CodeHS

North Carolina Introduction to Computer Science
Course Syllabus

Introduction

North Carolina Introduction to Computer Science introduces students to the foundational concepts of computer science and programming in JavaScript. With a unique focus on creative problem solving and real-world applications, students are challenged to explore how computing and technology can impact the world.

Course Overview

Prerequisites: There are no official prerequisites for the North Carolina Introduction to Computer Science course. This course is meant to be a first-time introduction to computer science and does not require students to come in with any computer programming experience.

Learning Environment: The course utilizes a blended classroom approach. The content is a mix of web-based and physical activities. Students will write and run code in the browser, create websites and digital artifacts, and engage in in-person collaborative exercises with classmates. 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, written programming exercises, free response exercises, collaborative creation projects, and research projects.

Programming Environment: Students write and run programs in the browser using the CodeHS editor. Students will be able to write both text-based and block-based JavaScript programs. Students gain programming experience early on in the course that will enable them to explore the rest of the course topics through computational thinking practices.

Course Resources: Access to a computer and high-speed internet is required. There is also an online textbook available for many modules and topics which can be accessed through the lesson plans or at https://codehs.gitbooks.io/introcs/content/

Course Breakdown

Module 1: Introduction to Programming with Karel the Dog (3 weeks, 15 hours)
This course begins with a strong focus on programming in order to allow students to create computational artifacts early on in the course. Students will be able to use their knowledge of programming to explore future topics in the course.

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: Programming with JavaScript (2 weeks, 10 hours)
This module introduces students to the basics of JavaScript, including variables, user input, control structures, functions with parameters and return values, and basic graphics, how to send messages to objects.

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 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.

Module 3: JavaScript Control Structures (2 weeks, 10 hours)
In this module, students learn how to use booleans and logical operators with control structures to make more advanced programs in JavaScript.

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 4: Basic Data Structures (2 weeks, 10 hours)

Students learn about lists and arrays which are essential basic data structures that any program will use.

Objectives / Topics Covered	● List/Array creation and basic operations ● Iterating through lists/arrays ● Finding and removing elements in lists/arrays
Assignments / Labs	● Basic list/array operations ○ Example Exercise: List of Places to Travel Create an array of the top 5 places you would like to travel called travelList. Print out the item at index 2. ● Iterating through arrays/lists ○ 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 generateBarcode function for you that creates a random barcode. Your job is to write the drawBarcode function. ● Basic list/array algorithms

Module 5: Digital Information (3 weeks, 15 hours)

In this module, students will learn about the various ways we represent information digitally as well as the physical elements of computers and networking, such as motherboards, RAM and routers. They will also learn about cybersecurity and the importance of copyright and creative credit.

Objectives / Topics Covered	● Internal Components of a Computer ● Peripheral Devices ● Network Devices ● Storage and Network Options ● Network Communication ● Network Management
Example Assignments / Labs	● Decimal to Binary ● Hexadecimal ● Bits to ASCII ○ Hello World in Bits ● Different Types of CPU ● RAM vs. Hard Drive ● 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? ● 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

Module 6: Creative Development (2-4 weeks, 10-20 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.

Objectives / Topics Covered	● What makes an engaging interface? ● Various User Interface (UI) Design techniques ● Accessibility issues ● Readability ● Rapid prototyping

	● User testing
Assignments / Labs	● Example exercises: ○ Research existing user interfaces ○ Assess the user interfaces of various web sites ○ 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 ■ Find and present an article about a particular UI design technique ■ Create your own live examples using this technique