



# CodeHS

## Texas Computer Science I Course Syllabus

1 year for High School (145-155 contact hours)

### Course Overview and Goals

The CodeHS Computer Science I curriculum 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 the CodeHS Computer Science I course, they will have learned material equivalent to a semester college introductory course in Computer Science and be able to program in JavaScript.

**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/6229>

**Prerequisites:** The Computer Science I 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: What is Computing? (5 weeks/25 hours)

Objectives / Topics Covered	<ul style="list-style-type: none"><li>● History of computers</li><li>● What is a computer?</li><li>● What is software?</li><li>● What is hardware?</li><li>● Future of computing</li></ul>
Example Assignments / Labs	<ul style="list-style-type: none"><li>● History<ul style="list-style-type: none"><li>○ Find out when the first computers were created</li><li>○ Research famous computer innovators</li><li>○ What roles do computers play in your life?</li><li>○ Example Activity:<ul style="list-style-type: none"><li>■ Summarize an era of advances in computers</li></ul></li></ul></li><li>● What is a computer?<ul style="list-style-type: none"><li>○ What parts do modern computers have?</li><li>○ What are input devices?</li></ul></li></ul>

	<ul style="list-style-type: none"> <li>○ What are output devices?</li> <li>○ Example Activity: <ul style="list-style-type: none"> <li>■ Draw a computer and label all of its parts, including the input devices and output devices</li> </ul> </li> <li>● Software/Hardware <ul style="list-style-type: none"> <li>○ What's the difference?</li> <li>○ What hardware components make up a computer?</li> <li>○ What is software used for?</li> <li>○ Example Activity: <ul style="list-style-type: none"> <li>■ Label the parts of your computer</li> </ul> </li> </ul> </li> <li>● Future of Computing <ul style="list-style-type: none"> <li>○ Research uses of Artificial Intelligence in use now</li> <li>○ Research new ways of storing data</li> <li>○ Example Class Activity: <ul style="list-style-type: none"> <li>■ In what ways can we use technology that we couldn't 10 years ago. Are these technological advances helpful or harmful overall?</li> </ul> </li> </ul> </li> <li>● Final Project <ul style="list-style-type: none"> <li>○ Create a presentation about a computer. Choose any computer -- a phone, an early computer model, drones, etc. Who built it and why? How does it interact with people? How do people interact with the computer?</li> </ul> </li> </ul>
--	---

## Unit 2: Introduction to Programming in JavaScript with Karel the Dog (3 weeks/15 hours)

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

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Commands</li> <li>● Defining vs. Calling Methods</li> <li>● Designing methods</li> <li>● Program entry points</li> <li>● Control flow</li> <li>● Looping</li> <li>● Conditionals</li> <li>● Classes</li> <li>● Commenting code</li> <li>● Preconditions and Postconditions</li> <li>● Top Down Design</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● 26 Karel Programming Exercises and Challenges in total</li> <li>● Program-specific tasks for Karel the Dog <ul style="list-style-type: none"> <li>○ 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.</li> </ul> </li> <li>● Teach Karel new commands like <code>turnRight()</code> or <code>makePancakes()</code> <ul style="list-style-type: none"> <li>○ 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 <code>makePancakes()</code> to help Karel solve this problem.</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>● Solve large Karel problems by breaking them down into smaller, more manageable problems using Top Down Design <ul style="list-style-type: none"> <li>○ 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.</li> </ul> </li> <li>● Using control structures and conditionals to solve general problems <ul style="list-style-type: none"> <li>○ 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.</li> </ul> </li> </ul>
--	---

### Unit 3: Karel Challenges (1.5 weeks, 7 hours)

In this module you'll take all the foundational concepts from Karel to solve some programming challenges.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Solving large and more complex problems using Karel</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● 5 Karel challenges to tie everything learned in the Karel module together <ul style="list-style-type: none"> <li>○ 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 locations.</li> </ul> </li> </ul>

### Unit 4: Digital Citizenship and Cyber Hygiene (7 weeks, 35 hours)

This module includes topics on Internet etiquette and how to stay safe on the world wide web. We will also look at the potential effects of our digital footprints, how to protect information from online risks, and the implications of hacking. Finally, the module includes how to find and cite quality resources online.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Digital Footprint and Reputation</li> <li>● Privacy and Security</li> <li>● Information Literacy</li> <li>● Creative Credit and Copyright</li> <li>● Hacking Ethics</li> </ul>
Example Assignments / Labs	<ul style="list-style-type: none"> <li>● Digital Footprint and Reputation <ul style="list-style-type: none"> <li>○ Example activities: <ul style="list-style-type: none"> <li>■ What is your digital footprint?</li> <li>■ Are you going to make any changes in what you post on social media?</li> </ul> </li> </ul> </li> <li>● Keeping data private and secure <ul style="list-style-type: none"> <li>○ Example activities: <ul style="list-style-type: none"> <li>■ Test out various passwords on a site</li> <li>■ Explore Google's privacy policy: What do they know about you?</li> </ul> </li> </ul> </li> <li>● Information Literacy <ul style="list-style-type: none"> <li>○ Example activities: <ul style="list-style-type: none"> <li>■ Create and test search queries</li> <li>■ Explore evidence for using sources</li> </ul> </li> </ul> </li> <li>● Different types of copyright licenses</li> </ul>

	<ul style="list-style-type: none"> <li>○ Example activities: <ul style="list-style-type: none"> <li>■ Create citations for sources</li> <li>■ Explore image search tools</li> </ul> </li> <li>● Hacking Ethics <ul style="list-style-type: none"> <li>○ Example activities: <ul style="list-style-type: none"> <li>■ Explore what penetration testing is</li> <li>■ Sign ethical hacker agreement</li> </ul> </li> </ul> </li> </ul>
--	--

### Unit 5: Javascript & Graphics (1.5 weeks/7 hours)

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

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Variables</li> <li>● User Input</li> <li>● Arithmetic Expressions</li> <li>● Graphics</li> <li>● Image Representation</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● 7 JavaScript &amp; graphics programming exercises in total</li> <li>● Using variables and getting user input using JavaScript <ul style="list-style-type: none"> <li>○ 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!</li> </ul> </li> <li>● Create basic graphics in JavaScript <ul style="list-style-type: none"> <li>○ 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.</li> </ul> </li> </ul>

### Unit 6: JavaScript Control Structures (3 weeks/15 hours)

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

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Booleans</li> <li>● For Loops</li> <li>● Conditionals</li> <li>● Nested Control Structures</li> <li>● While Loops</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● 22 control structures programming exercises in total</li> <li>● Using comparison and logical operators to control the flow of the program <ul style="list-style-type: none"> <li>○ 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.</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>● Using for loops <ul style="list-style-type: none"> <li>○ 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.</li> </ul> </li> <li>● Drawing basic graphics using control structures <ul style="list-style-type: none"> <li>○ 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.</li> </ul> </li> </ul>
--	---

### Unit 7: Control Structures Challenges (1 week, 5 hours)

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

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Solving large and more complex problems that incorporate control structures</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● 4 challenges using control structures to tie everything learned in the JavaScript Control Structures module together <ul style="list-style-type: none"> <li>○ 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.</li> </ul> </li> </ul>

### Unit 8: Functions and Parameters (2 weeks, 10 hours)

Learn to write reusable code with functions and parameters.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Functions with and without parameters</li> <li>● Functions with and without return values</li> <li>● Nested Control Structures</li> <li>● Local variables and scope</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● 14 functions programming exercises in total <ul style="list-style-type: none"> <li>○ 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.</li> <li>○ 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.</li> </ul> </li> </ul>

### Unit 9: Animation and Games (3 weeks, 15 hours)

Now, your graphics will come to life. You will learn how to make objects move around the screen. You will also learn how to let the user interact with your program with the mouse. At the end of this section, you will program your very own video game.

Objectives / Topics Covered	<ul style="list-style-type: none"><li>● Timers</li><li>● Randomizing Games</li><li>● Mouse Events</li><li>● Keyboard Events</li></ul>
Assignments / Labs	<ul style="list-style-type: none"><li>● 15 animations programming exercises in total</li><li>● Using timers to add randomizations to graphical programs<ul style="list-style-type: none"><li>○ 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.</li></ul></li><li>● Using mouse events for interactive programs<ul style="list-style-type: none"><li>○ 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.</li><li>○ 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.</li></ul></li><li>● Using keyboard events for interactive programs<ul style="list-style-type: none"><li>○ Example Exercise: Basic Snake Write a basic version of the snake game. The way our game works is by first creating a green square at the center of the screen. The snake should be moving to the right. If you hit an arrow key, you should change the snake's direction.</li></ul></li></ul>

### Unit 10: Project: Breakout (2 weeks, 10 hours)

Objectives / Topics Covered	<ul style="list-style-type: none"><li>● Basic graphics</li><li>● Mouse events</li><li>● Collision detection</li></ul>
Assignments / Labs	<ul style="list-style-type: none"><li>● Guided exercises to build a Breakout Game</li><li>● The Breakout Game is made up of bricks at the top of the screen, a paddle that you control at the bottom of the screen, and a ball that bounces around. Your goal is to direct the paddle with your mouse to bounce the ball until all of the bricks have been hit and disappear.</li></ul>

### Unit 12: Basic Data Structures (6 weeks/30 hours)

Objectives / Topics Covered	<ul style="list-style-type: none"><li>● List/Array creation and basic operations</li><li>● Iterating through lists/arrays</li></ul>
-----------------------------	---

	<ul style="list-style-type: none"> <li>● Finding and removing elements in lists/arrays</li> <li>● Object/Map basics</li> <li>● Iterating over an object/map</li> <li>● Set creation and basic operations</li> <li>● Grid creation and basic operations</li> </ul>
<p>Assignments / Labs</p>	<ul style="list-style-type: none"> <li>● 24 exercises in total</li> <li>● Basic list/array operations <ul style="list-style-type: none"> <li>○ Example Exercise: List of Places to Travel Create an array of the top 5 places you would like to travel called <code>travellist</code>. Print out the item at index 2.</li> </ul> </li> <li>● Iterating through arrays/lists <ul style="list-style-type: none"> <li>○ Example Exercise: Draw a Barcode In this program, you will draw a barcode on the screen given an array that represents the data in the barcode. The array will contain a boolean in it, and if the boolean is true, you will draw a vertical line in that position that runs from the top to the bottom of the screen. If not, you will not draw a line. We have written the <code>generateBarcode</code> function for you that creates a random barcode. Your job is to write the <code>drawBarcode</code> function.</li> </ul> </li> <li>● Basic list/array algorithms <ul style="list-style-type: none"> <li>○ Example Exercise: Remove From Line You are given an array of names of people who are in line for movie tickets. Use the <code>removeElement</code> to remove the first person from the line twice, as if you have just given them their tickets. You should write a function to print everyone in line. Then print the line before and after removing the people.</li> </ul> </li> <li>● Basics of objects/maps <ul style="list-style-type: none"> <li>○ Example Exercise: Sidekicks Police Commissioner Gordon has tasked you with building up a database of superhero sidekicks, just in case the superheroes are all busy. Given a superhero name, we need to be able to look up the name of that superhero's sidekick. We've already started the database for you, but you need to add to it. Add an entry to the <code>sidekicks</code> Object for Batman's sidekick Robin. Get the name of Mermaid Man's sidekick and print it out. Who is your sidekick? Add their name to the <code>sidekicks</code> Object as your sidekick</li> </ul> </li> <li>● Basics of sets <ul style="list-style-type: none"> <li>○ Example Exercise: Mutual Friends Write a program that prints the mutual friends between two people. You should create two sets, one for each person, and add friends (strings) to each set representing the friends of that person. Then, fill in the <code>mutualFriends</code> function that takes the two people as parameters and returns a new set that includes their mutual friends. Print out the set of mutual friends</li> </ul> </li> <li>● Iterating through a Grid <ul style="list-style-type: none"> <li>○ Example Exercise: Summing Grid Write a function called <code>sumGrid(grid)</code> that takes a grid as a parameter and fills each location in the grid</li> </ul> </li> </ul>

	with the sum of the row index and column index of that location.
--	--

**Unit 13: Final Project (2-4 weeks, 10-20 hours)**

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

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Project Planning</li> <li>● Final Project Creation</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● Collaborative open-ended final project which encourages creativity</li> <li>● Program Requirements: Your program: <ul style="list-style-type: none"> <li>○ must use JavaScript Graphics</li> <li>○ must allow the user to interact with your project with either their mouse or keyboard</li> <li>○ must use at least one timer</li> <li>○ must break down the program into multiple functions</li> <li>○ must utilize control structures where applicable</li> </ul> </li> </ul>

**Optional Supplemental Materials (Remainder of school year)**

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Extra practice with: <ul style="list-style-type: none"> <li>○ Karel</li> <li>○ Basic JavaScript</li> <li>○ JavaScript functions</li> <li>○ Graphics</li> <li>○ Animation</li> </ul> </li> <li>● Basic Data Structures in JavaScript</li> <li>● Game Design</li> <li>● Music Visualization</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● Several additional exercises and large projects covering the topics listed above</li> </ul>