Oklahoma Computer Science II Syllabus

High School (110 Contact Hours)

Course Overview and Goals

In this Oklahoma Computer Science II course, students will learn 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 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/24120/overview

Prerequisites

The Oklahoma Computer Science I course precedes this course. However, the Oklahoma Computer Science II course is highly visual, dynamic, and interactive, making it engaging and accessible for those new to computer science.

Course Breakdown

Module 1: Programming with Karel (4 weeks/20 hours)

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/24120/explore/module/33173

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	● 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.

Module 2: Karel Challenges (1.5 weeks, 8 hours)

In this module, students will take all the foundational concepts from Karel to solve some programming challenges.

Browse the full content of this unit at https://codehs.com/course/24120/explore/module/33174

Objectives / Topics Covered	● Solving large and more complex problems using Karel
Assignments / Labs	● 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 in any size world with tennis balls in any location.

Module 3: JavaScript & Graphics (2 weeks/10 hours)

In this module, students are introduced to the basics of JavaScript, including variables, user input, mathematics, basic graphics, and image representations.

Browse the full content of this unit at https://codehs.com/course/24120/explore/module/33176

Objectives / Topics Covered	● Variables ● User Input ● Arithmetic Expressions ● Graphics ● Image Representation
Assignments / Labs	● Using variables and getting user input using JavaScript ○ Example Exercise: Dinner Plans Prompt the user for their name, then ask them what time they 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.

Module 4: Graphics Challenges (1 week, 5 hours)

In this module, students will learn how to pair-program! They will also take all the foundational concepts from JavaScript Graphics to solve some programming challenges.

Browse the full content of this unit at https://codehs.com/course/24120/explore/module/33177

Objectives / Topics Covered	● Pair programming ● Solving large and more complex problems that incorporate graphics and user input
Assignments / Labs	● 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.)

Module 5: JavaScript Control Structures (3 weeks/15 hours)

In this module, students 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 https://codehs.com/course/24120/explore/module/33178

Objectives / Topics Covered	● Booleans ● For Loops ● Conditionals ● Nested Control Structures ● While Loops

Assignments / Labs	● 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.

Module 6: Control Structures Challenges (1 week, 5 hours)

In this module, students will take all the foundational concepts from the Control Structures unit to solve some programming challenges.

Browse the full content of this unit at https://codehs.com/course/24120/explore/module/33179

Objectives / Topics Covered	● Solving large and more complex problems that incorporate control structures
Assignments / Labs	● Example Exercise: Circles in Squares ○ Repeatedly draw a circle inside of a square and then a square inside that circle and so on. You should repeat this while the size is greater than `MIN_SIZE`. Each time you draw a shape, it should be centered on the screen and a new random color. You'll need to update the size based on the size of the previous one.

Module 7: Functions (2 weeks, 10 hours)

In this module, students learn to write reusable code with functions and parameters.

Browse the full content of this unit at https://codehs.com/course/24120/explore/module/33180

Objectives / Topics Covered	● Functions with and without parameters ● Functions with and without return values ● Nested Control Structures
Assignments / Labs	● 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.

Module 8: Animations (2 weeks, 10 hours)

In this module, 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/24120/explore/module/33372

Objectives / Topics Covered	● Timers ● Randomization ● Mouse Events
Assignments / Labs	● 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. ● 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.

Module 9: Data Structures: Arrays (2 weeks, 10 hours)

In this module, students learn about arrays and how to iterate through them.

Browse the full content of this unit at https://codehs.com/course/24120/explore/module/33373

Objectives / Topics Covered	● Lists/Arrays ● Adding/Removing From an Array ● Array Length ● Iterating Over an Array ● Applying Iteration
Assignments / Labs	● Example Exercise: Changing Circles ○ Write a program that creates a list of NUM_CIRCLES circles on the screen of different sizes from MIN_RADIUS to MAX_RADIUS. Every DELAY milliseconds, get a random color and change all of the circles to this color. All the circles should have the same color at all times. ● Example Exercise: Usernames and Passwords

	○ Write a program that displays a list of 5 usernames. The program will then ask the user to create a password for each user that will be stored in a separate list. Once all the passwords are entered the program should display the list of passwords.

Module 10: Software Development (2 weeks, 10 hours)

In this module, students will learn and use the software development life cycle to create a simple app that will solve an everyday problem.

Browse the full content of this unit at https://codehs.com/course/24120/explore/module/33308

Objectives / Topics Covered	● Software Development Life Cycle ● Prototyping ● Using Flowcharts ● User Testing ● Software Licenses
Assignments / Labs	● Example Exercise: Create a Flowchart ○ Create a flowchart that visualizes the flow of your app. You can use paper or a digital tool to create your flowchart. This flowchart will still be a draft - you can always come back and edit it later on as you continue development! ● Example Exercise: Test Your App! ○ Now that you have a plan, you are ready to conduct your user testing! You should have at least two different users test your app. In the editor, submit the data you collect during testing. This could be a link to images, videos, or a text document, or a summary of your findings typed into the editor.

Module 11: Data-Driven Insights (1 week/5 hours)

In this module, students will explore a problem that can be answered by analyzing a dataset. They will form statistical questions and use visualization techniques to analyze the data to answer the question.

Browse the full content of this module at https://codehs.com/course/24120/explore/module/33172

Objectives / Topics Covered	● The Value of Data ● Data Collection and Limitations ● Data Visualizations
Assignments / Labs	● Example Exercise: Topic Brainstorm ○ In this exercise, brainstorm and choose a topic that you will be exploring for your project. Allow your interests to drive your choice. Work with your partner to develop a problem statement to help you decide on the topic for your project. A problem statement is a detailed description of a problem or an issue that a project seeks to address.

Module 12: Computer Science Careers (2 days/2 hours)

In this module, students explore potential career paths in the field of computer science.

Browse the full content of this module at https://codehs.com/course/24120/explore/module/33182

Objectives / Topics Covered	- Computer Science Careers - Inclusive Coding
Assignments / Labs	- Example Exercise: CS Careers ○ This video describes 10 different popular computer science careers. After you watch, you will choose one of the careers given to research in more detail so make sure to pay attention to the different options!