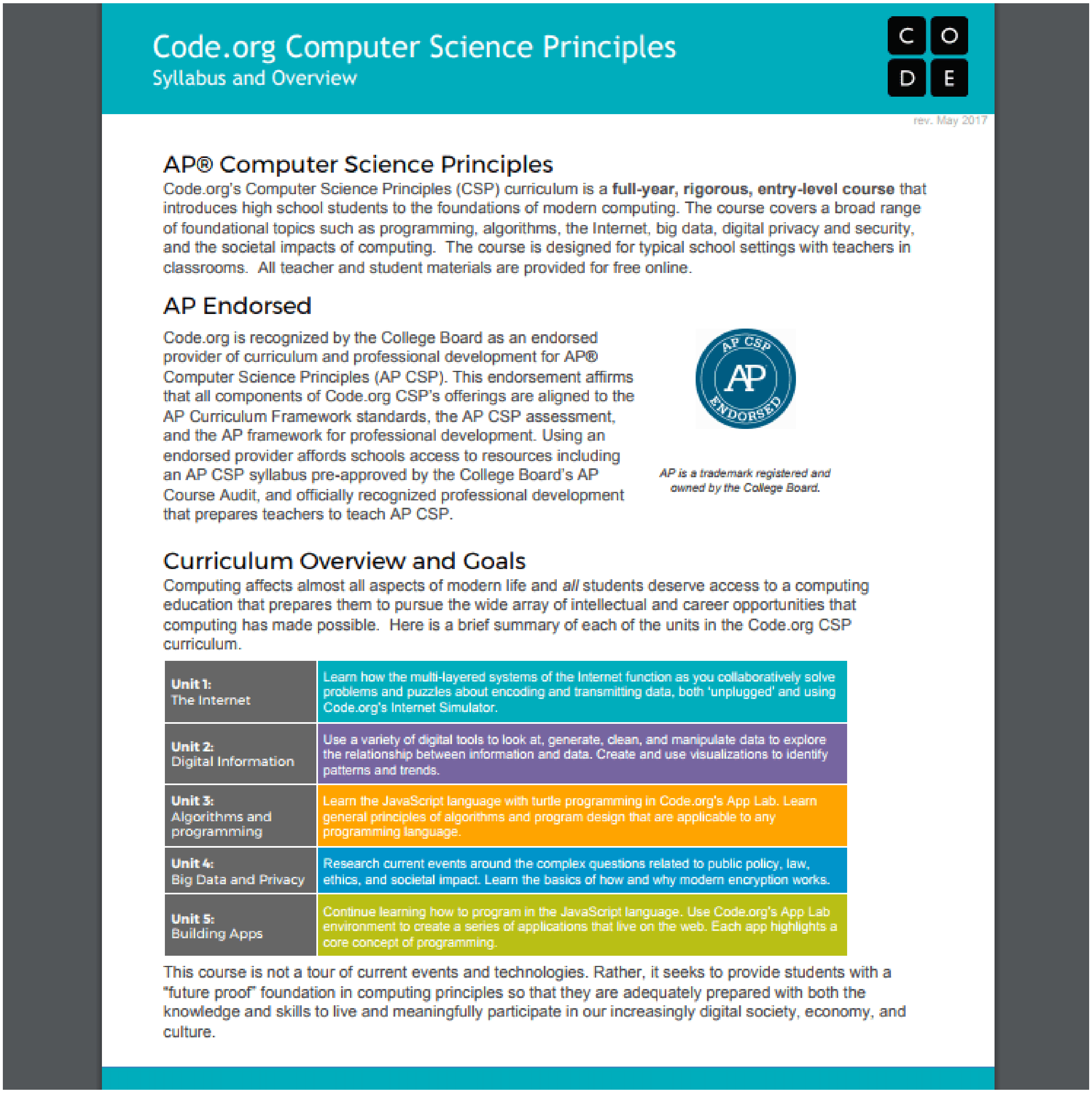
Create a new classroom section to start assigning courses and seeing your student progress.

Section	Grade	Course	Students	Login Info	
CS Principles	12	Computer Science Principles Current unit: Unit 1: The Internet	26	WNGSCG	Edit Hide Print certificates
CS Discoveries	8	Computer Science Discoveries Current unit: Unit 1: Problem Solving	26	PKDFWC	Edit Hide Print certificates

(https://studio.code.org/home)

Create and manage your classroom sections, check student progress, administer assessments, and integrate with your Google or Clever classroom - all from your **Teacher Dashboard (https://studio.code.org/home)**!

(/files/CSPSyllabusMay2017.pdf)



Code.org Computer Science Principles

Syllabus and Overview

rev. May 2017

AP® Computer Science Principles

Code.org's Computer Science Principles (CSP) curriculum is a **full-year, rigorous, entry-level course** that introduces high school students to the foundations of modern computing. The course covers a broad range of foundational topics such as programming, algorithms, the Internet, big data, digital privacy and security, and the societal impacts of computing. The course is designed for typical school settings with teachers in classrooms. All teacher and student materials are provided for free online.

AP Endorsed

Code.org is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). This endorsement affirms that all components of Code.org CSP's offerings are aligned to the AP Curriculum Framework standards, the AP CSP assessment, and the AP framework for professional development. Using an endorsed provider affords schools access to resources including an AP CSP syllabus pre-approved by the College Board's AP Course Audit, and officially recognized professional development that prepares teachers to teach AP CSP.

AP CSP ENDORSED

AP is a trademark registered and owned by the College Board.

Curriculum Overview and Goals

Computing affects almost all aspects of modern life and *all* students deserve access to a computing education that prepares them to pursue the wide array of intellectual and career opportunities that computing has made possible. Here is a brief summary of each of the units in the Code.org CSP curriculum.

Unit 1: The Internet	Learn how the multi-layered systems of the Internet function as you collaboratively solve problems and puzzles about encoding and transmitting data, both 'unplugged' and using Code.org's Internet Simulator.
Unit 2: Digital Information	Use a variety of digital tools to look at, generate, clean, and manipulate data to explore the relationship between information and data. Create and use visualizations to identify patterns and trends.
Unit 3: Algorithms and programming	Learn the JavaScript language with turtle programming in Code.org's App Lab. Learn general principles of algorithms and program design that are applicable to any programming language.
Unit 4: Big Data and Privacy	Research current events around the complex questions related to public policy, law, ethics, and societal impact. Learn the basics of how and why modern encryption works.
Unit 5: Building Apps	Continue learning how to program in the JavaScript language. Use Code.org's App Lab environment to create a series of applications that live on the web. Each app highlights a core concept of programming.

This course is not a tour of current events and technologies. Rather, it seeks to provide students with a "future proof" foundation in computing principles so that they are adequately prepared with both the knowledge and skills to live and meaningfully participate in our increasingly digital society, economy, and culture.

(/files/CSPSyllabusMay2017.pdf)

Teaching CS Principles as an AP course? Don't forget to submit the **2017 AP Endorsed Syllabus (/files/CSPSyllabusMay2017.pdf)** for the AP course audit.

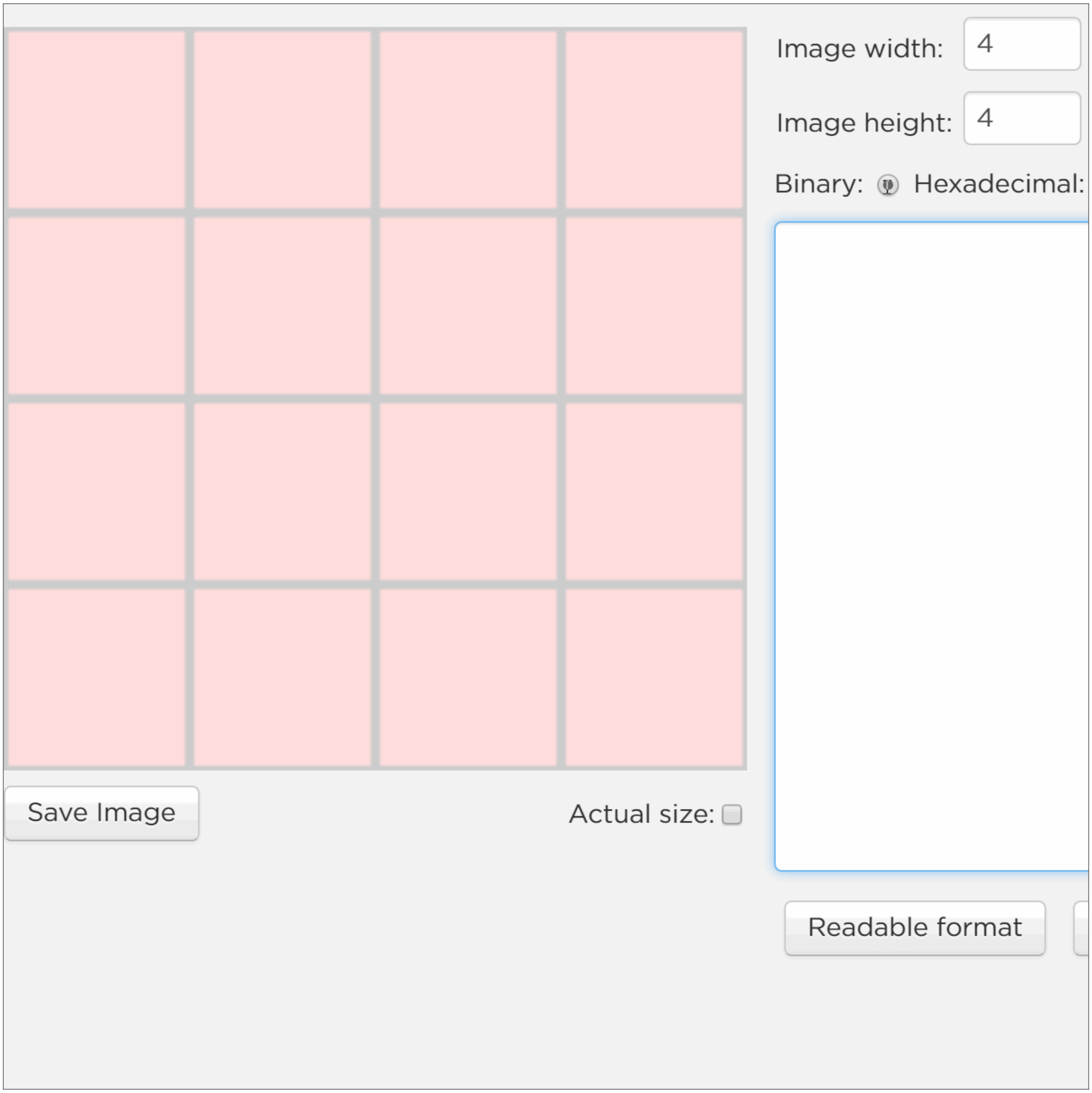


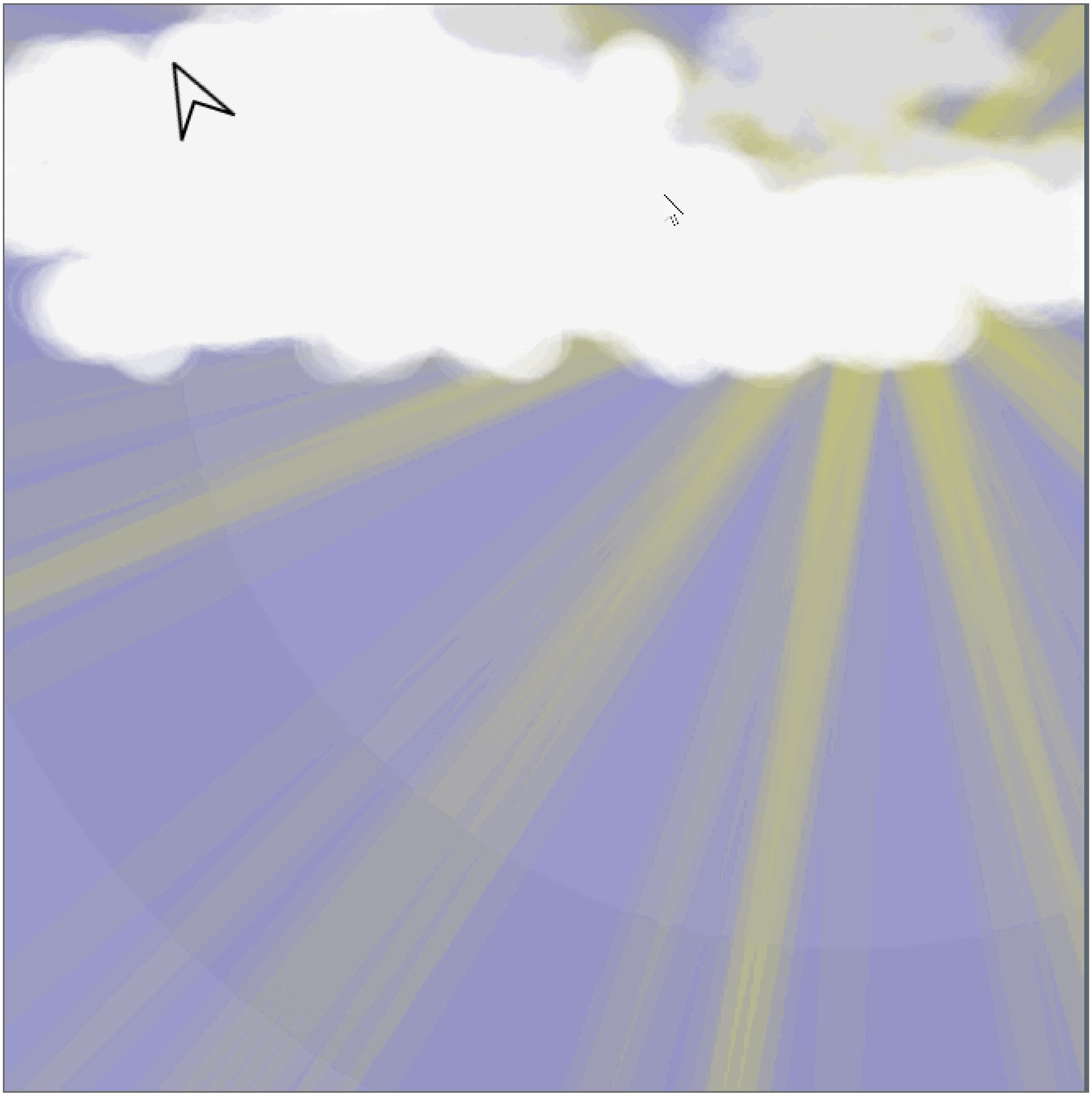
Image width: 4

Image height: 4

Binary: ○ Hexadecimal:

Actual size: □

Students get hands on experience with concepts like binary and pixels through **computational widgets. (/educate/csp/widgets)**



Quickly create shareable apps with **Code.org's App Lab (/educate/applab)** - an online block to text, JavaScript programming environment.

Don't miss these additional resources:

- **Course Materials List (#materials)**
- **CS Principles Video Library (https://www.youtube.com/playlist?list=PLzdnOPI1iJNfV5ljCxR8BZWJRT_m_6CpB)**

Unit Overview

Unit 1

(//studio.code.org/s/csp1)

The Internet

Students learn how the multi-layered systems of the Internet function as they collaboratively solve problems and puzzles about encoding and transmitting data, both 'unplugged' and using Code.org's Internet Simulator.

View unit (//studio.code.org/s/csp1)

Lesson plans (https://curriculum.code.org/csp/unit1/)

Unit 2

(//studio.code.org/s/csp2)

Digital Information

Using a variety of tools, students look at, generate, clean, and manipulate data to explore the relationship between information and data. Students also create and use visualizations to identify patterns and trends.

View unit (//studio.code.org/s/csp2)

Lesson plans (https://curriculum.code.org/csp/unit2/)

Unit 3

(//studio.code.org/s/csp3)

Algorithms and Programming

In this unit, students learn the JavaScript language with turtle programming in Code.org's App Lab. Students are introduced the general principles of algorithms and program design that are applicapble to any programming language.

View unit (//studio.code.org/s/csp3)

Lesson plans (https://curriculum.code.org/csp/unit3/)

Unit 4

(//studio.code.org/s/csp4)

Big Data and Privacy

Students research current events around the complex questions related to public policy, law, ethics, and societal impact. Students are also introduced to the basics of how and why modern encryption works.

View unit **(//studio.code.org/s/csp4)**

Lesson plans

(https://curriculum.code.org/csp/unit4/)

Unit 5

(//studio.code.org/s/csp5)

Building Apps

Students continue learning how to program in the JavaScript language. Using Code.org's App Lab environment, students create a series of applications that live on the web. Each app highlights a core concept of programming.

View unit **(//studio.code.org/s/csp5)**

Lesson plans **(https://curriculum.code.org/csp/unit5/)**

AP Explore Performance Task

(//studio.code.org/s/csp-explore)

Explore - Impact of Computing Innovations

This unit provides activities and resources for students to prepare for and complete the AP Explore Performance Task, which requires students to identify a computing innovation, explore its impact, and create a related digital artifact.

View unit (//studio.code.org/s/csp-explore)

Lesson plans (https://curriculum.code.org/csp/csp-explore/)

AP Create Performance Task

(//studio.code.org/s/csp-create)

Create - Applications from Ideas

This unit provides activities and resources for students to prepare for and complete the AP Create Performance Task, which asks students to develop a program on a topic that interests them or one that solves a problem.

View unit (//studio.code.org/s/csp-create)

Lesson plans (https://curriculum.code.org/csp/csp-create/)

Post-AP

(//studio.code.org/s/csppostap)

Making Data-backed Apps

After the AP Test, students learn how to use App Lab's database capabilities to make apps that store data in the cloud so that it can be retrieved later.

View unit (//studio.code.org/s/csppostap)

Keep up with CS Principles

Let us know how we're doing

We are always looking for ways to improve our courses. If something's not quite working, or you have ideas about features that you'd like to see, we'd love to hear from you!

Give feedback **(mailto:support@code.org)**

Subscribe for the latest news

Sign up for our monthly emails, which contain the latest news about tools, videos, and other important updates for CS Principles.

Sign up **(/educate/csp/CSPStatus_Signup)**

Chat about CS Principles in the forum

We have forums for educators to discuss and trade ideas about CS Principles and talk about the curriculum. Code.org forums are used for all of our courses, K-12.

Visit the forum **(http://forum.code.org/c/csp)**

Spread the word about CS Principles

Hand out these fliers around your school and consider sending home to parents.

Computer Science Principles
Curriculum Overview

Why Computer Science? Every 21st century student should have the opportunity to learn computer science. The basics of computer science help nurture creativity and problem-solving skills, preparing students for a future in any field or career.

Advanced Placement Computer Science for All Students!

[illegible]

Engaging Curriculum

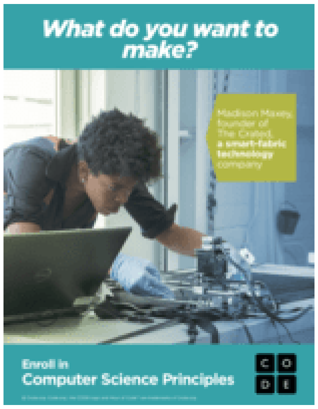
[illegible]

One Year Professional Learning Program

- Summer: [illegible]
- School Year: [illegible]

Hundreds of teachers participate each year. They overwhelmingly agree: "It's the best professional development I've ever attended."

(/files/CSPrinciples_1-pager.pdf)


What do you want to make?
Madison Maxey, founder of The Crated, a smart-fabric technology company
Enroll in Computer Science Principles
C O D E

(/files/programs/csp-student-flyer.pdf)

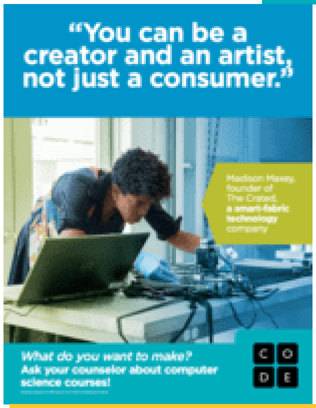
Hang these posters in your classroom and around your school:

"When things get tough, dig deep—your future is yours to build."
Rory and Kieran O'Reilly, co-founders of gifts.com, where you can build your own gifts
What do you want to make? Ask your counselor about computer science courses!
C O D E

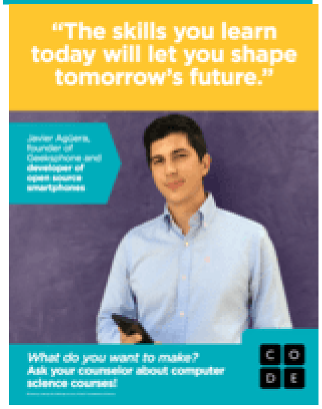
(/files/programs/kieran_rory_CS_recruitment.pdf)

"Coding is deeply creative work."
Nidhi Kulkarni and Erin Parker, co-founders of Spitfire Athlete, a strength training app
What do you want to make? Ask your counselor about computer science courses!
C O D E

(/files/programs/kulkarni_parker_CS_recruitment.pdf)


"You can be a creator and an artist, not just a consumer."
Madison Maxey, founder of The Crated, a smart-fabric technology company
What do you want to make? Ask your counselor about computer science courses!
C O D E

(/files/programs/madison_maxey_CS_recruitment.pdf)


"The skills you learn today will let you shape tomorrow's future."
Javier Agüera, founder of Geeksphone and developer of open source smartphones
What do you want to make? Ask your counselor about computer science courses!
C O D E

(/files/programs/javier_aguera_CS_recruitment.pdf)

Share these videos with students, parents, or administrators:

WHAT IS
COMPUTER SCIENCE PRINCIPLES?

WHAT'S UP WITH
COMPUTER SCIENCE PRINCIPLES?

Frequently Asked Questions

What materials do I need for this course?

Required Materials:

This course requires students have access to computers with a modern web browser. For more details, check out Code.org's technology requirements, **here (/educate/it)**.

Many lessons have handouts that are designed to guide students through activities. While these handouts are not required, we highly recommend their use. In addition to handouts, you will need the following:

- Unit 1 Lesson 2 requires some craft materials for constructing physical devices. The lesson recommends items like cups, string/yarn, construction paper, flashlights, slinkies, noise makers, markers, and glue.

Optional Materials

The following items are called for in lessons, but alternatives are offered below:

- (Unit 3, Lesson 1) A handful of legos per 2-3 students. Alternatives: post-it notes, construction paper
- (Unit 3, Lesson 2 - 3) Playing cards (1 deck per 6 students). Alternatives: write numbers of post-it notes.

- (Unit 4, Lesson 8) Clear dixie cups with beans. Alternatives: Any clear container (ziplock bag, empty water bottle, etc) with any small item (beads, raisins, coffee beans, etc)

The following supplies are completely optional but will be useful to have on hand for various lessons:

- Graph paper
- Chart paper
- Markers
- Post-it notes

How can I access answer keys?

With an approved teacher account you can find answer keys in a blue "Teacher Only" panel that shows in the online lessons and activities.

Teachers in our professional learning program will automatically be approved to view answer keys. If you need an approved teacher account, any teacher can apply for Code Studio access to protected teacher-only materials (answer keys, etc) through **this form (https://docs.google.com/forms/d/1f5QPKi3F_3nBDR8q9BcXCqixzY7SCQd7Seob0-JYizU/viewform)**.

How can I communicate with other teachers who are using the curriculum?

Check out our forum at **(forum.code.org)http://forum.code.org (http://forum.code.org)**. There you'll find a space for general CS Principles discussion as well as unit- and lesson-specific threads.

Can I give feedback on the lessons?

Yes, please! Our desire is that the curriculum will be a living document and not something set in stone. We are open to changes or alternatives to lessons so please send us your feedback by using the CS Principles forum.

How/Where can I get professional development for this course?

We have an in-person professional learning program... **click here (/educate/professional-learning/cs-principles)** to learn more!

How can I learn about what's new or changing with the curriculum?

We send out monthly updates! **Sign up for future emails (http://code.org/educate/csp/CSPStatus_Signup)**.

Ideas and Inspirations

Please see this page about **CS Principles Inspirations (/educate/csp/inspirations)** to read about the influential works in computer science education that form some of the course's philosophical underpinnings.

AP® is a registered trademark of the College Board.

Privacy Policy (/privacy) | About (/about) | Partners (/partners) | Blog (https://medium.com/@codeorg) | Donate (/donate) | Store (/shop) | Support (http://support.code.org/) | Terms (/tos)

Privacy Policy (/privacy) | About (/about) | Partners (/partners) | Blog (https://medium.com/@codeorg) | Donate (/donate) | Store (/shop) | Support (https://support.code.org/) | Terms (/tos)

English