

New Jersey Computer Science & Design Thinking Syllabus

High School (130-145 Contact Hours)

Course Overview and Goals

CodeHS's New Jersey Computer Science and Design Thinking 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 with the New Jersey Computer Science and Design Thinking Computer Science standards outlined by the New Jersey 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 students' mastery of the material from that unit and 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/17480/explore>

Prerequisites

The New Jersey Computer Science and Design Thinking 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: Introduction to Programming in JavaScript with Karel the Dog (4 weeks/20 hours)

The first module 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.

Objectives / Topics Covered	<ul style="list-style-type: none">● Commands● Defining vs. Calling Methods● Designing methods
-----------------------------	---

	<ul style="list-style-type: none"> ● Program entry points ● Control flow ● Looping ● Conditionals ● Classes ● Commenting code ● Preconditions and Postconditions ● Top-Down Design
Assignments / Labs	<ul style="list-style-type: none"> ● 26 Karel Programming Exercises and Challenges in total ● Program-specific tasks for Karel the Dog <ul style="list-style-type: none"> ○ Example Exercise: Pyramid of Karel Write a program to have Karel build a pyramid. There should be three balls in the first row, two in the second row, and one in the third row. ● Teach Karel new commands like <code>turnRight()</code> or <code>makePancakes()</code> <ul style="list-style-type: none"> ○ 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. ● Solve large Karel problems by breaking them down into smaller, more manageable problems using Top-Down Design <ul style="list-style-type: none"> ○ Example Exercise: The Two Towers In this program, Karel should build two towers of tennis balls. Each tower should be 3 tennis balls high. In the end, Karel should end up on top of the second tower, facing East. ● Using control structures and conditionals to solve general problems <ul style="list-style-type: none"> ○ Example Exercise: Random Hurdles Write a program that has Karel run to the other side of the first street, jumping over all of the hurdles. However, the hurdles can be in random locations. The world is fourteen avenues long.

Module 2: JavaScript & Graphics (1.5 weeks/7 hours)

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

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Variables ● User Input ● Arithmetic Expressions ● Graphics ● Image Representation
Assignments / Labs	<ul style="list-style-type: none"> ● 7 JavaScript & graphics programming exercises in total ● Using variables and getting user input using JavaScript <ul style="list-style-type: none"> ○ 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 <ul style="list-style-type: none"> ○ Example Exercise: Flag of the Netherlands

	<p>This program should draw the flag of 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.</p>
--	---

Module 3: 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.

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Booleans ● For Loops ● Conditionals ● Nested Control Structures ● While Loops
Assignments / Labs	<ul style="list-style-type: none"> ● 22 control structures programming exercises in total ● Using comparison and logical operators to control the flow of the program <ul style="list-style-type: none"> ○ 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 <ul style="list-style-type: none"> ○ 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 <ul style="list-style-type: none"> ○ 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 4: Functions and Parameters (5 weeks/25 hours)

Students learn to write reusable code with functions and parameters.

Objectives / Topics Covered	<ul style="list-style-type: none"> ● 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	<ul style="list-style-type: none"> ● 32 functions programming exercises in total <ul style="list-style-type: none"> ○ 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 <ul style="list-style-type: none"> ■ 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 <ul style="list-style-type: none"> ■ 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.
--------------------	---

Module 5: Project: Tell a Story (3 days/3 hours)

In this project, students will write a JavaScript program that tells a graphical story

Assignments / Labs	<ul style="list-style-type: none"> ● <i>Tell a Story!</i> In this activity, students write a JavaScript program that tells a graphical story in at least 4 scenes. Following the milestones and the pseudocode plan that students have laid out prior to this exercise, students write the code for their final project. They iterate and test their code along the way to make sure they have solved each milestone.
--------------------	--

Module 6: Basic Data Structures (6 weeks/30 hours)

This module introduces students to basic data structures that make storing and accessing data easier.

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

Assignments / Labs	<ul style="list-style-type: none"> ● 17 exercises in total ● Basic list/array operations <ul style="list-style-type: none"> ○ 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. ● Iterating through arrays/lists <ul style="list-style-type: none"> ○ 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. ● Basic list/array algorithms <ul style="list-style-type: none"> ○ Example Exercise: Remove From Line you are given an array of names of people who are in line for movie tickets. Use the <code>remove element</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.
--------------------	--

Module 7: Digital Information (3 weeks/15 hours)

In this module, students will learn about the various ways we represent information digitally. Topics covered include number systems, encoding data, programmatically creating pixel images, comparing data encodings, compressing and encrypting data. Students will work in pairs to develop their own data encryption algorithms and attempt to crack the encryptions of their peers. Their text encryption tool will be embedded in their portfolio websites.

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Number Systems ● Encoding Text with Binary ● Pixel Images ● Converting Binary to Hexadecimal
Assignments / Labs	<ul style="list-style-type: none"> ● Pixel Colors <ul style="list-style-type: none"> ○ How are colors represented and created using hexadecimal number systems? ○ What are pixel images, and how are they created? ● Encoding Text with Binary <ul style="list-style-type: none"> ○ How is information transmitted and sent using binary, and encoded to protect private information? ○ How is binary converted to decimal and back?

Module 8: The Internet and Computers (1.5 weeks/8 hours)

This module 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.

Objectives / Topics Covered	<ul style="list-style-type: none"> ● 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	<ul style="list-style-type: none"> ● Structure of the internet <ul style="list-style-type: none"> ○ Explore the differences between IPv4 and IPv6. Why are we running out of addresses? ○ Explore the different levels of the internet. ○ Example Activity <ul style="list-style-type: none"> ■ Trace a website request from the server, through the network, and to your computer ● How data is transmitted <ul style="list-style-type: none"> ○ 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: <ul style="list-style-type: none"> ■ 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 <ul style="list-style-type: none"> ○ 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 9: Project: The Effects of the Internet (3 days/3 hours)

In this project, students will choose an innovation that was enabled by the Internet and explore the positive and negative impacts of their innovation on society, economy, and culture. Students will develop a computational artifact that illustrates, represents, or explains the innovation’s purpose, its function, or its effect, and embed this artifact in their personal portfolio website.

Assignments / Labs	<ul style="list-style-type: none"> ● <i>The Effects of the Internet:</i> Students provide evidence of the extensive knowledge they have developed about a chosen Internet-based innovation and its impact(s). Students include citations, as applicable, within their written responses. <p>Within their computational artifact, students explain at least one beneficial effect and at least one harmful effect the Internet-based innovation has had or has the potential to have, on society, economy, or culture. They also identify data privacy, security, or storage concerns for the computing innovation.</p>
--------------------	---

Module 10: Data (1 week/5 hours)

In this module, students will explore using computational tools to store massive amounts of data, manipulate and visualize data, find patterns in data, and draw conclusions from data. Students will consider how the modern wealth of data collection has impacted society in positive and negative ways. Students will work in teams to investigate a question of personal interest and use public data to present a data-driven insight to their peers. They will develop visualizations to communicate their findings, and embed their visualizations in their portfolio websites.

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Visualizing and Interpreting Data ● Data Collection and Limitations
Assignments / Labs	<ul style="list-style-type: none"> ● Visualizing and Interpreting Data <ul style="list-style-type: none"> ○ Students examine a series of visualizations to determine the strength of the visualization in communicating data. ● Data Collection and Limitations <ul style="list-style-type: none"> ○ Students explore how visualizations can misrepresent data. Students determine the validity of visualizations based on data context.

Module 11: Project: Present a Data-Driven Insight (3 days/3 hours)

In this project, students will work with a partner to answer a question of personal interest using a publicly available data set. Students will need to produce data visualizations and explain how these visualizations led to their conclusions. They will develop a computational artifact that illustrates, represents, or explains their findings, communicate their findings to their classmates, and embed their artifact in their personal portfolio website.

Assignments / Labs	<ul style="list-style-type: none"> ● <i>Present a Data-driven Insight:</i> Students consider how the amount of collected data impacts our lives in ways that require considerable study and reflection for us to fully understand them. Students explore a question that can be answered by analyzing a dataset. They form a question and use visualization techniques to analyze the data to answer the question.
--------------------	---

Module 12: Project: Computing and the Environment (1 week/5 hours)

In this project, students will investigate an environmental issue of their interest and develop an innovative and sustainable solution to address the issue. Students will need to develop a system of investigation and gather sources for their research. They will then create a computational artifact that explains their findings and their proposed solution to the environmental issue and embed the artifact in their personal portfolio website.

Assignments / Labs	<ul style="list-style-type: none"> ● <i>Intro to Computing and the Environment:</i> Students will consider how digital technology is being used to help address global environmental issues. Students will also learn about the consequences of using digital technology and the impact it can have on the environment. Students will then learn about Digital Carbon Footprint and explore how they might be contributing to greenhouse gas emissions and what they can do to limit their contribution.
--------------------	---

	<ul style="list-style-type: none"> ● <i>Oil in the Cloud</i>: Students will take a look at a report that will serve as an example of how to investigate an environmental issue followed by brainstorming topics for their research project. ● <i>Create Your Artifact</i>: Students will spend time researching the topic of their choice and gathering useful sources. After they have enough information on their topic, they will focus on developing an innovative and sustainable solution to the environmental issue they researched. Once they are finished developing a solution, students will put together an artifact that explains their findings.
--	--

Module 13: Creative Development (3 weeks/15 hours)

In this module, students will brainstorm their own final project, discuss their ideas with their peers, scope their project to fit within the time constraints of the class, plan out milestones for incremental development, and create their own final product from scratch. This project allows students to think creatively about the applications of the concepts covered in the course, and create something of personal value.

<p>Objectives / Topics Covered</p>	<ul style="list-style-type: none"> ● What makes an engaging interface? ● Various User Interface (UI) Design techniques ● Accessibility issues ● Readability ● Rapid prototyping ● User testing
<p>Assignments / Labs</p>	<ul style="list-style-type: none"> ● Example exercises: <ul style="list-style-type: none"> ○ Research existing user interfaces ○ Assess the user interfaces of various websites ○ Research a problem, and gather empathy and insights for stakeholders who are involved in the problem ○ Create a problem statement based on user research ○ Generate potential solutions for a problem ○ Prototype and test several solutions ○ Design a website using paper prototypes, test these prototypes and get feedback from your peers, and improve your design before implementing it with code ○ UI Design Project <ul style="list-style-type: none"> ■ Find and present an article about a particular UI design technique ■ Create your own live examples using this technique

Optional Supplemental Materials

<p>Objectives / Topics Covered</p>	<ul style="list-style-type: none"> ● Extra practice with: <ul style="list-style-type: none"> ○ Karel
------------------------------------	---

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