

Computer Mathematics – Crosswalk (Summary of Revisions): 2016 *Mathematics Standards of Learning and Curriculum Framework*

Additions (2016 SOL)	Deletions from Computer Mathematics (2009 SOL)
<ul style="list-style-type: none"> COM.8 EKS – Construct a case statement COM.15 – Objects added to the list of data types 	<ul style="list-style-type: none"> COM.10 – [Combined with COM.15] COM.20 – [Included in COM.2]
Parameter Changes/Clarifications (2016 SOL)	Moves within Computer Mathematics (2009 SOL to 2016 SOL)
<ul style="list-style-type: none"> Many of the revisions to the standards and curriculum framework for Computer Mathematics focused on simplifying the language of the standards and updating vocabulary to modern programming language. Language included in standards that provided specific examples or situations, was often moved to the Essential Knowledge and Skills or Understanding the Standard sections of the Curriculum Framework. 	<ul style="list-style-type: none"> COM.2 EKS – Describe the interplay between hardware and software in program execution [Moved to COM.16 EKS] COM.6 – [Moved to COM.10] COM.7 – [Moved to COM.11] COM.8 – [Moved to COM.12] COM.9 – [Moved to COM.15] COM.10 - using appropriate variable data types, including representing structured data types [Moved to COM.15] COM.11 – [Moved to COM.16] COM.12 – [Moved to COM.6] COM.13 – [Moved to COM.7] COM.14 – [Moved to COM.8] COM.15 – [Moved to COM.13] COM.16 – [Moved to COM.14] COM.17 – [Moved to COM.9] COM.18 – [Moved to COM.17] COM.19 – [Moved to COM.18]

EKS = Essential Knowledge and Skills, referring to the column on the far right of the Curriculum Framework

EU = Essential Understandings, referring to the column on the far left of the Curriculum Framework

Comparison of Mathematics Standards of Learning – 2009 to 2016

2009 SOL		2016 SOL	
COM.1	The student will apply programming techniques and skills to solve practical real-world problems in mathematics arising from consumer, business, and other applications in mathematics. Problems will include opportunities for students to analyze data in charts, graphs, and tables and to use their knowledge of equations, formulas, and functions to solve these problems.	Problem Solving	
		COM.1	The student will design and apply computer programs to solve practical problems in mathematics arising from business and applications in mathematics.
COM.2	The student will design, write, test, debug, and document a program. Programming documentation will include preconditions and postconditions of program segments, input/output specifications, the step-by-step plan, the test data, a sample run, and the program listing with appropriately placed comments.	Program Design	
		COM.2	The student will design, write, document, test, and debug, a computer program.
COM.3	The student will write program specifications that define the constraints of a given problem. These specifications will include descriptions of preconditions, postconditions, the desired output, analysis of the available input, and an indication as to whether or not the problem is solvable under the given conditions.	COM.3	The student will write program specifications that define the constraints of a given problem.
COM.4	The student will design a step-by-step plan (algorithm) to solve a given problem. The plan will be in the form of a program flowchart, pseudo code, hierarchy chart, and/or data-flow diagram.	COM.4	The student will design an algorithm to solve a given problem.
COM.5	The student will divide a given problem into manageable sections (modules) by task and implement the solution. The modules will include an appropriate user-defined function, subroutines, and procedures. Enrichment topics might include user-defined libraries (units) and object-oriented programming.	COM.5	The student will divide a given problem into modules by task and implement the solution.
		COM.6	The student will translate mathematical expressions into programming expressions by declaring variables, writing assignment statements, and using the order of operations. [Moved from COM.12]
		COM.7	The student will select and call library functions to process data, as appropriate. [Moved from COM.13]
		COM.8	The student will implement conditional statements that include “if/then” statements, “if/then/else” statements, case statements, and Boolean logic. [Moved from COM.14]

2009 SOL		2016 SOL	
		COM.9	The student will implement pre-defined algorithms, including sort routines, search routines, and simple animation routines. [Moved from COM.17]
COM.6	The student will design and implement the input phase of a program, which will include designing screen layout and getting information into the program by way of user interaction, data statements, and/or file input. The input phase will also include methods of filtering out invalid data (error trapping).	Program Implementation	
		COM.10	The student will design and implement the input phase of a program, which will include designing screen layout, getting information into the program by way of user interaction and/or file input, and validating input.
COM.7	The student will design and implement the output phase of a computer program, which will include designing output layout, accessing a variety of output devices, using output statements, and labeling results.	COM.11	The student will design and implement the output phase of a computer program, which will include designing output layout, accessing available output devices, using output statements, and labeling results.
COM.8	The student will design and implement computer graphics, which will include topics appropriate for the available programming environment as well as student background. Students will use graphics as an end in itself, as an enhancement to other output, and as a vehicle for reinforcing programming techniques.	COM.12	The student will design and implement computer graphics to enhance output.
		COM.13	The student will implement various mechanisms for performing iteration with an algorithm. [Moved from COM.15]
		COM.14	The student will select and implement appropriate data structures, including arrays (one- and/or two-dimensional) and objects. [Moved from COM.16]
Data Manipulation			
COM.9	The student will define simple variable data types that include integer, real (fixed and scientific notation), character, string, and Boolean.	COM.15	The student will define and use appropriate variable data types that include integer, real (fixed and scientific notation), character, string, Boolean and object.
COM.10	The student will use appropriate variable data types, including integer, real (fixed and scientific notation), character, string, and Boolean. This will also include variables representing structured data types. [Combined with COM 15]		
COM.11	The student will describe the way the computer stores, accesses, and processes variables, including the following topics: the use of variables versus constants, variables' addresses, pointers, parameter passing, scope of variables, and local versus global variables.	COM.16	The student will describe the way the computer stores, accesses, and processes variables, including the following topics: the use of variables versus constants, parameter passing, scope of variables, and local versus global variables.

2009 SOL		2016 SOL	
COM.12	The student will translate a mathematical expression into a computer statement, which involves writing assignment statements and using the order of operations. [Moved to COM.6]		
COM.13	The student will select and implement built-in (library) functions in processing data. [Moved to COM.7]		
COM.14	The student will implement conditional statements that include “if/then” statements, “if/then/else” statements, case statements, and Boolean logic. [Moved to COM.8]		
COM.15	The student will implement loops, including iterative loops. Other topics will include single entry point, single exit point, preconditions, and postconditions. [Moved to COM.13]		
COM.16	The student will select and implement appropriate data structures, including arrays (one-dimensional and/or multidimensional), files, and records. Implementation will include creating the data structure, putting information into the structure, and retrieving information from the structure. [Moved to COM.14]		
COM.17	The student will implement pre-existing algorithms, including sort routines, search routines, and simple animation routines. [Moved to COM.9]		
COM.18	The student will test a program, using an appropriate set of data. The set of test data should be appropriate and complete for the type of program being tested.	Program Testing	
		COM.17	The student will test a program using an appropriate set of data. The test data should include boundary cases and test all branches of a program.
COM.19	The student will debug a program, using appropriate techniques (e.g., appropriately placed controlled breaks, the printing of intermediate results, other debugging tools available in the programming environment), and identify the difference between syntax errors and logic errors.	COM.18	The student will debug a program using appropriate techniques (e.g., appropriately placed controlled breaks, the printing of intermediate results, other debugging tools available in the programming environment), and identify the difference among syntax errors, runtime errors, and logic errors.
COM.20	The student will design, write, test, debug, and document a complete structured program that requires the synthesis of many of the concepts contained in previous standards. [Included in COM.2]		