Technology Applications TEKS Review Work Group D Recommendations

Strand: Computational Thinking									
Substrand: Foundations									
Kindergarten	**Grade 1**	**Grade 2**	**Grade 3**	**Grade 4**	**Grade 5**	**Grade 6**	**Grade 7**	**Grade 8**	**Rationale**
Computational thinking--foundations. The student explores the core concepts of computational thinking, a set of problem-solving processes that involve decomposition, pattern recognition, abstraction, and algorithms. The student is expected to:									To best align to the core concepts of computational thinking and computer science, the work group has decided to split the computational thinking strand into foundations and applications. The current student expectations were broad and did not explicitly address computational thinking; therefore, the work group incorporated some concepts into their recommendations.
NEW (A) identify a problem or task such as making a sandwich and break it down (decompose) into smaller pieces	NEW (A) identify and discuss a problem or task and break down (decompose) the solution into sequential steps	NEW (A) identify and communicate a problem or task and break down (decompose) multiple solutions into sequential steps	NEW (A) decompose story problems into smaller, manageable subproblems and identify a solution to the problem	NEW (A) decompose story problems into smaller, manageable subproblems and discuss and document various solutions to the problem	NEW (A) decompose a real-world problem into smaller, manageable subproblems using graphic organizers such as learning maps, concept maps, or other representations of data	NEW (A) decompose real-world problems into structured parts by using visual representation	NEW (A) decompose real-world problems into structured parts by using flowcharts	NEW (A) decompose real-world problems into structured parts by using pseudocode	Students are exposed to the foundations of decomposition so that when they get to grades 3-5, they can better represent the process. The student expectations listed scaffold into grade levels 3-5 by introducing identification and decomposition of the problem and its solutions. Students are introduced to the idea that there can be multiple solutions to a problem and different ways to accomplish a task. This concept shows students how to analyze a problem and break it down in different ways in the context of computer science. These student expectations scaffold into deeper thinking in middle school. The work group would like to scaffold logical sequences that build into computer science concepts. The student expectations listed scaffold into grade levels 6-8 by introducing real-world problems and multiple documentation methods. This concept shows students how to analyze a problem and break it down in different ways in the context of computer science.

Technology Applications TEKS Review Work Group D Recommendations

Strand: Computational Thinking									
Substrand: Foundations									
Kindergarten	**Grade 1**	**Grade 2**	**Grade 3**	**Grade 4**	**Grade 5**	**Grade 6**	**Grade 7**	**Grade 8**	**Rationale**
NEW (B) identify simple patterns and make predictions based on the pattern	NEW (B) identify the simple patterns found in the solutions to everyday problems or tasks	NEW (B) identify complex patterns and make predictions based on the pattern	NEW (B) identify patterns in story problems	NEW (B) identify patterns in story problems and make predictions based on the pattern	NEW (B) identify patterns in real-world problems and make predictions based on the pattern	NEW (B) analyze the patterns and sequences found in visual representations	NEW (B) analyze the patterns and sequences found in flowcharts	NEW (B) analyze the patterns and sequences found in pseudocode and identify its variables	The concept introduces the idea of simple and complex patterns. Additionally, students are given the foundation to recognize the connection found in patterns to create predictions in outcomes. The concept demonstrates how identifying patterns and sequences leads to expected outcomes in written or visual form. The concept demonstrates how identifying patterns and sequences leads to expected outcomes and variations in patterns and sequences may alter to different results.
						NEW (C) define abstraction and distinguish between generalized information versus specific information in the context of solving a problem or completing a task	NEW (C) identify abstraction and analyze how an algorithm the student created can be generalized to solve additional problems	NEW (C) practice abstraction by developing a generalized algorithm that can solve different types of problems	Abstraction is such a complex concept that the work group recommends placing it at only grades 6-8. The work group recognizes that abstraction can be correlated to multiple subject areas, not just computer science. TEKS guide: abstraction should be defined at these grade levels as generalization. Abstraction aligns vertically to CS II (c)(2)(g) and CS III (c)(2)(A).

Strand: Computational Thinking									
Substrand: Foundations									
Kindergarten	**Grade 1**	**Grade 2**	**Grade 3**	**Grade 4**	**Grade 5**	**Grade 6**	**Grade 7**	**Grade 8**	**Rationale**
NEW (C) identify algorithms (step-by-step instructions) using a sequential process such as first, next, then, and last	NEW (C) create a simple algorithm (step-by-step instructions) as it applies to an everyday task	NEW (C) create and troubleshoot simple algorithms (step-by-step instructions) that include conditionals such as if-then statements as they apply to an everyday task	NEW (C) debug simple algorithms (set of procedures) by identifying and removing errors	NEW (C) debug algorithms (set of procedures) by identifying and removing errors	NEW (C) compare multiple algorithms for the same task and determine which algorithm is the most appropriate for that task	NEW (D) analyze different techniques used in debugging and apply them to an algorithm	NEW (D) analyze different techniques used in debugging and apply them to an algorithm	NEW (D) develop, compare, and improve algorithms for a specific task to solve a problem	Students gain experience creating simple algorithms using simple everyday tasks as the basis for building foundational skills. Additionally, students are introduced to computer science vocabulary. The work group acknowledges that this will be new material for primary teachers. This concept is vertically aligned to the computer science standards and the CSTA Standards for Students. Algorithms are fundamental to the understanding of computer science and should be addressed. The work group would encourage instructional materials developers to provide examples of grade appropriate tasks. TEKS guide: provide a scenario that is grade level appropriate. For example, making a peanut butter and jelly sandwich, tying your shoes, brushing your teeth. The student expectations are repeated at grades 6 and 7 so that fundamentals of debugging are reinforced. Debugging is a version of problem solving. Vertically aligned with CSTA standards and Computer Science I (c)(4)(J). TEKS guide: debugging and other strategies are defined in the TEKS guide. Work group recommends that instructional materials developers include definition of debugging and provide examples of debugging strategies.

Technology Applications TEKS Review Work Group D Recommendations

Strand: Computational Thinking									
Substrand: Foundations									
Kindergarten	**Grade 1**	**Grade 2**	**Grade 3**	**Grade 4**	**Grade 5**	**Grade 6**	**Grade 7**	**Grade 8**	**Rationale**
						NEW (E) analyze the benefits of using iteration (code and sequence repetition) in algorithms	NEW (E) analyze the benefits of using iteration (code and sequence repetition) in algorithms	NEW (E) analyze the benefits of using iteration (code and sequence repetition) in algorithms	Students have been using loops since grade 2; therefore, in grades 6-8, the work group recommends challenging students the benefits of using iteration. The student expectation also aligns to code re-use. TEKS Guide: define iteration in terms of computer science "repetition of steps within a program to create efficiency"

Strand: Computational Thinking									
Substrand: Applications									
Kindergarten	**Grade 1**	**Grade 2**	**Grade 3**	**Grade 4**	**Grade 5**	**Grade 6**	**Grade 7**	**Grade 8**	**Rationale**
Computational thinking--applications. The student, with guidance from an educator, applies the fundamentals of computer science. The student is expected to:			Computational thinking--applications. The student applies the fundamentals of computer science. The student is expected to:						
		NEW (A) identify and explore what a variable is in a sequence of code	NEW (A) use variables within a program to store data	NEW (A) use variables within a program to modify data	NEW (A) use variables within a program to store and modify data.	NEW (A) define and label variables that relate to their programming or algorithm	NEW (A) manipulate and rename variables and describe different data types	NEW (A) construct named variables with multiple data types and perform operations on their values	The work group would like to introduce variables at grade 2 to build a foundation for future learning. The work group believes the concept of variables is not developmentally appropriate at kindergarten and grade 1. In alignment to the CSTA standards, students in grades 3-5 begin using variables in the context of programming. Recommendation for TEKS guide: storing data is the same thing as saving data. This concept vertically aligns with the Fundamentals of Computer Science (c)(4)(F) "demonstrate an understanding of and use variables within a program, story, game, or animation."

Strand: Computational Thinking									
Substrand: Applications									
Kindergarten	**Grade 1**	**Grade 2**	**Grade 3**	**Grade 4**	**Grade 5**	**Grade 6**	**Grade 7**	**Grade 8**	**Rationale**
NEW (A) identify and create a sequence of code with or without technology such as solving a maze using drag-and-drop programming or creating step-by-step directions for student movement to a specific location	NEW (A) create a sequence of code that solves a simple problem with or without technology	NEW (B) create a sequence of code that includes loops to solve a simple problem with or without technology	NEW (B) create programs that include sequences, loops, and conditionals to express ideas or address a problem	NEW (B) create programs that include sequences, loops, conditionals, and events to express ideas or address a problem	NEW (B) create programs that include sequences, loops, conditionals, and events to solve an everyday problem	NEW (B) create programs that address different subproblems within a real-world context	NEW (B) create programs with nested loops that address different subproblems within a real-world context	NEW (B) create programs and combine control structures, including nested loops and compound conditionals, that address real-world situations	Students develop familiarity with sequences of code and the idea that computer programs can be used to solve problems. The work group would like to acknowledge that technology is not always the only method of instruction for students in these grade levels to successfully gain understanding of this concept. At grades 3-5, students are introduced to foundational tools (sequences, loops, conditionals, and events) of computer science to develop their knowledge and ability to solve problems using programming and computational thinking. Students are expected to apply the foundational skills developed in earlier grades to address increasingly complex real-world problems and provide solutions.

Strand: Computational Thinking									
Substrand: Applications									
Kindergarten	**Grade 1**	**Grade 2**	**Grade 3**	**Grade 4**	**Grade 5**	**Grade 6**	**Grade 7**	**Grade 8**	**Rationale**
					NEW (C) analyze code and how it may be reused to develop new or improved programs.			NEW (C) modify and implement previously written code to develop new and improved programs	Students should understand the importance of efficiency in reusing previously written code. The work group believes this naturally occurs during the coding process. Students apply the concept of code reuse when appropriate. This concept is vertically aligned to computer science standards. Specifically, this aligns with Fundamentals of Computer Science (c)(4)(J). Students should practice and learn the skill. This skill is only in grades 5 and 8 because of the complexity of the topic. Students should gain an understanding of the concept but not overuse it. The ultimate goal is for students to get as much practice with the programming skills as possible.

Strand: Computational Thinking									
Substrand: Applications									
Kindergarten	**Grade 1**	**Grade 2**	**Grade 3**	**Grade 4**	**Grade 5**	**Grade 6**	**Grade 7**	**Grade 8**	**Rationale**
~~K-2.1.C explore virtual environments, simulations, models, and programming languages to enhance learning;~~			~~3-5.1.C use virtual environments to explore systems and issues.~~						
~~K-2.1.D create and execute steps to accomplish a task; and~~									
~~K-2.1.E evaluate and modify steps to accomplish a task.~~									
~~K-2.4.A identify what is known and unknown and what needs to be known regarding a problem and explain the steps to solve the problem;~~			~~3-5.4.A identify information regarding a problem and explain the steps toward the solution;~~			~~6.4.D use multiple processes and diverse perspectives to explore alternative solutions;~~	~~7.4.D use multiple processes and diverse perspectives to explore alternative solutions;~~	~~8.4.D use multiple processes and diverse perspectives to explore alternative solutions;~~	