CodeHS

New York Computer Science 9-12

1 year for High School (110-120 contact hours)

Course Overview and Goals

New York Computer Science 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.

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 supplemental 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. For micro:bit, students write and run programs in JavaScript in the browser using the MakeCode editor and will download their programs to their micro:bit devices for further testing.

More information: Browse the content of this course at https://codehs.com/course/19963/overview

Prerequisites: The New York Computer Science 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

Module 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. We 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/course/19963/explore/module/27397

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

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: Programming with Karel (3 weeks/15 hours)

In this module, students learn about 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.

Browse the full content of this unit at https://codehs.com/course/19963/explore/module/27369

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	● 27 Karel programming exercises 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 columns. Each stack of pancakes should have three pancakes. Create a function called `makePancakes()` to help Karel solve this problem. The world should end up exactly as shown here. ○ ● 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 the first row, jumping over all of the hurdles. Karel should only jump if there is a hurdle blocking the way. However, the hurdles can be in random locations. The world is fourteen columns long. You must write a function named `jumpHurdle()` as part of your solution.

Module 3: 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/course/19963/explore/module/27398

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
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

Module 4: JavaScript Basics (1 week/5 hours)

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

Browse the full content of this unit at https://codehs.com/course/19963/explore/module/27371

Objectives / Topics Covered	● Variables ● User Input ● Arithmetic Expressions ● Constants ● Collaborative Programming ● Random Numbers

	• Functions

Assignments / Labs	• 12 JavaScript 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!

Module 5: The Canvas and Graphics (1 week/5 hours)
Students learn how to add graphics objects and position them on the canvas.

Browse the full content of this unit at https://codehs.com/course/19963/explore/module/27372

Objectives / Topics Covered	• JavaScript Canvas • JavaScript Graphics • Positioning Graphics Objects • Mouse Events • Keyboard Events

Assignments / Labs	• 7 JavaScript programming and graphics exercises in total
	  ○ 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.

Module 6: 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.

Browse the full content of this unit at https://codehs.com/course/19963/explore/module/27374

Objectives / Topics Covered	• Booleans • If/Else Statements • Logical Operators • Comparison Operators • Conditionals • While Loops • Break Statements • For Loops • Nested Control Structures

Assignments / Labs	• 31 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: Jukebox ■ 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. ● Drawing basic graphics using JavaScript ○ 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.

Module 7: Functions (2 weeks/10 hours)

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

Browse the full content of this unit at https://codehs.com/course/19963/explore/module/27376

Objectives / Topics Covered	● Parameters ● Return Values ● Default Parameters ● Scope
Assignments / Labs	● 12 functions programming exercises in total ● Using various kinds of functions such as functions with and without parameters, and functions with and without return values ○ 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. In order to check if the inputted integer is even or odd, you should define and call a function named `isEven()`. This function should return a Boolean value of `true` or `false` depending if the number is even or not.

Module 8: Animation and Games (3 weeks/15 hours)
Students learn how to make objects move around the screen and let users interact using the mouse!

Browse the full content of this unit at https://codehs.com/course/19963/explore/module/27378

Objectives / Topics Covered	• Timers • Randomizing Games • Mouse Events • Keyboard Events

Assignments / Labs	• 19 animations programming exercises in total
	• Throughout the lessons in this module, you will be developing a simple game that incorporates basic animation techniques and input events.
	• 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: 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.
	  If you're feeling adventurous, you can extend this to draw a small red circle whenever you click.
	  If you're feeling really adventurous, you can have a bouncing ball on the screen and see if you can remove it when it gets clicked. You can use remove(obj) to remove something from the screen and getElementAt(x, y) to get an object at the given position. It will return the object or will return null if there is no object there.
	• Using keyboard events for interactive programs
	  ○ 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.

Module 9: Final Project (2-4 weeks, 10-17 hours)
Students apply the skills they've learned throughout the course to create an original program!

Browse the full content of this unit at https://codehs.com/course/19963/explore/module/27396

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

Module 10: Computer Science Careers (1 week/5 hours)
Students take some time to explore and discover different computer science careers. Students will learn more about resumes and develop one of their own. Students will also research about professional student organizations and the benefits they offer to their members.

Browse the full content of this module at https://codehs.com/course/19963/explore/module/27399

Objectives / Topics Covered	● Computer Science Careers ● Computer Science used in non-CS Careers ● Inclusive Coding ● Create an Online Portfolio ● Student Organizations
Assignments / Labs	● Career Research ● Computer Science concepts that are used in non-computer science positions ● How can a computer program's bias become dangerous? ● Why is it important to have a more diverse group of people in the computer science field? ● Exploring online portfolio options ● Understanding and developing a resume ● What are student organizations? ○ How can they help you learn about careers and competitive opportunities?

Module 11: Intro to Physical Computing with micro:bit (1-2 weeks/5-10 hours)
Students will be introduced to their micro:bit device and the MakeCode editor, where they will build and test programs throughout the course. They will also be guided through the process to download programs from the editor to their device and will learn a few basic commands to get started writing micro:bit programs.

Browse the full content of this unit at https://codehs.com/course/19963/explore/module/27400

Objectives / Topics Covered	● Intro to physical computing ● Goal Setting ● Analog vs. digital ● Variables ● Connecting external components
Example Assignments / Labs	● 4 explorations ● 11 exercises total ● Example exercises: ○ Blinking Diamond ■ Blink LEDs between a large diamond shape and small

	diamond shape every half a second as long as the program is running ○ Moving Bright Box ■ Light the corners on the grid with the highest brightness for half a second. Light the corners of the inner box with a brightness of 150 for half a second. Light the middle LED with a brightness of 50 for half a second. Repeat until the program is manually ended. ○ Opposite Blinking External LEDs ■ Blink two external LEDs opposite one another. One LED should be lit while the other is off. After 1 second, the lit LED should turn off and the unlit LED should turn on. After 1 second, they should switch again. This should continue until the program is manually ended

Optional Supplemental Materials (Remainder of school year)

These supplemental materials should be used following the Prerequisite Modules mentioned:

Supplementary Modules	**Prerequisite/Recommended Module(s)**	**# of activities**
JavaScript Midterm Exam	Programming with Karel, JavaScript Basics, The Canvas and Graphics, Control Structures	1
JavaScript Final Exam	Programming with Karel, JavaScript Basics, The Canvas and Graphics, Control Structures, Functions, Animation and Games, JavaScript Project	1
Karel Challenges	Programming with Karel	5
Practice: Karel	Programming with Karel and Karel Challenges	23
Practice: Console Challenges	Programming with Karel, JavaScript Basics	17
Practice: Graphics and Animation	The Canvas and Graphics, Control Structures	10
Graphics Challenges	Control Structures	3
Control Structures Challenges	Control Structures	3
Practice: Functions	Functions	7
Functions Challenges	Functions, Practice: Functions	3
Animation Challenges	Animation & Games	2
Project: Breakout	Programming with Karel, JavaScript Basics, The Canvas and Graphics, Control Structures, Functions, Animation and Games, JavaScript Project	4