



# CodeHS

## Texas Computer Science I Course Syllabus

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

### Course Overview and Goals

The CodeHS Texas Computer Science I curriculum fosters students' creativity and innovation by presenting opportunities to design, implement, and present meaningful programs through a variety of media. Through computational thinking and data analysis, students identify task requirements, plan search strategies, and use computer science concepts to access, analyze, and evaluate information needed to solve problems. Students learn digital citizenship by researching current laws, regulations, and best practices and by practicing integrity and respect. Students gain an understanding of the principles of computer science through the study of technology operations, systems, and concepts.

**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/21296>

**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)

Students learn about the history of computing, and about the various parts that make up modern computers. Students also consider the impact computing has had on today's world, and the impacts computing could potentially have in the future.

Objectives / Topics Covered	<ul style="list-style-type: none"><li>● Digital information</li><li>● Number systems</li><li>● What is a computer?</li><li>● What is software?</li><li>● What is hardware?</li><li>● Software licenses</li><li>● Future of computing</li></ul>
-----------------------------	--

<p>Example Assignments / Labs</p>	<ul style="list-style-type: none"> <li>● Encoding data <ul style="list-style-type: none"> <li>○ Create your own encoding scheme</li> <li>○ Encode images using binary</li> <li>○ Example Activity: <ul style="list-style-type: none"> <li>■ Write a message by encoding the characters in binary, using the ASCII codes.</li> </ul> </li> </ul> </li> <li>● Using different number systems <ul style="list-style-type: none"> <li>○ Convert numbers between decimal, binary, and hexadecimal</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> <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> </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> </ul>
-----------------------------------	--

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

Students learn the basics of programming by giving Karel the Dog commands in a grid world.

<p>Objectives / Topics Covered</p>	<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>● Commenting code</li> <li>● Top down design</li> <li>● Debugging strategies</li> </ul>
<p>Example Assignments / Labs</p>	<ul style="list-style-type: none"> <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</li> </ul> </li> </ul>

	<p>guests on the 2nd, 4th, and 6th avenue. Each stack of pancakes should have three pancakes.</p> <p>Create a method called <code>makePancakes()</code> to help Karel solve this problem.</p> <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)

Students apply all the foundational concepts from Intro to Karel to solve new challenges.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Solving large and more complex problems using Karel</li> </ul>
Example Assignments / Labs	<ul style="list-style-type: none"> <li>● 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)

Students learn about Internet etiquette and how to stay safe on the world wide web. They also look at the potential effects of their digital footprints, how to protect information from online risks, and the implications of cyberbullying. Finally, students learn 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> </ul>
Example 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</li> </ul>

	<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> <li>● Different types of copyright licenses <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> </ul> </li> </ul>
--	---

### Unit 5: JavaScript Basics (1 week/5 hours)

Students learn the basics of JavaScript including variables, user input, mathematics, and functions.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Variables</li> <li>● User Input</li> <li>● Arithmetic Expressions</li> <li>● Constants</li> <li>● Collaborative Programming</li> <li>● Random Numbers</li> <li>● Functions</li> </ul>
Example Assignments / Labs	<ul style="list-style-type: none"> <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> </ul>

### Unit 6: The Canvas and Graphics (1 week/5 hours)

Students learn how to add graphics objects and position them on the canvas.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● JavaScript Canvas</li> <li>● JavaScript Graphics</li> <li>● Positioning Graphics Objects</li> </ul>
Example Assignments / Labs	<ul style="list-style-type: none"> <li>● Example Exercise: Create Your Own Meme In this exercise, you are going to create your own meme! The only requirements are that you add at least one image and one text element.</li> </ul>

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

Students apply what they have learned about graphics and basic JavaScript to complete a set of challenges.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Solving large and more complex problems using graphics</li> </ul>
Example Assignments / Labs	<ul style="list-style-type: none"> <li>● Graphics challenges to tie everything learned in the The Canvas and Graphics module <ul style="list-style-type: none"> <li>○ 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.)</li> </ul> </li> </ul>

## 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 advanced programs in JavaScript.

Objectives / Topics Covered	<ul style="list-style-type: none"><li>● Booleans</li><li>● If/Else Statements</li><li>● Logical Operators</li><li>● Comparison Operators</li><li>● Conditionals</li><li>● While Loops</li><li>● Break Statements</li><li>● For Loops</li><li>● Nested Control Structures</li></ul>
Example Assignments / Labs	<ul style="list-style-type: none"><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><li>● Using for loops<ul style="list-style-type: none"><li>○ Example Exercise: Jukebox<ul style="list-style-type: none"><li>■ In the days before the internet, many restaurants would have a jukebox that allowed customers to choose what music they wanted to play. Customers would enter a coin and choose from the jukebox's music collection by selecting a song's number. You could choose one song per coin. In this exercise, you will create a digital jukebox where the user can enter any number of quarters to create a playlist of songs.</li></ul></li></ul></li><li>● Drawing basic graphics using JavaScript<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 on 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 9: Control Structures Challenges (1 week, 5 hours)

Students apply the foundational concepts from the Control Structures module to solve new challenges.

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

	<p>The computer picks a number between 1 and 100, and you have to guess it. The computer will tell you whether your guess was too high, too low, or correct. Your assignment is to generate a random number and let the user guess numbers until they guess the correct number. Make sure to let the user know what they should do at the beginning of the program!</p>
--	---

### Unit 10: Functions (2 weeks/10 hours)

Students learn to write reusable code with functions, parameters, and return values, and explore the impact of variable scopes.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Parameters</li> <li>● Return Values</li> <li>● Default Parameters</li> <li>● Scope</li> </ul>
Example Assignments / Labs	<ul style="list-style-type: none"> <li>● Using various kinds of functions such as functions with and without parameters, and functions with and without return values <ul style="list-style-type: none"> <li>○ Example Exercise: Is it even? Write a program that continually asks the user for integers and then prints whether their input is even or odd. The user should keep entering numbers until they enter 0; at that point, print "Done!" on its own line.</li> </ul> <p>In order to check if the inputted integer is even or odd, you should define and call a function named <code>isEven()</code>. This function should return a Boolean value of <code>true</code> or <code>false</code> depending if the number is even or not.</p> </li> </ul>

### Unit 11: Animation and Games (3 weeks/15 hours)

Students learn how to make objects move around the screen and let users interact using the mouse!

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>
Example Assignments / Labs	<ul style="list-style-type: none"> <li>● Throughout the lessons in this module, you will be developing a simple game that incorporates basic animation techniques and input events.</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: Target Draw a target on the screen that moves to aim at where your mouse</li> </ul> </li> </ul>

	<p>is located.</p> <p>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.</p> <ul style="list-style-type: none"> <li>● Using keyboard events for interactive programs <ul style="list-style-type: none"> <li>○ Example Exercise: Basic Snake</li> </ul> <p>Write a basic version of the snake game.</p> <p>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.</p> </li> </ul>
--	---

### Unit 12: Project: Breakout (2 weeks/10 hours)

Students learn how to make their own Breakout game from scratch using JavaScript.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Basic graphics</li> <li>● Mouse events</li> <li>● Collision detection</li> </ul>
Example Assignments / Labs	<ul style="list-style-type: none"> <li>● Guided exercises to build a Breakout Game</li> <li>● Breakout 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 13: Basic Data Structures (3 weeks/10-15 hours)

Students learn about arrays, adding/removing from them and iterating through them, their methods, and string manipulation.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Array creation and basic operations</li> <li>● Iterating through arrays</li> <li>● Array methods</li> <li>● String manipulation</li> </ul>
Example Assignments / Labs	<ul style="list-style-type: none"> <li>● Basic array operations <ul style="list-style-type: none"> <li>○ Example Exercise: List of Places to Travel</li> </ul> <p>Create an array of the top 5 places you would like to travel called <code>travelList</code>. Print out the item at index 2.</p> </li> <li>● Iterating through arrays <ul style="list-style-type: none"> <li>○ Example Exercise: Draw a Barcode</li> </ul> <p>In this program, you will draw a barcode on the screen given an array that represents the data in the barcode.</p> </li> <li>● Array methods <ul style="list-style-type: none"> <li>○ Example Exercise: Mutual Friends</li> </ul> <p>In this program, you are going to create function that will find the mutual friends between two array lists of friends.</p> </li> </ul>

### Unit 14: Final Project (2-4 weeks/10-20 hours)

Students learn about what makes an engaging and accessible user interface, and will employ an iterative design process including rapid prototyping and user testing to design and develop their own engaging projects.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Collaborative Programming</li> <li>● Project Planning</li> <li>● Pseudocode</li> <li>● Prototype</li> <li>● Testing</li> </ul>
Example 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 utilize mouse interaction from the user</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>

**Unit 15: Computer Science Careers (2-4 weeks/10-20 hours)**

Students learn about a variety of computer science careers and organizations, and what the next steps could look like for them if interested.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Careers and internships</li> <li>● CS career preparation</li> <li>● Legal and ethical responsibilities</li> <li>● Workplace readiness</li> </ul>
Example Assignments / Labs	<ul style="list-style-type: none"> <li>● Exploring computer science careers, internships, and organizations</li> <li>● Learning about CS resumes and certifications</li> <li>● Researching about a major ethical or legal topic in CS</li> <li>● Reflecting on what it means to be a leader and the skills required to be successful in the workplace</li> </ul>

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

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Midterm</li> <li>● Final Exam (JS topics)</li> <li>● Extra practice with Karel and JavaScript topics</li> </ul>
-----------------------------	--