CodeHS

Illinois Introduction to Computer Science
1 year for High School (128-148 contact hours)

Course Overview and Goals

The CodeHS introduction to computer science 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 Introduction to Computer Science 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 module 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. Many modules end with a comprehensive module test that assesses students' mastery of the material from that module.

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/19648.

Prerequisites

The Illinois Introduction to Computer Science 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: Programming with Karel (3 weeks/15 hours)
Students learn the basics of programming by giving Karel the Dog commands in a grid world.

Browse the full content of this module at https://codehs.com/course/19648/explore/module/26721

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 2: 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 module at https://codehs.com/course/19648/explore/module/26723

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 3: 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 module at https://codehs.com/course/19648/explore/module/26728

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 4: 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 module at https://codehs.com/course/19648/explore/module/26729

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 5: 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 module at https://codehs.com/course/19648/explore/module/26730

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 6: 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 module at https://codehs.com/course/19648/explore/module/26731

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 7: JavaScript Project (2-4 weeks/10-20 hours)

Students apply the skills they've learned throughout the course to create an original program!

Browse the full content of this module at https://codehs.com/course/19648/explore/module/26734

Objectives / Topics Covered	● Collaborative Programming ● Project Planning ● Pseudocode
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 8: 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/19648/explore/module/26755

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 9: Data (1 week/5 hours)

Students explore using computational tools to store massive amounts of data, manipulate and visualize data, find patterns in data, and pull conclusions from data.

Browse the full content of this module at https://codehs.com/course/19648/explore/module/26756

Objectives / Topics Covered	● Filtering and Cleaning Data ● Patterns and Trends ● Search Tools ● Tables, Diagrams and Displays ● Interactive Visualizations ● Combining Data Sources
Assignments / Labs	● Storing, collecting, manipulating, and visualizing data ● Impact of visually representing data to make information easier to analyze ● Data-driven decisions using visualizations ● Data collection and limitations

Module 10: Present a Data-Driven Insight (3 days/3 hours)

Students apply data concepts into a project where they can explore their own data application.

Browse the full content of this module at https://codehs.com/course/19648/explore/module/26757

Objectives / Topics Covered	● Collaboration ● Processing data ● Explain and communicate findings
Assignments / Labs	● Topic Brainstorm ● Gathering Resources ● Make a Visualization ● Sharing Your Findings

Module 11: What is Cybersecurity? (1 week/5 hours)

Students explore the basics of cybersecurity. Students learn about why cybersecurity is important, recent threats to cybersecurity, and different careers in the field.

Browse the full content of this module at https://codehs.com/course/19648/explore/module/26747

Objectives / Topics Covered	● What is Cybersecurity? ● Impact of Cybersecurity ● The CIA Triad
Assignments / Labs	● What is Cybersecurity? ○ Why is cybersecurity important? ○ Cybersecurity in the news ○ Cybersecurity and IoT (Internet of Things) ○ How do we prevent cyber attacks? ○ Example activities: ■ Summarize and discuss recent cyber attacks ■ Explore a threat map to see where cyber attacks are coming from and which countries are being targeted ● Impact of Cybersecurity ○ Why do we care about cybersecurity? ○ What information is at risk? ○ What are the impacts of cyber attacks? ○ Cybersecurity workforce ○ What are current cybersecurity careers? ○ Example activities: ■ Review resources and reflect on or discuss ● What information do cyber criminals steal? ● What do cyber criminals do with stolen information? ● The CIA Triad ○ What is the CIA triad? (confidentiality, integrity, availability) ○ What are “secure systems?” ○ What do confidentiality, integrity, and availability mean in cybersecurity? ○ Example activities: ■ Determine where scenarios break part of the CIA Triad

Module 12: Digital Citizenship and Cyber Hygiene (1-2 weeks/5-10 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/course/19648/explore/module/26748

Objectives / Topics Covered	● Digital Footprint and Reputation ● Cyberbullying ● Internet Safety ● Privacy and Security ● Information Literacy ● 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 you 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

Module 13: System Administration (2-3 weeks/10-15 hours)
Students will compare and contrast common operating systems (Windows, Linux, OS) and explain the importance of application security. They will investigate security options and implement user accounts to enforce authentication and authorization. Students will also demonstrate how to work with basic and advanced command prompts.

Browse the full content of this module at https://codehs.com/course/19648/explore/module/26749

Objectives / Topics Covered	● Operating Systems ● Software and Applications ● Application Security ● Browser Configuration ● System Administration ● Command Line Interface
Example Assignments / Labs	● Understanding Operating Systems ● Comparing Operating Systems ○ Installing an OS ● File Management ○ What Processor are you Running? ● Software Licenses ● Antivirus Software ○ Data Backups ● Using Cache ● Popup Blockers ● User Accounts ○ Admin vs. Standard

	● Host Security ○ Using a Log ● System Commands ○ cd, ls, mk etc ● Network Commands ○ ipconfig, netstat etc

Module 14: Software Security (2-3 weeks/10-15 hours)

In this module, students will learn what happens when running a web application and how to look inside web apps using developer tools, source code, and more. They will learn basic SQL and common attacks like SQLi. Students will also be able to recommend solutions for flawed security systems.

Browse the full content of this module at https://codehs.com/course/19648/explore/module/26750

Objectives / Topics Covered	● Inside Web Applications ● Developer Tools ● The Value of Data ● SQL Overview ○ What is SQL? ○ Structuring Data in SQL ○ Basic Querying in SQL ○ Filtering Queries in SQL ● Clients, Servers, Databases ● Common Security Problems ● SQL Injection ○ SQLi Overview ○ Types of SQLi ○ Preventing SQLi
Example Assignments / Labs	● Inside Web Applications ○ View page source (images, navigation and page layout, stylesheets, JavaScript, minified code ○ Example activities: ■ View page source scavenger hunt ■ Getting started with OWASP ● Developer Tools ○ Use the inspect tools to look more deeply inside of web apps ○ How does view page source compare to inspect in terms of information about the site / app? ○ Example activities: ■ Practice using the Chrome developer tools ■ Change a favorite site using the Chrome developer tools on your end only. Take a screenshot of your change. ● Data Visualizations ● Design a Survey ● SQL Overview ○ What is SQL? ○ How do we structuring data using SQL? ○ How do we query databases using SQL? ○ Example activities:

	■ Use the SELECT statement to query a database ■ Use the WHERE clause to query a database ● Clients, Servers, Databases ● Common Security Problems ○ What is the “Fortification Principle”? ○ What are some tips about HTTP vs. HTTPS, password fields and CAPTCHA that can help us to navigate more securely on the Web? ● SQL Injection ○ SQLi Overview ■ What is SQLi? ■ Why is SQLi a problem? ■ What happens during a SQLi attack? ■ What is the the fallout of a SQLi attack? ■ How does SQLi work? ■ How do hackers use SQL in a SQLi? ○ What are the types of SQLi (error-based, union-based, blind) ■ What is the underlying SQL behind the scenes that hackers may be trying to hack? ○ How to we mitigate or prevent SQLi? ■ What are the OWASP recommendations? ■ How can we tell if our code is vulnerable? ○ Example activities: ■ Discuss the Equifax SQL injection attack ■ Practice basic SQLi on a safe site ■ Research SQLi prevention

Module 15: Networking Fundamentals (2-3 weeks/10-15 hours)

This module explores the structure and design of the internet and networks, and how this design affects the reliability of network communication, the security of data, and personal privacy. Students will learn how the Internet connects computers all over the world by use of networking protocols.

Browse the full content of this module at https://codehs.com/course/19648/explore/module/26751

Objectives / Topics Covered	● Introduction to the Internet ● Notational Systems ● Data Representation ● Internet Hardware ● Internet Addresses ● Domain Name System (DNS) ● Routing ● Packets and Protocols ● The Internet and Cybersecurity ● Impact of the Internet
Example Assignments / Labs	● Introduction to the internet ○ What is the Internet? How does it work? What have been its impact on society? ○ Why do we need protocols for the Internet? ○ Example Activity ■ Explore the different levels of the internet.

- Decimal to Binary
- Hexadecimal
- Bits to ASCII
    - Hello World in Bits
- Internet hardware
    - Vocabulary: bandwidth, bitrate, latency
    - Why are protocols so important?
    - How do we send data over the Internet?
    - Example Activities
        - Explore how data is able to be transmitted across the ocean by using underwater cables
        - Explore the role of simple and complex networks and routers
- Internet Addresses
    - Vocabulary: Internet Protocol (IP)
    - How do IP addresses compare to postal addresses?
    - How IP addresses work?
    - Example Activities
        - Explore the differences between IPv4 and IPv6. Why are we running out of addresses?
        - Trace a website request from the server, through the network, and to your computer
- Domain Name System (DNS)
    - How does DNS help with sending digital information and IP addresses?
    - Example Activities
        - Explore the process of how requesting a web resource works
- Routing
    - How is routing used to send messages / data?
    - Why is redundancy a good thing for the Internet? (fault tolerant)
- Packets and Protocols
    - How data is transmitted?
    - How are internet packets able to find their way to your computer?
    - Example Activities:
        - 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?
        - 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.
    - What are the standard protocols for the Internet and how do they work? (TCP/IP, HTTP)
- The Internet and Cybersecurity
    - What are cybercrime and cyberwarfare?
    - How do we network attacks? (certificate authorities, public key encryption)

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 and Graphics, Control Structures	1
JavaScript Final Exam	Programming with Karel, JavaScript 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	Control Structures	10
Graphics Challenges	Control Structures, Graphics and Animations	3
Control Structures Challenges	Control Structures	3
Practice: Functions	Functions & Parameters	7
Functions Challenges	Functions, Practice: Functions	3
Animation Challenges	Animation & Games	2
Project: Breakout	Programming with Karel, JavaScript and Graphics, Control Structures, Functions, Animation and Games, JavaScript Project	4
Extension: Visualizing Music	Data Structures (array)	9
Extension: Data Structures - Arrays - Lists - Objects - Sets - Grids	Complete all modules in main course	78
Practice: Data Structures Challenges	Data Structures	6
Project: Tic Tac Toe	Data Structures	4
Project: Helicopter Game	Data Structures	24