Computing Ideas Scope and Sequence

The Computing Ideas course is a first computer science course introducing the basics of programming with Karel the Dog, the basics of designing a web page, and how information and images are represented with computers. Students will learn to code using blocks to drag and drop, but they can switch between blocks and text as desired. Students will create a portfolio on the web of projects they build throughout the course.

With a unique focus on creativity, problem solving and project based learning, Computing Ideas gives students the opportunity to explore several important topics of computing using their own ideas and creativity and develop an interest in computer science that will foster further endeavors in the field.

Module 1: Introduction to Programming with Karel the Dog	
50 hours (10 weeks)	Students learn what it means to program. Students utilize problem decomposition, control structures, and functions to solve puzzles with Karel the Dog.
CSTA Standards Addressed 2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-14 Create procedures with parameters to organize code and make it easier to reuse. 2-AP-19 Document programs in order to make them easier to follow, test, and debug. 3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. 3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. 3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. 3A-AP-18 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	

3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs. 3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects. 3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.

Module 2: What is Computing?	
25 hours (5 weeks)	Students explore the different types of computers and how they affect their everyday lives and discuss relevant applications of computing.
CSTA Standards Addressed 2-IC-20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. 3A-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers. 3B-AP-08 Describe how artificial intelligence drives many software and physical systems. 3B-IC-27 Predict how computational innovations that have revolutionized aspects of our culture might evolve.	

Module 3: Web Design	
30 hours (6 weeks)	Students learn HTML, CSS, and the processes involved in viewing web pages on the Internet. Students create several simple web pages using the CodeHS online editor, including their own website about themselves.
CSTA Standards Addressed 2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution. 2-NI-04 Model the role of protocols in transmitting data across networks and the Internet. 3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. 3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	

<table>
<tr><th colspan="2">Module 4: Digital Information</th></tr>
<tr><td>25 hours (5 weeks)</td><td>Introduces the various ways we represent information digitally. Students learn about pixels, binary, hexadecimal, colors, and digital image representations.</td></tr>
<tr><td colspan="2"><b>CSTA Standards Addressed</b>
2-DA-07 Represent data using multiple encoding schemes.
3A-DA-09 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.
3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</td></tr>
</table>

<table>
<tr><th colspan="2">Module 5: The Internet</th></tr>
<tr><td>25 hours (5 weeks)</td><td>Explores the structure and design of the internet, the security of data, and personal privacy. Students will learn about the social and ethical implications of the Internet.</td></tr>
<tr><td colspan="2"><b>CSTA Standards Addressed</b>
2-NI-04 Model the role of protocols in transmitting data across networks and the Internet.
2-IC-20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.
2-IC-23 Describe tradeoffs between allowing information to be public and keeping information private and secure.</td></tr>
</table>