California Software and Systems Development II Course Syllabus

High School - One Year (125 hours)

Course Overview and Goals

The California Software and Systems Development II course builds on foundational programming skills, introducing students to computer graphics, algorithm design, databases, physical computing, and artificial intelligence. Through hands-on projects, students will develop technical and career readiness skills for software development.

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. Several 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 HTML, CSS, and JavaScript in the browser using the CodeHS editor. Due to the fact that different browsers treat HTML and CSS differently, 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/

Prerequisites

Students that take the CodeHS California Software and Systems Development I course will be prepared to take this course. Students with some familiarity with building webpages using HTML, CSS, and JavaScript may also take this course.

Technology Requirements

To complete all activities and exercises in this course, students must have access to the 3rd party sites and tools listed here: CA Software and Systems Development II Course Links

More Information

Browse the content of this course at https://codehs.com/course/26141/explore

Course Breakdown

Module 1: Physical Computing (15 hours/3 weeks)

This unit introduces students to the foundations of computer science by exploring computer organization, digital logic, programming languages, and binary systems. Through interactive lessons and hands-on projects with the micro:bit, students learn core programming concepts like loops, conditionals, functions, and working with sensors

Topics Covered	● Computer Organization ● Digital Logic ● Digital Information ● Setting Up your micro:bit ● Programming with micro:bit
Example Assignments	● Memory System Explorer ○ In this activity, students use the Memory System Explorer to learn more about how a computer organizes, stores, and accesses data. ● Dice Simulator ○ Program the micro:bit to display a random number like a digital die. ● Blinking First Letter ○ Control an LED to blink when a micro:bit button is pressed.

Module 2: JavaScript Objects (15 hours/3 weeks)

Students learn about objects in graphics, how to create object properties and methods, iterate through them, and build constructors.

Topics Covered	● Intro to the Canvas and Graphics ● More Graphics Objects ● Intro to Objects ● Object Methods ● Object Constructors ● UML Object Diagrams
Example Assignments	● Create Your Meme ○ Students use graphics objects and text to design a personalized meme. ● Movie Database ○ Build an object to store and display information about movies. ● Fireflies ○ Animate glowing fireflies on the canvas using object properties. ● Bank Account ○ Simulate a simple bank account using object methods. ● New Person ○ Practice using object constructors to create a person profile. ● Create a UML Diagram ○ Students read a JavaScript constructor function and draw the matching UML class diagram.

Module 3: Sets and Grids (10 hours/2 weeks)

Students learn about sets, and grids. These are the essential basic data structures that any program will use.

Topics Covered	● Intro to Sets ● Intro to Grids ● Looping Over a Grid ● Grid Example: Get a Row

	● Building Tic Tac Toe
Example Assignments	● Mutual Friends ○ Students will use set operations to identify mutual friends between different users, simulating real-world applications of data comparison. ● Total Network of Friends ○ Students will combine multiple sets to find the full network of unique friends, reinforcing concepts of union and data aggregation. ● Building a Database ○ Students will organize data into a grid structure, creating a basic simulation of how databases store and access information. ● Summing Grid ○ Students will practice nested loops by calculating the total of all values in a grid, applying both iteration and data analysis skills. ● Check Win ○ Students will develop logic to determine winning conditions in a Tic Tac Toe game, using grid traversal and conditional statements.

Module 4: Artificial Intelligence (10 hours/2 weeks)

This unit explores the foundations of artificial intelligence, including how machines learn, reason, and interact with humans. Students investigate key concepts like supervised, unsupervised, and reinforcement learning, neural networks, and fuzzy logic, while applying AI principles to real-world scenarios like games, robotics, and intelligent systems.

Topics Covered	● Human and Artificial Intelligence ● Supervised Learning ● Unsupervised Learning ● Reinforcement Learning ● Intelligent Computing ● Fuzzy Logic ● Bayesian Inferences ● Intelligent Computing
Example Assignments	● The Turing Test as a Measure of Intelligence ○ Students will explore and reflect on how the Turing Test is used to evaluate whether machines can exhibit human-like intelligence. ● Exploration: Neural Networks ○ Students will interact with a simplified model of a neural network to understand how machines learn patterns from input data. ● Explore a Teachable Machine Project ○ Students will experiment with Google's Teachable Machine, training an AI model and testing its ability to recognize objects or sounds. ● Game: Hot & Cold ○ Students will simulate reinforcement learning by designing an AI that learns to find a hidden object based on feedback. ● Design an Intelligent Machine ○ Students will apply their knowledge of AI by brainstorming and proposing a design for a machine that performs an intelligent task in the real world.

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

Students investigate an environmental issue of their interest and develop an innovative and sustainable solution to address the issue.

Topics Covered	● Intro Computing and the Environment ● Oil in the Cloud
Example Assignments	● Computing and the Environment ○ Students will explore how computing technologies impact the environment, including energy use and e-waste. ● Reducing Your Digital Carbon Footprint ○ Students will evaluate their personal digital habits and identify ways to reduce their environmental impact. ● Email vs Postal Mail ○ Students will compare the environmental costs of digital and physical communication to determine which is more sustainable. ● Oil in the Cloud Report ○ Students will analyze a report on data center energy consumption and reflect on the hidden costs of cloud computing. ● Put Your Artifact Together ○ Students will design and create a digital artifact that presents a researched solution to a real-world environmental problem related to computing.

Module 6: Algorithms (10 hours/2 weeks)

This module introduces key algorithm concepts, including how algorithms are designed, used, and evaluated. Students explore searching, sorting, recursion, and informal run-time analysis to understand algorithm efficiency and problem-solving strategies in computing.

Topics Covered	● Introduction to Algorithms ● Searching Algorithms ● Sorting Algorithms ● Introduction to Recursion ● Informal Run-Time Analysis
Example Assignments	● Build an Algorithm ○ Students will practice designing a step-by-step algorithm to solve a real-world problem, focusing on clarity and logic. ● Search the Fruit Array ○ Students will apply linear search techniques to locate specific values within an array of fruit names. ● Insertion Sort Simulation ○ Students will step through the insertion sort process visually to understand how elements are reordered in a list. ● Recursion Exercise ○ Students will write a recursive function to solve a simple problem, learning how functions can call themselves to repeat tasks. ● Evaluating Algorithms with Statement Execution Counts ○ Students will estimate the efficiency of different algorithms by counting how many steps or statements they execute.

Module 7: Software Security (15 hours/3 weeks)

Students learn what happens when running a web application and how to look inside web apps using developer tools, source code, and more. Students learn basic SQL and common attacks like SQLi, and by the end of the module, students will be able to recommend solutions for flawed security systems.

Topics Covered	● Inside Web Apps ● Developer Tools ● The Value of Data ● What is SQL? ● Filtering Queries in SQL
Example Assignments	● Using View Page Source ○ Students will investigate the structure of a webpage by viewing and analyzing its underlying HTML using browser tools. ● Local Storage ○ Students will explore how websites store data locally in the browser and how this affects user experience and privacy. ● US Recorded Music Revenue ○ Students will interpret a real-world data visualization to understand how digital trends have impacted the music industry over time. ● Your First SELECT Statement ○ Students will write their first SQL query to retrieve information from a database using the SELECT command. ● Ravenclaw House ○ Students will apply filtering in SQL using the WHERE clause to extract data specific to characters from Ravenclaw in a fictional database.

Module 8: Databases (15 hours/3 weeks)

Students explore the role that data plays in creating websites. Students will learn about the various ways that data is taken from webpages, as well as ways to secure themselves from unwanted data collection. Students also learn how to incorporate data collection into their own websites, and collect simple information from users.

Topics Covered	● Databases ● Storing Data with Firebase ● Using APIs ● Exporting and Importing Data ● Designing Databases for Business
Example Assignments	● Database Relationships ○ Students will explore how different tables in a database are connected using keys, and understand how relationships improve data organization and retrieval. ● Storing and Collecting Data with Firebase ○ Students will use Firebase to store real-time data and practice reading from and writing to a cloud database. ● Creating a Dictionary API ○ Students will design their own API to simulate how applications fetch and serve dictionary data to users. ● Import Your Playlist

	○ Students will learn how to import structured data into a project and use it to populate features like a digital playlist. ● Design Your Database Schema ○ Students will apply their understanding of entities and attributes to create a schema for a business-related database project.

Module 9: Project: Collecting Data (10 - 15 hours/2 - 3 weeks)

Students create a multi-file webpage that tracks the number of clicks that items of content on the site receive, allowing the web owner to make decisions about which content should be kept, and which should be changed out the next time that they update their site. Students test out one another's webpages and write a written response highlighting the content that they will change on the next iteration and why.

Topics Covered	● Define Your Product ● Develop Hypotheses ● Create a User Persona ● Create Your Data Collection Site ● Collecting User Data ● Analyzing User Data ● Making Decisions
Example Assignments	● Create Your Own User Persona ○ Students create a user persona for their website to make predictions about how users will interact with their site, and how to tailor their site to the needs of potential users. ● Data Collection Site ○ Students will create a website that tracks user data. They will use this data to make informed decisions about how to improve their website in future lessons.

Module 10: Computer Science Careers (10 - 15 hours/2 - 3 weeks)

Students take some time to explore and discover different computer science careers and professional student organizations. They will also examine inclusive coding, how to avoid bias in computer programming, and develop their own interactive resume as their final project.

Topics Covered	● Computer Science Careers ● Creating a Resume ● Layout and Web Design Theory ● Interactive Resume Development
Example Assignments	● CS Careers ○ Students will explore a wide range of computer science career paths and identify roles that align with their own interests and skills. ● Coding in the Wild ○ Students will examine real-world examples of how computer science is applied across industries like health, entertainment, and transportation. ● Draft Your Resume ○ Students will begin crafting a traditional resume by organizing their academic background, skills, and experiences. ● Explore Resume Layouts ○ Students will study common resume layout strategies such as

	Z-pattern and F-pattern to improve visual organization and readability. ● Final Draft Interactive Resume ○ Students will create and publish an interactive web-based resume that showcases their skills using HTML, CSS, and thoughtful design choices.