vidcode Computer Science I

In Vidcode's Computer Science I course (made up of Creative Coding I and II), students learn the fundamentals of computer science and programming in JavaScript. Special attention is given to encouraging students to identify as a programmer and using code to enhance and personalize visual media. Students also learn about different applications of JavaScript programming including interactivity, algorithms, game creation and data art.

The courses teach students beginning to advanced JavaScript concepts including **functions, variables, objects, loops, and conditionals**.

Creative Coding aligns with the TEKS Computer Science I course and includes the following:

§126.33. Computer Science I	Vidcode Creative Coding
(1) Creativity and innovation. The student develops products and generates new understandings by extending existing knowledge. The student is expected to: (A) participate with electronic communities as a learner, initiator, contributor, and teacher/mentor; (B) extend the learning environment beyond the school walls with digital products created to increase teaching and learning in the other subject areas; and (C) participate in relevant, meaningful activities in the larger community and society to create electronic projects.	Creative Coding I Units 1-4: Students share projects through Vidcode's online community. Creative Coding II Units 1-4: Students create interactive projects and publish them online.
(4) Critical thinking, problem solving, and decision making. The student uses appropriate strategies to analyze problems and design algorithms. The student is expected to: (A) use program design problem-solving strategies to create program solutions; (B) define and specify the purpose and goals of solving a problem; (C) identify the subtasks needed to solve a problem; (D) identify the data types and objects needed to solve a problem; (E) identify reusable components from existing code; (F) design a solution to a problem; (G) code a solution from a program design; (H) identify and debug errors; (I) test program solutions with appropriate valid and invalid test data for correctness; (J) debug and solve problems using error messages, reference materials, language documentation, and effective strategies; (L) analyze and modify existing code to improve the underlying algorithm;	Creative Coding I Unit 1: Students use programming for problem-solving and learn the importance of sequence in programming. Unit 2: Students identify reusable components in their code and store them in variables. Unit 3: Students identify and debug errors in their programs. Students also learn to use arithmetic operators and the math library to generate and use random numbers. Unit 4: Students create more complex programs using branching conditionals.

(M) create program solutions that exhibit robust behavior by understanding, avoiding, and preventing runtime errors, including division by zero and type mismatch; (N) select the most appropriate algorithm for a defined problem; (O) demonstrate proficiency in the use of the arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division; (P) create program solutions to problems using available mathematics libraries, including absolute value, round, power, square, and square root; (Q) develop program solutions that use assignment; (R) develop sequential algorithms to solve non-branching and non-iterative problems; (S) develop algorithms to decision-making problems using branching control statements; (T) develop iterative algorithms and code programs to solve practical problems; (W) generate and use random numbers.	Creative Coding II Unit 1: Students divide their programs into subtasks and identify components that can be abstracted and used in different situations. Unit 2: Students develop iterative algorithms for applying changes to lists of data. Unit 3: Students learn more important strategies for debugging and avoiding runtime errors. Unit 4: Students select the most appropriate algorithm to solve a variety of problems.
(6) Technology operations, systems, and concepts. The student understands technology concepts, systems, and operations as they apply to computer science. The student is expected to: (P) demonstrate an understanding of the concept of a variable; (Q) demonstrate an understanding of and use reference variables for objects; (S) demonstrate an understanding of the concept of scope; (T) identify and use the structured data type of one-dimensional arrays to traverse, search, and modify data; (U) choose, identify, and use the appropriate data type and structure to properly represent the data in a program problem solution;	Creative Coding I Unit 1: Students begin using variables as references for objects. Unit 2: Students implement arrays to solve problems Unit 4: Students identify the scope of a variable. Creative Coding II Units 1-4: Students demonstrate use of appropriate data type and structure to solve a given problem.