CodeHS

California 9-12 Course Syllabus

1 year for High School (130-140 contact hours)

Course Overview and Goals

CodeHS's California 9-12 teaches the foundations of computer science and basic programming, with an emphasis on helping students develop logical thinking and problem-solving skills. Once students complete this course, they will have learned material equivalent to a semester college introductory course in computer science and be able to program in JavaScript. The course is fully aligned to the California 9-12 Computer Science standards outlined by the California Board of Education.

Learning Environment: The course utilizes a blended classroom approach. The content is fully web-based, with students writing and running code in the browser. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students. Each unit of the course is broken down into lessons. Lessons consist of video tutorials, short quizzes, example programs to explore, and written programming exercises, adding up to over 100 hours of hands-on programming practice in total. Each unit ends with a comprehensive unit test that assesses student's mastery of the material from that unit as well as challenge problems where students can display their understanding of the material.

Programming Environment: Students write and run JavaScript programs in the browser using the CodeHS editor.

More information: Browse the content of this course at https://codehs.com/course/10113

Prerequisites: The California 9-12 course is designed for complete beginners with no previous background in computer science. The course is highly visual, dynamic, and interactive, making it engaging for new coders.

Course Breakdown

Unit 1: Digital Citizenship and Cyber Hygiene (3 weeks/15 hours)

This module includes topics on Internet etiquette and how to stay safe on the world wide web. Students will also look at the potential effects of our digital footprints, how to protect information from online risks, and the implications of cyberbullying. Finally, the module includes how to find and cite quality resources online. Browse the full content of this module at https://codehs.com/library/course/10113/module/15228

Objectives / Topics Covered	- What is Cybersecurity? - The CIA Triad - Digital Footprint and Reputation - Cyberbullying - Internet Safety - Privacy and Security - Information Literacy - Visualizing and Interpreting Data - Data Collection and Limitations - Creative Credit and Copyright

	● Hacking Ethics

Example Assignments / Labs	● Digital Footprint and Reputation
	  ○ What is a digital footprint?
	  ○ What is *your* digital footprint and reputation?
	  ○ What does it mean that the internet is public and permanent?
	  ○ Who looks at your digital footprint and reputation?
	  ○ What are some recommended social media guidelines?
	  ○ How can you maintain your digital footprint?
	  ○ What does your digital footprint say about you?
	  ○ Example activities:
	    ■ What is your digital footprint?
	    ■ Are you going to make any changes in what you post on social media?
	● Cyberbullying
	  ○ What is cyberbullying?
	  ○ What are the impacts of cyberbullying?
	  ○ Are there cyberbullying roles?
	  ○ What do you do if you are being bullied?
	  ○ What do you do if you see bullying?
	  ○ How can you be an upstander?
	  ○ Example activities:
	    ■ Explore cyberbullying scenarios: What would you do?
	● Internet Safety
	  ○ What are some ways to stay safe online?
	  ○ What are some online safety guidelines?
	  ○ Example activities:
	    ■ Explore Internet safety scenarios: What would you do?
	● Privacy and Security
	  ○ What are data privacy and security?
	  ○ How can you keep personal data secure and private?
	  ○ What can happen if your data is stolen and what can you do about it?
	  ○ Example activities:
	    ■ Test out various passwords on a site
	    ■ Explore Google's privacy policy: What do they know about you?
	● Information Literacy
	  ○ What is information literacy?
	  ○ How can you do effective internet searches?
	  ○ What are some techniques for judging source legitimacy and identifying misinformation?
	  ○ Example activities:
	    ■ Create and test search queries
	    ■ Explore evidence for using sources
	● Creative Credit and Copyright
	  ○ What is copyright?
	  ○ What are the different types of copyright licenses
	  ○ Example activities:
	    ■ Create citations for sources
	    ■ Explore image search tools
	● Hacking Ethics
	  ○ What are hackers?

	○ Are there different kinds of hackers? (white, black, grey)
	○ What are bug bounty programs?
	○ Is hacking always illegal?
	○ What are the consequences of illegal hacking?
	○ Example activities:
	   ■ Explore what penetration testing is
	   ■ Sign ethical hacker agreement
	● Final project: Create a Public Service Announcement
	○ Create a Public Service Announcement (PSA) to teach your peers about your selected topic in digital citizenship and cyber hygiene. You can select any of the topics covered in this module. Be creative and make it fun! You could make a video, song, poster, or slideshow.

Module 2: The ABCs of Cryptography (1-2 weeks/5-10 hours)

In this module, students will dive into the history of cryptography systems, the motivation behind using encryption systems, and basic cryptography systems. Additionally, they will explore topics on how to use cryptography, cryptology, and cryptanalysis to decode a message without the use of a key.

Browse the full content of this module at https://codehs.com/library/course/10113/module/11945

Objectives / Topics Covered	● Cryptography, Cryptology, Cryptanalysis ● History of Cryptography ● Why do we Need to Encrypt Data? ● Basic Cryptography Systems: Caesar Cipher ● Basic Cryptography Systems: Cracking the Caesar Cipher ● Basic Cryptography Systems: Vigenère Cipher

Example Assignments / Labs	● Cryptography, Cryptology, Cryptanalysis
	○ Why do we need some secrecy in our transparent information age?
	○ Explain general encryption with data, keys
	○ Example activities:
	   ■ Video and discussion on securing the cloud
	   ■ Passing notes in class (offline activity)
	● History of Cryptography
	○ Why do we encrypt?
	○ What are some classic encryption techniques?
	○ What is the flaw in substitution ciphers?
	○ What was The Enigma during WW2?
	○ What is modern cryptography and how has cryptography changed over time?
	○ What is 256-bit key encryption and how does this help cryptography overall?
	○ Example activities:
	   ■ How did the Enigma work?
	● Why do we Need to Encrypt Data?
	○ Explore the CIA Triad and encryption
	○ Example activities:
	   ■ Telephone game with math (offline)
	   ■ Modulo math activity sheet
	● Basic Cryptography Systems: Caesar Cipher
	○ Explore examples of the Caesar cipher

	○ Example activities:
	■ Practice with a Caesar Cipher JavaScript program
	■ Modify the program to create the decrypting Caesar program
	● Basic Cryptography Systems: Cracking the Caesar Cipher
	○ How do we solve the Caesar Cipher with brute force and using letter frequency analysis?
	○ Example activities:
	■ Practice cracking Caesar Cipher with brute force
	■ Practice cracking Caesar Cipher with letter frequency
	● Basic Cryptography Systems: Vigenère Cipher
	○ Explore examples of the Vigenère Cipher
	○ Example activities:
	■ Practice with a Vigenère Cipher JavaScript program

Unit 3: Introduction to Programming in JavaScript with Karel the Dog (4 weeks/20 hours)

Module three is programming with Karel. Karel is a dog that only knows how to move, turn left, and place tennis balls in his world. Students can give Karel these commands to instruct him to do certain things. We use Karel to show students what it means to program, and allow them to focus on problem solving.

Browse the full content of this unit at https://codehs.com/library/course/10113/module/15229

Objectives / Topics Covered	● Commands
	● Defining vs. Calling Methods
	● Designing methods
	● Program entry points
	● Control flow
	● Looping
	● Conditionals
	● Classes
	● Commenting code
	● Preconditions and Postconditions
	● Top Down Design
Assignments / Labs	● 26 Karel Programming Exercises and Challenges in total
	● Program-specific tasks for Karel the Dog
	○ Example Exercise: Pyramid of Karel
	Write a program to have Karel build a pyramid. There should be three balls on the first row, two in the second row, and one in the third row.
	● Teach Karel new commands like `turnRight()` or `makePancakes()`
	○ Example Exercise: Pancakes
	Karel is the waiter. He needs to deliver a stack of pancakes to the guests on the 2nd, 4th, and 6th avenue. Each stack of pancakes should have three pancakes.
	Create a method called `makePancakes()` to help Karel solve this problem.
	● Solve large Karel problems by breaking them down into smaller, more manageable problems using Top Down Design
	○ Example Exercise: The Two Towers
	In this program, Karel should build two towers of tennis balls. Each tower should be 3 tennis balls high.
	At the end, Karel should end up on top of the second tower, facing East.
	● Using control structures and conditionals to solve general problems
	○ Example Exercise: Random Hurdles

	Write a program that has Karel run to the other side of first street, jumping over all of the hurdles. However, the hurdles can be in random locations. The world is fourteen avenues long.

Unit 4: Karel Challenges (1.5 weeks, 7 hours)

In this module students take all the foundational concepts from Karel to solve some programming challenges.
Browse the full content of this unit at https://codehs.com/library/course/10113/module/15230

Objectives / Topics Covered	● Solving large and more complex problems using Karel
Assignments / Labs	● 5 Karel challenges to tie everything learned in the Karel module together 　○ Example Exercise: Super Cleanup Karel 　Karel's world is a complete mess. There are tennis balls all over the place, and you need to clean them up. Karel will start in the bottom left corner of the world facing east, and should clean up all of the tennis balls in the world. This program should be general enough to work on any size world with tennis balls in any location.

Unit 5: Networks and the Internet (1.5 weeks/8 hours)

This unit explores the structure and design of the internet, and how this design affects the reliability of network communication, the security of data, and personal privacy.
Browse the full content of this unit at https://codehs.com/library/course/10113/module/15231

Objectives / Topics Covered	● Structure of the internet ● How network data is transmitted ● Hardware involved in the transmission of data ● How the internet has impacted everyday life
Example Assignments / Labs	● Structure of the internet 　○ Explore the differences between IPv4 and IPv6. Why are we running out of addresses? 　○ Explore the different levels of the internet. 　○ Example Activity 　　■ Trace a website request from the server, through the network, and to your computer ● How data is transmitted 　○ How are internet packets able to find their way to your computer? 　○ Explain in your own words how a request from your computer travels through the various levels of servers to reach and return the correct webpage and resources? 　○ Example Activity: 　　■ As a class, create a protocol that will allow one classmate to send another classmate a note, without the need for talking to each other. ● Hardware involved 　○ Explore the role of routers 　○ Why are protocols so important? 　○ Explore how data is able to be transmitted across the ocean by using underwater cables

Unit 6: Javascript & Graphics (1.5 weeks/7 hours)

Introduces students to the basics of JavaScript, including variables, user input, mathematics, basic graphics, and image representations.

Browse the full content of this unit at https://codehs.com/library/course/10113/module/15232

Objectives / Topics Covered	● Variables ● User Input ● Arithmetic Expressions ● Graphics ● Image Representation

Assignments / Labs	● 7 JavaScript & graphics programming exercises in total
	● Using variables and getting user input using JavaScript
	   ○ Example Exercise: Dinner Plans
	     Prompt the user for their name, then ask them what time you should meet for dinner.
	     Greet them by name and tell them you will meet them at the time they specified!
	● Create basic graphics in JavaScript
	   ○ Example Exercise: Flag of the Netherlands
	     This program should draw the flag for the Netherlands. The top third of the canvas is red, the middle third is white, and the bottom third is blue, as shown below. You will need to use Rectangle objects in this program.

Unit 7: Graphics Challenges (1 week, 5 hours)

In this module students learn how to pair program! Students will also take all the foundational concepts from JavaScript Graphics to solve some programming challenges.

Browse the full content of this unit at https://codehs.com/library/course/10113/module/15233

Objectives / Topics Covered	● Pair programming ● Solving large and more complex problems that incorporate graphics and user input

Assignments / Labs	● 3 graphics challenges to tie everything learned in the JavaScript & Graphics module together
	   ○ Example Exercise: Ghost
	     Write a program to draw a ghost on the screen. You must do this by using the constant values given (this will allow us to easily alter the size or color of the ghost.)

Unit 8: JavaScript Control Structures (3 weeks/15 hours)

Students learn how to use control structures such as if/else statements and loops to make more advanced programs in JavaScript.

Browse the full content of this unit at https://codehs.com/library/course/10113/module/15234

Objectives / Topics Covered	● Booleans ● For Loops ● Conditionals ● Nested Control Structures ● While Loops
Assignments / Labs	● 22 control structures programming exercises in total ● Using comparison and logical operators to control the flow of the program

	○ **Example Exercise: Inventory** Write a program that keeps track of a simple inventory for a store. While there are still items left in the inventory, ask the user how many items they would like to buy. Then print out how many are left in inventory after the purchase. You should use a while loop for this problem. Make sure you catch the case where the user tries to buy more items than there are in the inventory. In that case, you should print a message to the user saying that their request isn't possible. ● Using for loops 　○ **Example Exercise: All Dice Values** 　Write a program that prints all possible dice rolls with 2 dice To do so, you should use a double for loop. ● Drawing basic graphics using control structures 　○ **Example Exercise: Caterpillar** 　This graphics program should draw a caterpillar. A caterpillar has NUM_CIRCLES circles. Every other circle is a different color, the even circles are red, and the odd circles are green (by even we mean when i is an even number). Use a for loop to draw the caterpillar, centered vertically in the screen. Also, be sure that the caterpillar is still drawn across the whole canvas even if the value of NUM_CIRCLES is changed.

Unit 9: Control Structures Challenges (1 week, 5 hours)

In this module students take all the foundational concepts from the Control Structures unit to solve some programming challenges.

Browse the full content of this unit at https://codehs.com/library/course/10113/module/15235

Objectives / Topics Covered	● Solving large and more complex problems that incorporate control structures
Assignments / Labs	● 4 challenges using control structures to tie everything learned in the JavaScript Control Structures module together 　○ **Example Exercise: Circles in Squares** 　Repeatedly draw a circle inside of a square and then a square inside that circle and so on. You should repeat this while the size is greater than `MIN_SIZE`. Each time you draw a shape, it should be centered on the screen and a new random color. You'll need to update the size based on the size of the previous one.

Unit 10: Functions, Events, and Lists (5 weeks, 25 hours)

Students learn to write reusable code with functions and parameters.

Browse the full content of this unit at https://codehs.com/library/course/10113/module/15236

Objectives / Topics Covered	● Functions with and without parameters ● Functions with and without return values ● Nested Control Structures ● Local variables and scope ● Timers ● Randomizing Games ● Mouse Events ● Keyboard Events ● Simulations
Assignments / Labs	● 32 functions programming exercises in total

○ Example Exercise: Vertical Lines
Write a function that draws vertical lines on the graphics canvas. If a line is vertical, then the x-values for the endpoints are the same.
The parameters to your function should be the x location, and the length, and all of your lines should start at y position 0.
○ Example Exercise: Is it even?
Write a function called isEven that returns a boolean of whether or not a value is even or odd. The isEven function should not print anything out or return a number. It should only take in a number and return a boolean. Once you've written this function, write a program that asks the user for integers and prints whether the number they entered is even or odd using your isEven function. You should let the user keep entering numbers until they enter the SENTINEL given.
○ Using timers to add randomizations to graphical programs
■ Example Exercise: Paint Splatter
Write a program that splatters paint on the screen every DELAY milliseconds.
To splatter paint, pick a random color and draw CIRCLES_PER_SPLATTER circles of that color at random places on the screen. The radius of each circle should be a random value between MIN_RADIUS and MAX_RADIUS.
Remember to use helper functions.
○ Using mouse events for interactive programs
■ Example Exercise: Teleporting Ball
Extend our bouncing ball program. Whenever you click, the ball should teleport to that spot and change to a random color.
■ Example Exercise: Target
Draw a target on the screen that moves to aim at where your mouse is located.
A target consists of a horizontal line that goes from 0 to the window width and a vertical line that goes from 0 to the window height. The lines should cross paths where the mouse is.

Unit 11: Final Project (2-4 weeks, 10-20 hours)

In this module students take all the skills they've learned throughout the course to create a final program with a partner!

Browse the full content of this unit at https://codehs.com/library/course/10113/module/15237

Objectives / Topics Covered	● Project Planning ● Final Project Creation

Assignments / Labs	● Collaborative open-ended final project which encourages creativity
	● Program Requirements:
	Your program:
	   ○ must use JavaScript Graphics
	   ○ must allow the user to interact with your project with either their mouse or keyboard
	   ○ must use at least one timer
	   ○ must break down the program into multiple functions
	   ○ must utilize control structures where applicable

Optional Supplemental Materials (Remainder of school year)

Objectives / Topics Covered	- Extra practice with: - Karel - Basic JavaScript - JavaScript functions - Graphics - Animation - Basic Data Structures in JavaScript - Game Design - Music Visualization
Assignments / Labs	Several additional exercises and large projects covering the topics listed above