Course Overview and Goals

The Nexitintroduction 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 NexitIntroduction 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 Nexitto 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.

Programming Environment: Students write and run JavaScript programs in the browser using the Nexiteditor.

Prerequisites

The Intro to Computer Science in JavaScript 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.

	● Designing methods ● Program entry points ● Control flow ● Looping ● Conditionals ● Classes ● Commenting code ● Preconditions and Postconditions ● Top Down Design
Assignments / Labs	● 30 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. ○ Example Exercise: Super Cleanup Karel

Unit 2: Karel Challenges (2 weeks, 10 hours)

Objectives / Topics Covered	• Solving large and more complex problems using Karel
Assignments / Labs	• Several Karel challenges to tie everything learned in the Karel module together

Unit 3: Javascript & Graphics (2 weeks/10 hours)

Objectives / Topics Covered	• Variables • User Input • Arithmetic Expressions • Booleans
Assignments / Labs	• Using variables and getting user input using JavaScript ○ Example Exercise: Grocery Store Prompt the user for their name, and then how many apples, and then how many oranges they would like to buy. Then print out the name that was given, as well as how many apples and oranges they wanted.

Unit 4: JavaScript Control Structures (4 weeks/20 hours)

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

	● 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. Hint: You can’t use i for both for loops. ● Drawing basic graphics using JavaScript ○ Example Exercise: French Flag This program should draw the French flag. The left third of the canvas is blue, the middle third is white, and the right third is red. You will need to use Rectangle objects in this program. ○ 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 5: Functions and Parameters (4 weeks/20 hours)

Objectives / Topics Covered	● Functions with and without parameters ● Functions with and without return values ● Nested Control Structures ● Local variables and scope
Assignments / Labs	● Using various kinds of functions such as functions with and without parameters, and functions with and without return values ○ Example Exercise: Horizontal Lines Write a function that draws horizontal lines on the graphics canvas. If a line is horizontal, then the y-values for the endpoints are the same. The parameters to your function should be the y location, and the length, and all of your lines should start at x position 0.

	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.

Unit 6: JavaScript and Graphics Challenges (1 week/5 hours)

Objectives / Topics Covered	● Solving large and more complex problems using in JavaScript and graphics
Assignments / Labs	● Graphics Challenges to tie everything in the module together ○ Example Exercise: Ghosts Write a program to draw ghosts on the screen. You must do this by writing a function called drawGhost, which takes three parameters, the center x location of the ghost, the center y location of the ghost and the color of the ghost.

Unit 7: Animation and Games (5 weeks/25 hours)

Objectives / Topics Covered	● Timers ● Randomizing Games ● Mouse Events ● Keyboard Events
Assignments / Labs	● 13 exercises in total ● 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

	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.

Unit 8: Project: Breakout (4 weeks/20 hours)

Objectives / Topics Covered	● Basic graphics ● Mouse events ● Collision detection
Assignments / Labs	● Guided exercises to build a Breakout Game ● The Breakout Game is made up of bricks at the top of the screen, a paddle that you control at the bottom of the screen, and a ball that bounces around. Your goal is to direct the paddle with your mouse to bounce the ball until all of the bricks have been hit and disappear.

Unit 9: Optional Supplemental Materials (Remainder of school year)

Objectives / Topics Covered	● Extra practice with: ○ Karel

Assignments / Labs	● Several additional exercises and large projects covering the topics listed above