South Carolina Fundamentals of Computer Science Syllabus

High School (165 Contact Hours)

Course Overview and Goals

This course introduces students to the field of computer science through exploration of a wide range of computing topics. Using critical thinking and creativity, students will design and build computational artifacts through both individual and team projects. Along the way, they'll develop foundational knowledge in computer operation, programming, web design, data, physical computing, machine learning, and the societal and ethical dimensions of computing. The course also connects students to computing careers and the role computing plays across all other career fields.

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.

Programming Environment

For the programming and web design units, students write and run programs in the browser using the CodeHS editor. For the micro:bit unit, students write and run programs in the browser using the MakeCode editor and will download their programs to their micro:bit devices for further testing.

Prerequisites

This 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.

Technology Requirements

To complete all activities and exercises in this course, students must have access to the 3rd party sites and tools listed here: South Carolina Fundamentals of Computing Course Links

More Information

Browse the content of this course at https://codehs.com/course/28010

Course Breakdown

The full standards alignment can be found at
https://codehs.com/standards/framework/South_Carolina_Fundamentals_of_Computing/mapping/2213

Module 1: Digital Citizenship and Cyber Hygiene (3 weeks / 15 hours)

Students explore internet etiquette and how to stay safe online. They examine the potential effects of their digital footprints, how to protect personal information from online risks, and the social and legal implications of cyberbullying and digital misconduct.

Objectives / Topics Covered	• Digital Footprint • Cyberbullying • Internet Safety • Privacy and Security • Information Literacy • Copyright • Cyber Ethics • PSA Project

Example Assignments / Labs	• **Building a Positive Digital Footprint**
	  ○ Students reflect on how their online actions contribute to a lasting digital reputation and identify practices for building a positive digital presence.
	• **Cyber Ethics Scenarios**
	  ○ Students analyze real-world ethical dilemmas related to online behavior and digital law, evaluating consequences and responsibilities.
	• **Public Service Announcement Project**
	  ○ Students research a digital citizenship topic, select a target audience, and create an original PSA to inform others about responsible technology use.

Module 2: Cybersecurity Foundations (2 weeks / 10 hours)

Students explore the core principles of cybersecurity, including the CIA Triad, symmetric and asymmetric cryptography, and authentication methods. They examine the ethics of hacking and consider career pathways in cybersecurity.

Objectives / Topics Covered	• What is Cybersecurity? • The CIA Triad • Cryptography: Then, Now, and Future • Asymmetric Encryption • Authentication Methods • Hacking Ethics and Legality

Example Assignments / Labs	• **Enhanced Caesar Cipher**
	  ○ Students apply concepts of symmetric encryption by encoding and decoding messages using a modified Caesar cipher.
	• **Diffie-Hellman Visualizer Reflection**
	  ○ Students use an interactive visualizer to explore how two parties can establish a shared secret key over an unsecured channel.
	• **Reflection: Penetration Testing**
	  ○ Students examine the ethics and methods of authorized penetration testing and discuss how it contributes to cybersecurity defense.

Module 3: Data Literacy and Visualization (1 week / 5 hours)

Students learn to read and create data visualizations, examine how data is collected and its limitations, and apply these skills by building an original data dashboard from real data.

Objectives / Topics Covered	• Data visualization • Data collection • Misleading data • Data dashboards

Example Assignments / Labs	• **Design a Survey**
	  ○ Students plan a data collection survey, identifying potential sources of bias and describing limitations in their methodology.
	• **Screen Time Usage Dashboard**
	  ○ Students create a data dashboard visualizing real screen-time data and analyze the patterns and insights their chosen visualizations reveal.

Module 4: What is Computing? (1-2 weeks / 5-10 hours)

Students review the history of computing, explore the hardware and software components of modern computers, and investigate the impact computing has had on the world today and the potential impacts it may have in the future.

Objectives / Topics Covered	• Computer history • Computer organization • Software • Hardware • Artificial intelligence

Example Assignments / Labs	• **Label Your Computer**
	  ◦ Students identify and label the hardware components of a computer, connecting physical parts to their functions within the system.
	• **Hardware vs. Software**
	  ◦ Students categorize computing components and explain how hardware and software interact to execute programs.
	• **AI: Is It a Bad Thing?**
	  ◦ Students research and debate the societal impacts of artificial intelligence, citing current examples to support their position.

Module 5: Programming with Karel (5 weeks / 25 hours)

Students learn foundational programming concepts by writing commands for Karel the Dog in a grid-based world. Topics include functions, loops, conditionals, decomposition, and debugging, building a strong foundation for JavaScript programming.

Objectives / Topics Covered	• Karel basics • Functions • Top-down design • Code style • For loops • Conditionals • Debugging • While loops

Example Assignments / Labs	• **Pyramid of Karel**
	  ◦ Students write a program to move Karel and place tennis balls in a pyramid shape, applying sequencing and function decomposition.
	• **Lots of Hurdles**
	  ◦ Students use for loops to program Karel to jump over a series of hurdles, reinforcing the concept of iteration.
	• **Random Hurdles**
	  ◦ Students combine while loops and conditionals to navigate Karel through a randomly generated set of obstacles.
	• **Staircase**
	  ◦ Students apply top-down design and functions to build a staircase pattern, practicing decomposition of complex problems.

Module 6: Karel Challenges (1-2 weeks / 5-10 hours)

Students apply all Karel programming skills to a set of open-ended challenge problems, combining functions, loops, and conditionals to solve increasingly complex grid puzzles.

Objectives / Topics Covered	• Function decomposition • Loops and conditionals • General solutions • Creative problem solving

Example Assignments / Labs	• **Fetch**
	  ○ Students program Karel to retrieve and place tennis balls across the board using a combination of loops and conditionals.
	• **Tower Builder**
	  ○ Students write a program that constructs towers of varying heights, requiring functions and control structures to solve.
	• **Racing Karel**
	  ○ Students write an efficient Karel program to complete a race course, practicing clean and concise code design.

Module 7: JavaScript Basics (1-2 weeks / 5-10 hours)

Students learn the fundamentals of JavaScript including variables, user input, arithmetic, random numbers, and basic functions. They write interactive console programs that respond to user data.

Objectives / Topics Covered	• Output and printing • Variables • User input • Math operations • Pair programming • Random numbers • Basic Functions

Example Assignments / Labs	• **Mad Lib**
	  ○ Students use variables and user input to build a customizable Mad Lib story program, practicing string concatenation and data collection.
	• **T-Shirt Shop**
	  ○ Students write a program that calculates the cost of a t-shirt order, applying arithmetic operators and variables to a real-world scenario.
	• **Digital Business Card**
	  ○ Students combine variables and formatted string output to generate a personalized digital business card.

Module 8: The Canvas and Graphics (1 week / 5 hours)

Students learn how to create and position graphics objects on the CodeHS canvas using JavaScript. They combine shapes, colors, and coordinates to produce original visual programs.

Objectives / Topics Covered	• Canvas coordinates • Basic shapes • Images and text • Proportional positioning

Example Assignments / Labs	• **Raise the Flag**
	  ○ Students use graphics objects and coordinate positioning to draw an original flag design on the canvas.
	• **Create Your Meme**
	  ○ Students combine image and text graphics objects to produce a meme, applying layering and positioning skills.
	• **Color the Rainbow**
	  ○ Students draw a multi-colored arc pattern on the canvas, practicing color values, shape sizing, and coordinate math.

Module 9: Graphics Challenges (1 week / 5 hours)

Students apply canvas and graphics programming skills to open-ended creative challenges, designing original compositions using shapes, colors, and positioning.

Objectives / Topics Covered	• Function decomposition • Const variables • Creative graphics

Example Assignments / Labs	• **Ghost**
	  ○ Students program a ghost graphic using layered shapes and styling, planning their composition before coding.
	• **Draw Something**
	  ○ Students create an original canvas drawing of their choice, demonstrating mastery of graphics objects and positioning.

Module 10: Physical Computing with micro:bit (1-2 weeks / 5-10 hours)

Students explore physical computing by programming the BBC micro:bit to interact with sensors and real-world inputs. They apply the design thinking process to identify a problem and develop a physical computing prototype.

Objectives / Topics Covered	• Intro to micro:bit and MakeCode • Sensors and APIs • Design cycle and prototyping

Example Assignments / Labs	• **Temperature Monitor**
	  ○ Students program the micro:bit to read temperature sensor data and display a response when the temperature exceeds a threshold.
	• **Light Detector**
	  ○ Students use the micro:bit's light sensor to detect changes in ambient light and trigger a corresponding output.
	• **Develop the Prototype**
	  ○ Students design and build a physical computing solution using the micro:bit to address a user-identified real-world problem.

Module 11: Networks and the Internet (1 week / 5 hours)

Students explore the structure and design of the internet and how that design affects the reliability of network communication, the security of data, and personal privacy. They trace how data travels from sender to receiver across interconnected networks.

Objectives / Topics Covered	• The Internet • Network hardware • IP addressing • Domain Name System • Routing • Packets and protocols • Societal impact

Example Assignments / Labs	• **IPv4 vs IPv6**
	  ○ Students compare the two major internet addressing systems and analyze why the transition from IPv4 to IPv6 is necessary.
	• **Redundancy**
	  ○ Students investigate how packet routing and redundant network paths ensure reliable communication even when parts of the network fail.
	• **What is the Digital Divide?**
	  ○ Students research disparities in internet access around the world and reflect on the equity implications of

	unequal connectivity.

Module 12: Project: IT Professional (1 week / 5 hours)

Students explore cybersecurity career pathways and develop professional communication skills used in IT roles, including technical writing and instructional video production.

Objectives / Topics Covered	• Customer service • Knowledge base writing • Instructional video production

Example Assignments / Labs	• **Draft a KB Article**
	  ○ Students write a knowledge base article explaining a technical concept, applying professional writing conventions for an IT audience.
	• **Record Your Video**
	  ○ Students create a short instructional video on a technology topic, planning a script and storyboard to simulate real workplace documentation tasks.

Module 13: Web Design (4 weeks / 20 hours)

Students learn HTML and CSS and apply them to build multi-page websites. Topics progress from basic HTML structure and text formatting to CSS selectors, styling, and linking external style sheets, culminating in a complete personal website project.

Objectives / Topics Covered	• HTML tags and structure • Text formatting • Links and images • Lists and tables • HTML and CSS styling • Multi-file websites • How websites work

Example Assignments / Labs	• **Personal Library**
	  ○ Students build an HTML page listing their favorite books, using lists, links, and images to structure and enhance the content.
	• **Restaurant Menu**
	  ○ Students apply CSS selectors and styling rules to design a visually styled restaurant menu web page.
	• **Your First Website**
	  ○ Students create a complete multi-page personal website, combining HTML structure and a linked CSS style sheet to demonstrate mastery of web design

	fundamentals.

Module 14: Intro to AI (2 weeks / 10 hours)

Students develop an understanding of the main concepts and vocabulary around artificial intelligence. They explore how AI models are trained, how they are applied across industries, and how bias can emerge in training data and models.

Objectives / Topics Covered	• Intelligence and AI • Machine learning • AI in industry • Bias in AI

Example Assignments / Labs	• **Explore and Reflect: Test the Intelligence of an AI**
	  ○ Students interact with an AI system and evaluate its responses against criteria for human-like intelligence, connecting to concepts from the Turing Test.
	• **AI in Industry Bias**
	  ○ Students examine documented cases of algorithmic bias in real-world AI applications and discuss the ethical and social implications.
	• **Turning on a Light: Apply the Model**
	  ○ Students apply a simple AI decision-making model to a real-world scenario, tracing how the model uses inputs to produce an output.

Module 15: Machine Learning (1-2 weeks / 5-10 hours)

Students learn the fundamentals of machine learning, covering the ML lifecycle and the three major learning paradigms: supervised, unsupervised, and reinforcement learning. They explore how bias can affect learning models and interact with real ML tools.

Objectives / Topics Covered	• Neural networks • Supervised learning • Unsupervised learning • Reinforcement learning

Example Assignments / Labs	• **Apples vs Bananas**
	  ○ Students train a supervised learning model to classify images and evaluate how training data size affects model accuracy.
	• **How to Teach an AI to Play a Game**
	  ○ Students explore reinforcement learning by observing and reflecting on how an AI agent learns a task through

	trial, error, and reward signals.

Module 15: Final Project (1-2 weeks / 5-10 hours)

Students apply design thinking to plan, prototype, and build an original web page. They create low-fidelity prototypes, conduct user testing, incorporate feedback, and then develop their final project using HTML, CSS, and JavaScript.

Objectives / Topics Covered	• Intro to Design Thinking • User Interface Design Principles • Rapid Prototyping • User Testing • Iterative Development

Example Assignments / Labs	• **Make Your Paper Prototypes!**
	  ○ Students create low-fidelity paper prototypes of their project interface and prepare questions and tasks for user testing sessions.
	• **Test and Revise**
	  ○ Students conduct user testing with peers, collect feedback, and use it to improve their prototype before moving to development.
	• **Write the Code!**
	  ○ Students develop their final project web page, applying HTML, CSS, and JavaScript skills practiced throughout the course.

(Optional Module) Career and Workplace (10-15 hours)

Students explore computing career pathways, student organizations, certification opportunities, and workplace skills. They practice professional communication through resume writing and mock interviews, and learn essential personal safety practices in technology environments. Students taking more than one CodeHS course in their pathway may have already completed some of these activities in a previous course.

Topics Covered	• Computing careers • Student organizations • Certifications • Resume and interviewing • Workplace safety

Supplementary Module Guide:

These units can be used during the course for added practice or after the course has been completed for further review.

Supplementary Module	**Prerequisite/Recommended Module(s)**	**# of activities**
Extra Karel Practice	Programming with Karel Karel Challenges	12
Extra Karel Puzzles	Programming with Karel Karel Challenges	11
Functions and Parameters Practice	Functions, Events and Lists	8
Animation and Games - Extended User Interaction - *Mouse and Key Events* - *Drawing Lines* - *Crazy Ball Game*	Functions, Events and Lists	26
Extended Array Concepts - *Indexing Into an Array* - *Finding an Element* - *Removing an Element*	Functions, Events and Lists	12
Extended Data Structures - *Objects/Maps* - *Iterating Over an Object* - *Sets and Grids* - *Looping over a Grid*	Functions, Events and Lists	40
Web Design - HTML - Formatting Text - Links, Images, Lists, Tables - CSS by Tag, Class, ID	No prerequisites	75