CodeHS

Tennessee Introduction to Computer Science Syllabus
1 year for High School (115-135 contact hours)

Course Overview and Goals

The Tennessee Computer Science Foundations course is intended to introduce students to the fundamentals of computer science. The course is designed to cover the Tennessee High School Computer Science standards.

The course explores the fundamentals of computer science including the impact of technology on society, cybersecurity, data analysis, and programming. Throughout the course, students will use JavaScript as they explore basic programming concepts such as variables, loops, conditional statements, and lists.

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 90 hours of hands-on programming practice in total. Units end with a comprehensive unit test that assesses students' mastery of the material from that unit as well as challenge problems where students can display their understanding of the material.

Development Environment: Students write and run JavaScript files in the browser using the CodeHS editor. We highly recommend that all student computers use an up-to-date version of the Chrome browser. You can download an up-to-date version of Chrome for free here: https://www.google.com/chrome/browser/

More information: Browse the content of this course at https://codehs.com/course/19781

Prerequisites: There are no prerequisites for the Tennessee Introduction to Computer Science course.

Course Breakdown

Unit 1: Our Digital World (3 weeks/15 hours)
This module includes topics on Internet etiquette and how to stay safe on the world wide web. It will also look at the potential effects of our digital footprints and how to navigate the legal aspects of the internet, including copyright laws, terms of service, and privacy policies. Finally, the module looks forward by exploring careers in computer science and the future of technology.

Browse the full content of this module at codehs.com/library/course/19781/module/26925

Objectives / Topics Covered	- Digital Footprint and Reputation - Cyberbullying - Internet Safety - Creative Credit and Copyright - Terms of Service and Privacy Policies

	● Computer Science Careers ● Future of Computing

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?
	● 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
	● Terms of Service and Privacy Policies
	   ○ Privacy Policies - What are you agreeing to?
	   ○ Exploring companies' privacy policies
	   ○ Example activities:
	      ■ Explore and reflect on Apple's Privacy Policy
	      ■ Find a company of your choice and examine their privacy policy
	● Computer Science Careers
	   ○ Computer Science all around us
	   ○ Computer Science Careers
	   ○ Coding in the Wild
	   ○ Inclusive Coding
	   ○ Example activities:
	      ■ Explore and reflect different career options
	      ■ Reflect on ways that computer science is used in other fields
	● Future of Computing
	   ○ Using DNA Storage

	○ Artificial Intelligence
	○ Example activities:
	    ■ Class activity: Advancing technology
	    ■ Compare and contrast the benefits of AI
	
	● 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.

Unit 2 Cybersecurity (2 weeks/8 hours)

This module includes topics on cybersecurity, including an introduction to the CIA Triad. Students start by understanding the threats and how the can be hacked, and then armed with this knowledge, they look at ways to protect themselves.

Browse the full content of this module at codehs.com/library/course/19781/module/26948

Objectives / Topics Covered	● What is Cybersecurity? ● The CIA Triad ● Security ● Hacking Ethics ● Web Tracking

Example Assignments / Labs	● What is Cybersecrity?
	    ○ Ransomware attacks
	    ○ Internet of Things
	    ○ Cybersecurity and connected devices
	    ○ Threat Maps
	    ○ Example activities:
	        ■ Ransomware Simulator
	        ■ Explore and reflet on the WannaCry attck
	● The CIA Triad
	    ○ What is the CIA Triad
	    ○ Breach of various parts of the triad
	    ○ Example activities:
	        ■ Breach in Confidence reflection
	        ■ A Balance Game - Interactive activity
	● Security
	    ○ What is 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?
	● 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 ● Web Tracking ○ Who can track you ○ Ways to mitigate being tracked ○ Example activities: ■ Conversation with a data broker ■ Reducing being tracked

Unit 3: Data (2 weeks/8 hours)

This module includes topics on data and data science. Students start by understanding what data is available and how it can be organized and presented. From there, they get into looking at different ways to visualize the data and understand the limitations of their data.

Browse the full content of this module at codehs.com/library/course/19781/module/26949

Objectives / Topics Covered	● What is Data Science? ● Information Literacy ● Gathering Data ● Visualizing and Interpreting Data ● Data Collection and Limitations
Example Assignments / Labs	● What is Data Science? ○ What is data science ○ ○ Lookup, Compute, and Relate data ○ Example activities: ■ Exploring various data representations ■ What does a data scientist do? ● 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 ● Gathering Data ○ Qualitative versus Quantitative data ○ Sorting Data ○ Data Science Life Cycle ■ Collecting and structuring data ■ Planning data for analysis ● Visualizing and Interpreting Data ○ Exploring Data Visualizations ○ Choosing a visualization to represent data ○ Example activities: ■ Exploring data on National Parks and other topics ● Data Collection and Limitations ○ Surveys ○ Automated data collections

	○ Limitations of data
	○ Example activities:
	■ Creating various graphs
	■ Exploring self-driving cars

Unit 4: Introduction to Programming with Karel the Dog (4 weeks/20 hours)

Module four 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.

Browse the full content of this unit at codehs.com/library/course/19781/module/26926

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	● 26 Karel Programming Exercises and Challenges 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 avenue. Each stack of pancakes should have three pancakes.
	     Create a method called `makePancakes()` to help Karel solve this problem.
	● 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 first street, jumping over all of the hurdles. However, the hurdles can be in random locations. The world is fourteen avenues long.

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.
Browse the full content of this unit at codehs.com/library/course/19781/module/26929

Objectives / Topics Covered	- Variables - User Input - Arithmetic Expressions - Graphics - Image Representation
Assignments / Labs	7 JavaScript & graphics 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! - Create basic graphics in JavaScript - 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.

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.
Browse the full content of this unit at codehs.com/library/course/19781/module/26931

Objectives / Topics Covered	- Booleans - For Loops - Conditionals - Nested Control Structures - While Loops
Assignments / Labs	22 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: 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 - 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.

Unit 7: Functions, Events, and Lists (5 weeks, 25 hours)

Learn to write reusable code with functions and parameters.

Browse the full content of this unit at codehs.com/library/course/19781/module/26933

Objectives / Topics Covered	● 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 ● APIs

Assignments / Labs	● 32 functions programming exercises in total
	  ○ 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
	    ■ 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: 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.

	○ Using keyboard events for interactive programs
	■ Example Exercise: Basic Snake

Unit 8: 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!

Browse the full content of this unit at codehs.com/library/course/19781/module/26934

Objectives / Topics Covered	- Project Planning - Final Project Creation - Testing and peer feedback
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