The Big Idea: *The best way to overcome prejudice and inequality in Computer Science is to integrate it into the school day.*

Coding (computer programming) means telling a computer what to do in a language it can understand. **Computer Science** is the study of computation. *Coding is a small part of Computer Science*.

Computational Thinking is the transferrable part of Computer Science. It is composed of five fundamental skills that can be applied to any type of problem:

Decomposition - breaking a problem into smaller problems
Generalization - comparing this problem to other problems
Abstraction - deciding which details don't matter
Pattern Recognition - deciding which parts repeat
Algorithm Design - solving all the problems of one type

What do we give students when we teach them to code?

1. Normalize computer programming
2. Identify as a programmer
3. Learn Computational Thinking skills
4. Use programming tools to for self-expression

What level of Computer Science is appropriate for my students?

Elementary School (3-5)

- Exposure to Computational Thinking skills.
- Understand that coding is about problem solving.

Middle School (6-8)

- Use text-based programming tools on the web.
- Practice Computational Thinking skills.
- Understand that coding is math.

High School (9-12)

- Transfer Computational Thinking skills to new domains.
- Offer dedicated Computer Science courses.

Replace or enhance **1 in 4** creative projects (posters, dioramas, videos, models, skits) with a creative coding challenge. Use the following sample lesson plans as inspiration.

Intro to Programming and JavaScript

Big Idea: You can be a programmer.

Module: Make Your Own Filter https://app.vidcode.io/project/filters

Time: 1 hour

15 minutes background
40 minutes coding
5 minutes reflection

Vocabulary

Function, Argument, Type, String, Number

Standards

CCSS.MATH.PRACTICE.MP7 Look for and make use of structure.

NGSS Engineering Practice 5 Using mathematics and computational thinking

Background (15 mins)

The basics of JavaScript

JavaScript is a programming language. Since computers don't speak human languages like English or Spanish, we use programming languages to talk to them. JavaScript is the programming language that we can use to talk directly to web pages.

Functions are the main way of getting things done in JavaScript. A function is an action that has a name. Functions are written with parentheses.

```
blur(50);
```

Some functions need one or more **arguments** inside the parentheses. This tells the function extra information about what to do.

Every argument has a **type**. In the function call below, “blue” is a string and 25 is a number.

```
tint(“blue”, 25);
```

A **string** is a word or words. Strings are always written with quotation marks.
A **number** is, you know, a number.

Reference: http://www.w3schools.com/js/js_functions.asp

Code Challenge (40 mins)

Think about a time when you use filters, like in Instagram or Facebook, or even Photoshop. When you choose a filter or customize settings with sliders, you are running code that a programmer has written!

Follow the “Make Your Own Filter” tutorial. Read the instructions on the left and do what they say. When you’ve followed all the instructions, click the button to move on to the next part.

Sample Solution:
Students’ solutions will differ. Check that they have the right syntax for each filter they’ve chosen to use.

```
blur(10);
noise(5);
exposure(20);
vignette(30);
tint("purple", 30);
```

Reflection (5 mins)

Who’s a programmer? *You are!*
Does code look like you expected? Is it easy, hard, straightforward, confusing?
What is a function? What functions did we use today?
What is an argument? Point out some of the arguments in your code.
What are two types of arguments you used?
What is the difference between a string and a number? How can you tell them apart by looking?

Stop Motion Animation

Big Idea: Objects, properties, and arrays are ways of holding data.

Module: Make a Stop Motion https://app.vidcode.io/project/stopmotion

Time: 1 hour

10 minutes background
35 minutes coding
10 minutes sharing
5 minutes reflection

Vocabulary

Object, Property, Dot Notation, Array

Standards

CCSS.MATH.PRACTICE.MP2 Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP6 Attend to precision.

NGSS Engineering Practice 5 Using mathematics and computational thinking

Background (10 mins)

More JavaScript basics

Objects are the main way of keeping track of data. Think of an object like a picnic basket.

```
basket
```

Properties are things held inside your object. In our picnic basket, these are the sandwiches and napkins and stuff. We can access the properties using **dot notation**.

```
basket.sandwiches
```

Arrays are lists that hold your data. They are another **type** like **strings** and **numbers**. Arrays always start and end with square brackets, and the items in the array are separated by commas. Here is an array full of strings:

```
[“ham”, “tuna”, “chicken salad”]
```

When we pack our picnic basket with three types of sandwiches, the code would look like this:

```
basket.sandwiches = [“ham”, “tuna”, “chicken salad”];
```

Always remember to end every line of code with a semicolon! That’s how JavaScript knows it’s finished!

Code Challenge (35 mins)

A stop motion animation is an animation created by putting together a series of still pictures together so they seem like they’re moving. This can make it seem like objects are moving on their own.

Follow the “Make a Stop Motion” tutorial. Read the instructions on the left and do what they say. When you’ve followed all the instructions, click the button to move on to the next part.

Optional: Instead of using the provided stills, take pictures with your computer or your phone. Import them using the menu on the left. Example subjects include: cut paper into shapes and move them around, take pictures of a friend in various poses and animate them together to make a dance.

Sample Solution:

```
movie.frames = ["img1.jpg", "img2.jpg", "img3.jpg", "img4.jpg"];
movie.interval = 100;
blur(50);
```

Sharing (10 mins)

Take turns share your stop-motion animations with the class.

Reflection (5 mins)

What is an array? Does the order of an array matter?
What object did we use today? (movie)
What two properties of movie did we use? (frames, interval)
If you were an object, what would be your properties? Can you think of a property that is a number? (age, shoe size) A string? (lastName, favoriteSong) An array? (siblings, booksToRead)
How does JavaScript know it’s the end of a line? ;

6th Grade Ancient Civ Vocabulary

Big Idea: Planning and execution make the difference between a memorable meme and a forgettable one.

Module: Make a Meme https://app.vidcode.io/project/graphics

Time: 1 hour

10 minutes background
20 minutes video production
20 minutes coding
10 minutes sharing

Standards

CCSS.ELA-LITERACY.RH.6-8.4
Determine the meaning of words and phrases as they are used in a text, including vocabulary specific to domains related to history/social studies.
CCSS.ELA-LITERACY.RH.6-8.7
Integrate visual information (e.g., in charts, graphs, photographs, videos, or maps) with other information in print and digital texts.
CCSS.MATH.PRACTICE.MP6 Attend to precision.

CCSS.MATH.PRACTICE.MP7 Look for and make use of structure. CCSS.MATH.PRACTICE.MP8 Look for and express regularity in repeated reasoning.

NGSS Engineering Practice 8 Obtaining, evaluating, and communicating information

Background (10 mins)

A meme is a tiny idea that people like and share because it is funny, weird, or gross.

Ever think about where the word “meme” comes from? It’s related to “memory.” Memes are things that are easy to remember! And what kinds of things do you work hard to remember? Vocab words! In this lesson, we’ll make our vocabulary words easy to remember by turning them into memes.

Watch this video as inspiration: https://www.flocabulary.com/ancient-egypt/

Video Challenge (20 mins)

In teams, choose a vocabulary word and shoot a short video about it.

Irrigation Polytheism Empire Ziggurat Levee City-State

Use the Flocabulary video as inspiration. You don't have to be too literal. Be interesting!

Sample ideas:

Shoot a page in your social studies book.
Make some simple paper puppets and move them around with your hands.
Act out the word in character.
Film yourself writing or typing the definition.
Charades!

Code Challenge (20 mins)

Add text and effects to your video to make it really memorable. Animate a special effect that goes with the word. Add some cool music. Make sure you've conveyed the word and what it means!

Code reference: https://app.vidcode.io/reference

text('I love coding!', 60, 55); *Creates text on your video at position (x,y).*

text.color = "green"; *Changes the color of your text.*

text.size = "50px"; *Changes the size of your text.*

text.font = "Times"; *Changes the font of your text. Possible fonts: "Arial", "Comic Sans MS", "cursive", "serif", "monospace"*

audio("rock"); *Plays music over your video. You can change it to "dance", "electronic", "funk", "rock", or "retro"!*

Sharing (10 mins)

Publish the finished memes, and show them one-by-one on the class projector. Have a face-off: which team did each word better? Vote on the most memorable meme. Turn the videos off and have a pop quiz!

Hour of Code: Climate Facts

The goal of this Hour of Code lesson is for students to research and understand a fact about the Earth's climate, engaging with the work of scientists and artists in response to climate change.

Students should take their research, and plan a 30-second video sharing a fact that they learned. They can use props, art they made, their environment and other actors in their videos.

Big Idea: Coding levels up your presentation skills.

Module: Climate Science & Code https://app.vidcode.io/project/hourofcode-science

Time: 1 hour

25 minutes video production
25 minutes coding
10 minutes reflection

Video Challenge (25 mins)

Before your students start filming, they should plan out what they're going to be creating. Choose a topic: what fact about climate do they want to share?

Have students create a storyboard or write a short script, and use it as their guide. Or, if they want to adlib, have them write a short summary of their fact or topic.

Think about the story: what will your filming environment look like? How many people will you film? What props will you use?

How will you incorporate effects and graphics into your video during editing?

You can choose to record the actual video through the Vidcode interface, or from a program installed on your students' computers, such as PhotoBooth.

Go the the Project Page https://app.vidcode.io/project/hourofcode-science. Students can record their videos directly onto the interface or click the background to exit record, and upload their video with the button on the right.

Code Challenge (25 mins)

Once they're happy with their videos, it's time to start coding! They should follow the steps on the left of the screen to go learn how to edit their videos with code.

This can be done individually, or in pairs.

Use the docs or Reference page for more information on how to edit videos with JavaScript.
https://app.vidcode.io/reference

Reflection (10 mins)

After publishing their videos, students can click 'View your Video here' and share the url of the video with their classmates.

Students should talk about what they learned about climate change, and how they used art and code to create their video.

What did they learn? What is JavaScript? What is creative coding? How can they use these things in the future?

Video Storyboard

BEGINNING

1. Notes: ________________________________ 2. Notes: ________________________________

MIDDLE

3. Notes: ________________________________ 4. Notes: ________________________________

END

5. Notes:________________________________ 6. Notes:________________________________

Project Assessment Rubric

	Unsatisfactory	Competent	Proficient	Distinguished
Project Content	Project does not convey the required information or understanding.	Project shows some understanding of the subject.	Project reflects understanding of the subject.	Project reflects understanding and synthesis of the subject.
Code Execution	Program does not work, or has major flaws that prevent its intended use.	Program mostly works, and has only minor flaws.	Program works in the way the student intended.	Program is functional and refined, with extra features that exceed the requirements.
Code Practice	Program is difficult to read. Code contains lines that do not work or are out of order.	Program can be read and is in a logical order.	Program is well-organized, easy to read and understand.	Program is well-organized, makes good use of whitespace and comments. Variables have helpful names.
Reflection	Student cannot describe how their code works.	Student can mostly describe how their code works.	Student can describe how their code works and can make changes that have desired effects.	Student can describe how their code works and how they wrote it, and help others debug their code.
Habits of mind	Student is not aware of the goal of the program, is frequently off- task, does not offer their own ideas, and gives up when it is difficult.	Student is aware of the goal of the program, returns to the task when asked, has some ideas when prompted, asks for help when stuck.	Student understands the goal of the program, has their own ideas, rarely goes off-task, and attempts to solve problems first before asking for help.	Student embraces the goal of the program and chooses to try out new ideas and multiple solutions, even when they are challenging.